Explainable AI Planning: Overview and the Case of Contrastive Explanation

Part 2: Explaining the Space of Plans

Jörg Hoffmann and Daniele Magazzeni

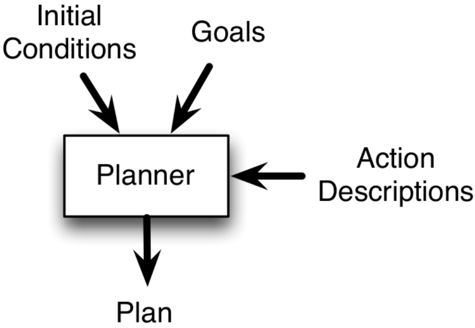
SAARLAND UNIVERSITY COMPUTER SCIENCE

KING'S College LONDON

September 23, 2019

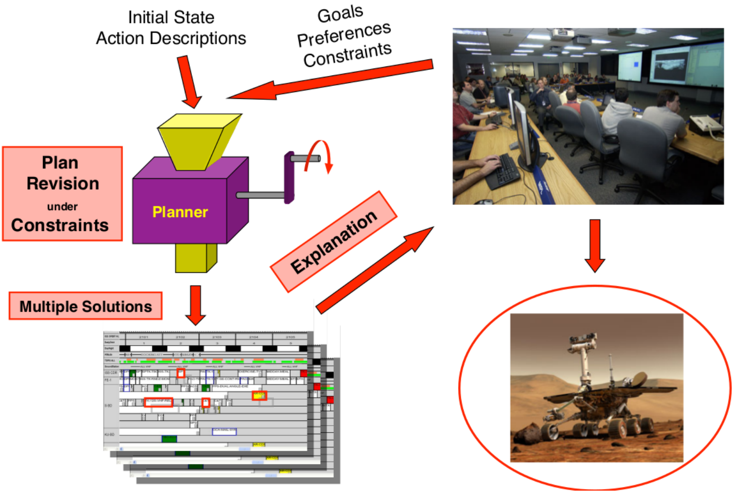
Agenda

The Traditional View of AI Planning



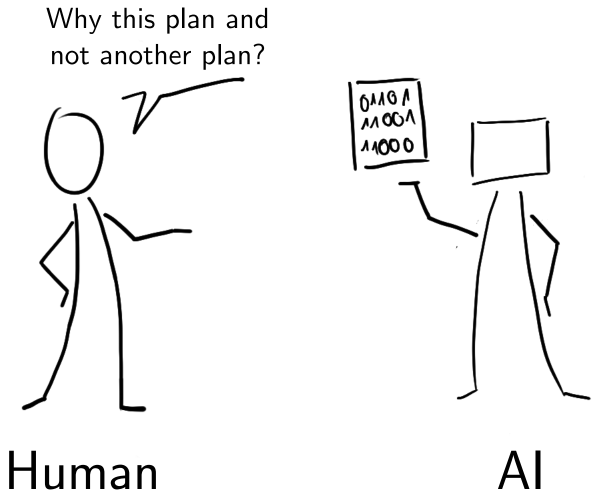
(Figure from [Smith (2012)])

A Typical Reality: Interactive Decision Making



(Figure from [Smith (2012)])

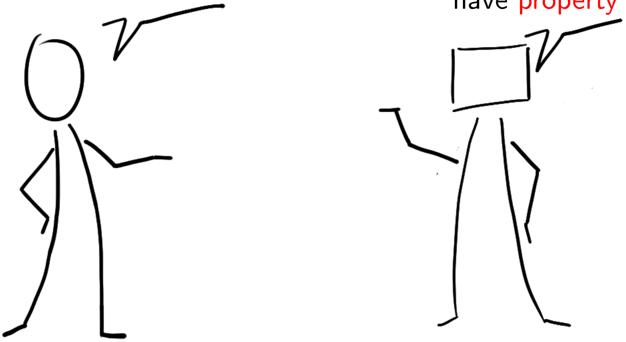
The Problem


Why this plan and not another plan?
0110 1
11001
11000
Human
AI

Our Approach: Plan-Property Dependencies

Why does this plan not have property A?

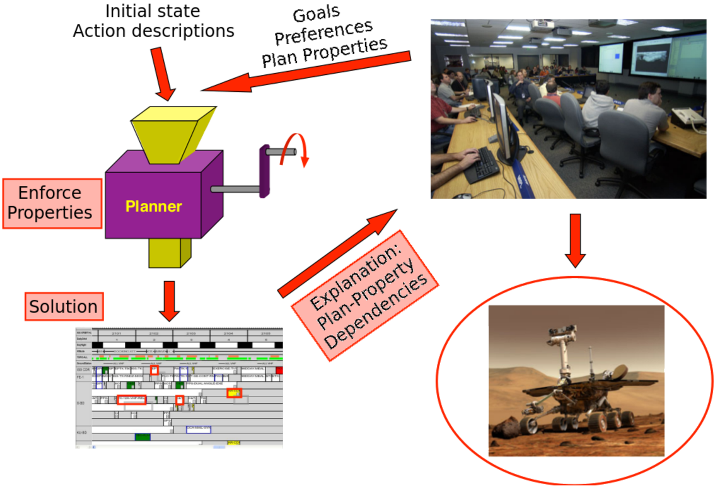
Because all plans with property A have property B!



Human

AI

Our Approach in Interactive AI Planning



(Figure adapted from [Smith (2012)])

Classical Planning

FDR Planning: Syntax

A finite-domain representation (FDR) task is a tuple $\tau = (V, A, c, I, G)$:

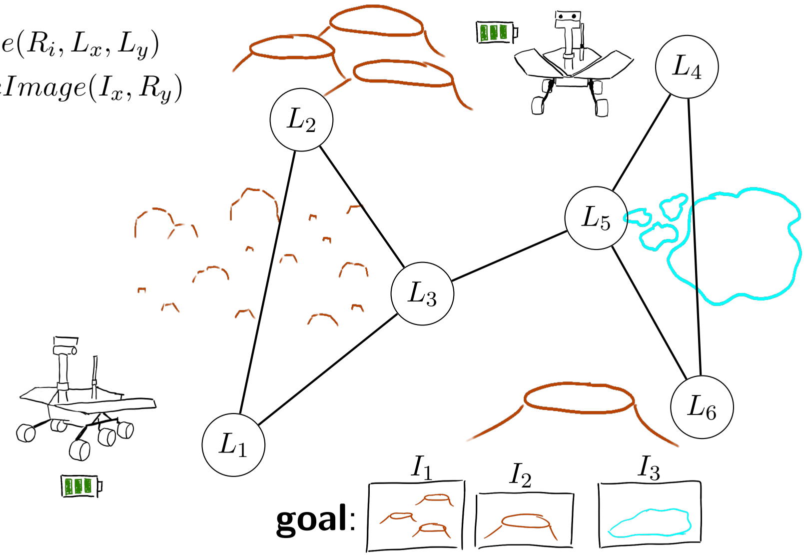
- V variables, each $v \in V$ with a finite domain D_v; a state is a complete assignment to V;
- A actions, each $a \in A$ has precondition pre_a and effect eff_a, both partial assignments to V;
- $c : A \to \mathbb{R}_0^+$ action-cost function;
- I initial state; G goal partial assignment to V;

FDR Planning: Semantics

Action a applicable in state s if $pre_a \subseteq s$. Outcome state $s[[a]]$ like s except that $s[[a]](v) = \mathit{eff}_a(v)$ for those v on which eff_a is defined. Outcome state of iteratively applicable action sequence π denoted $s[[\pi]]$. Sequence π applicable in I is plan if $G \subseteq I[[\pi]]$.

(Toy) Example: IPC Rovers

- $drive(R_i, L_x, L_y)$
- $takeImage(I_x, R_y)$



Oversubscription Planning

Oversubscription Planning [Smith (2004); Domshlak and Mirkis (2015)]

An OSP task is a tuple $\tau = (V, A, c, I, G^{\text{hard}}, G^{\text{soft}}, b)$:

- V variables, A actions, c action-cost function, I initial state;
- G^{hard} hard goal;
- G^{soft} soft goal;
- $b \in \mathbb{R}_0^+$ cost budget.

$\pi = \langle a_1, \dots, a_n \rangle$ is plan if $\sum_{i=1}^{n} c(a_i) \leq b$ and $G^{\text{hard}} \subseteq I[[\pi]]$.

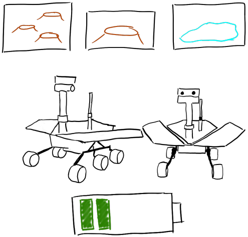
Plan quality: Usually additive soft-goal rewards. Here:

- User preferences hard to specify/elicitate. Iterative planning instead.
- Goal-exclusion dependencies to support that process.

OSP and Our Explanation Problem in Rovers

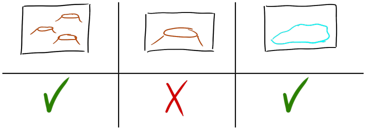
planning task

soft goals



plan

$drive(R_1, L_1, L2)$
$takeImage(I_1, R_1)$
$drive(R_2, L_4, L_5)$
$takeImage(I_3, R_2)$



Plan Properties

→ Plan properties over soft goals in OSP:

Plan Properties

OSP task $\tau = (V, A, c, I, G^{\text{hard}}, G^{\text{soft}}, b)$, Π its set of plans.

- Plan property: propositional formula ϕ over atoms $g \in G^{\text{soft}}$
- Conjunctive plan property: ϕ has form $\bigwedge_{g \in A} g$ or $\neg \bigwedge_{g \in B} g$

Simple special case: In general, any function $\Pi \to \{\mathit{true}, \mathit{false}\}$

- e. g. temporal plan trajectory constraints.
- e. g. deadlines, resource bounds.

Compilation: Into (additional variables/actions and) goal facts!

- e. g. LTL formulas
- Here: action-set properties, easy special case of LTL

Π-Entailment

→ Π in the role of a knowledge base:

Π-Entailment

OSP task $\tau = (V, A, c, I, G^{\text{hard}}, G^{\text{soft}}, b)$, Π its set of plans π.

- π satisfies ϕ, $\pi \models \phi$: if ϕ true given truth value assignment to $g \in G^{\text{soft}}$ defined by $g \in I[[\pi]]\ ?\ g \mapsto \mathit{true} : g \mapsto \mathit{false}$
- $\mathcal{M}_\Pi(\phi) := \{\pi \mid \pi \in \Pi, \pi \models \phi\}$
- ϕ Π-entails ψ, written $\Pi \models \phi \Rightarrow \psi$: if $\mathcal{M}_\Pi(\phi) \subseteq \mathcal{M}_\Pi(\psi)$

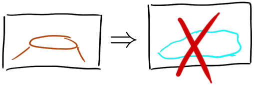
→ Special case focus here:

Goal Exclusions

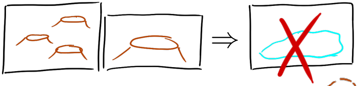
- Goal exclusion: entailment of form $\Pi \models \bigwedge_{g \in A} g \Rightarrow \neg \bigwedge_{g \in B} g$
- Non-dominated: $\nexists\ (A', B') \neq (A, B)$: $A' \subseteq A$, $B' \subseteq B$, $\Pi \models \bigwedge_{g \in A'} g \Rightarrow \neg \bigwedge_{g \in B'} g$

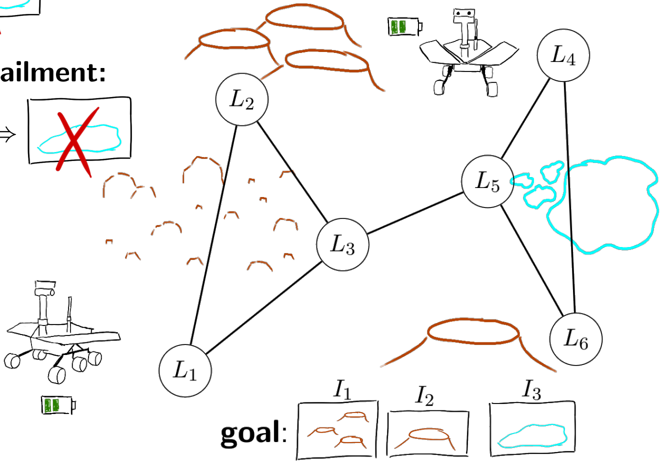
Π-Entailment: Rovers Example

Π-entailment:



Dominated Π-entailment:





Global Explanations

$\rightarrow$ All entailment relations over plan properties in the task:

Global Explanation (GE)

OSP task $\tau = (V, A, c, I, G^{\text{hard}}, G^{\text{soft}}, b)$, Π its set of plans.

- $[\phi]_\Pi$: equivalence class, i. e. set of ψ with $\mathcal{M}_\Pi(\phi) = \mathcal{M}_\Pi(\psi)$
- Global explanation (GE) for τ: strict partial order over equivalence classes, $[\phi]_\Pi < [\psi]_\Pi$ iff $[\phi]_\Pi \neq [\psi]_\Pi$ and $\Pi \models \phi \Rightarrow \psi$

$\rightarrow$ More practical variant for goal exclusions:

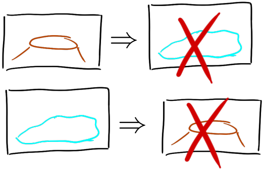
Goal-Exclusion GE

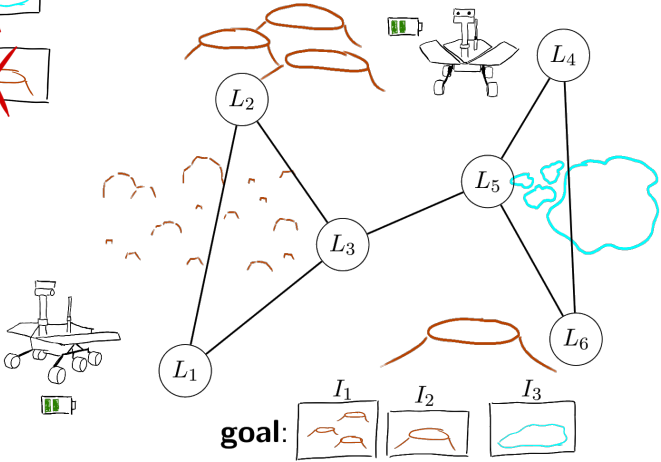
OSP task $\tau = (V, A, c, I, G^{\text{hard}}, G^{\text{soft}}, b)$, Π its set of plans.

- Goal-exclusion GE for τ: strict partial order over conjunctive plan properties induced by the non-dominated goal exclusions $\Pi \models \bigwedge_{g \in A} g \Rightarrow \neg \bigwedge_{g \in B} g$

Goal-Exclusion GE: Rovers Example

All non-dominated goal exclusions:





Local Explanations

$\rightarrow$ In response to user question "Why not property ϕ?":

Local Explanation (LE)

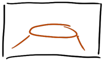
OSP task $\tau = (V, A, c, I, G^{\text{hard}}, G^{\text{soft}}, b)$, Π its set of plans.

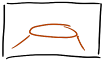
- Local explanation (LE) for ϕ: $\{\psi \mid \Pi \models \phi \Rightarrow \psi\}$
- Goal-exclusion LE for $\phi = \bigwedge_{g \in A} g$:
  $\{\psi \mid \psi = \neg \bigwedge_{g \in B} g, \Pi \models \phi \Rightarrow \psi$ is non-rhs-dominated$\}$
- Non-rhs-dominated: $\nexists\ B'$: $B' \subsetneq B$, $\Pi \models \bigwedge_{g \in A} g \Rightarrow \neg \bigwedge_{g \in B'} g$

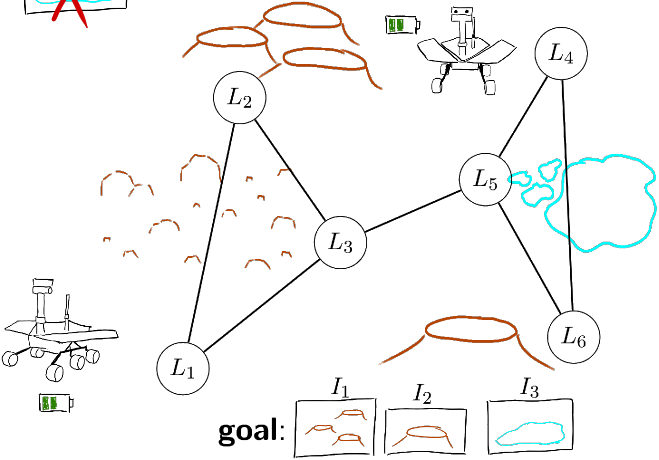
Remarks:

- Smaller and easier to compute than GE (see next section).
- Relative to current plan π in iterative planning: only those ψ where $\pi \not\models \psi$ (i. e. new properties entailed by ϕ).

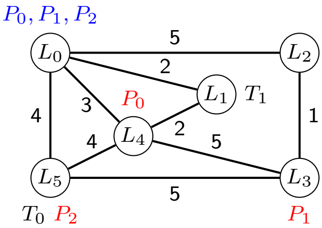
Goal-Exclusion LE: Rovers Example

Why does this plan not achieve  **?**

Because  $\Rightarrow$ 



Another Example: Transportation (IPC "NoMystery")



- Variables V: $T_0, f_0, T_1, f_1, P_0, P_1, P_2$
- Actions A: $drive(T_i, L_x, L_y)$, $load(T_i, P_j, L_x)$, $unload(T_i, P_j, L_x)$ Driving consumes fuel as indicated
- Initial state I: as shown; $I(f_0) = 13, I(f_1) = 0$
- Goal G^{soft}: $at(P_0, L_4), at(P_1, L_3), at(P_2, L_5)$

Non-dominated goal exclusions:

- $\Pi \models P_0 \Rightarrow \neg(P_1 \wedge P_2)$
- $\Pi \models P_1 \Rightarrow \neg(P_0 \wedge P_2)$
- $\Pi \models P_2 \Rightarrow \neg(P_0 \wedge P_1)$

$\rightarrow$ In other words: "G^{soft} is not solvable as a whole, but each of its subsets is".

Discussion/Literature/Challenges

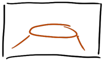
Framework intention:

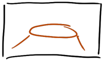
- Plan properties = language for (or finite set of) properties relevant to user preferences.
- Elucidate plan-property dependencies for interactive planning (instead of just fixing an optimization objective).
- Goal exclusions merely a simple starting point, yet powerful through compilation (see later).

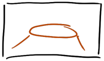
Positioning in literature:

- "Does $\Pi \models \phi \Rightarrow \psi$?" = model checking of planning task.
  $\Rightarrow$ Framework = exhaustive model checking of entailments within a set P of plan properties.
- Working hypothesis: Meaningful concept/special case.
  Computation: Exploit relatedness across individual checks.
- Very little prior work on model checking for planning models [Vaquero *et al.* (2013)].

Discussion/Literature/Challenges

Why does this plan not achieve  **?**

Because  $\Rightarrow$ 

But why  $\Rightarrow$  **?** "Why does $\Pi \models \phi \Rightarrow \psi$?"

- Idea 1: Extend set P of plan properties to elucidate "the causal chain between" ϕ and ψ.

  $\rightarrow$ One instance of problem *how to identify the relevant set* P.

- Idea 2: Find minimal relaxation (superset) of Π in which $\Pi \models \phi \Rightarrow \psi$ no longer holds.

  $\rightarrow$ Drop hard goals, increase cost budget, ...

Non-Dominated Goal Exclusions from MUGS

Minimal Unsolvable Goal Subset (MUGS)

OSP task $\tau = (V, A, c, I, G^{\text{hard}}, G^{\text{soft}}, b)$, Π its set of plans.

- Minimal unsolvable goal subset (MUGS): unsolvable $G \subseteq G^{\text{soft}}$, every $G' \subsetneq G$ solvable

Proposition (Non-dominated Goal Exclusions from MUGS)

Non-dominated $\Pi \models \bigwedge_{g \in A} g \Rightarrow \neg \bigwedge_{g \in B} g \Leftrightarrow A \cup B$ MUGS

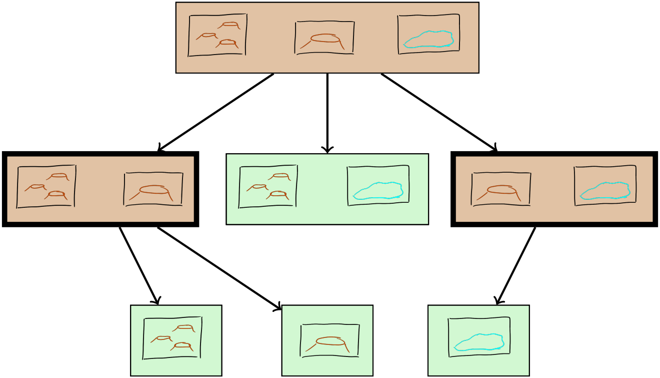
Proof.

A Π-entailment $\Pi \models \bigwedge_{g \in A} g \Rightarrow \neg \bigwedge_{g \in B} g$ clearly holds iff $A \cup B$ is unsolvable. Non-dominated entailments result from set-inclusion minimal A and B, corresponding to the set-inclusion minimality of MUGS. $\square$

$\rightarrow$ Compute and represent goal-exclusion GE via MUGs.

Systematic Weakening

1. Start with G^{soft}
2. Select open node G, call planner to test solvability, cache result, expand G if unsolvable
3. children of G: $G' \subset G$ where $|G'| = |G| - 1$



Experiments: Global Explanations

domain	Reference Coverage				SysS/W Coverage						Search Fraction				#MUGS, $x =$ average		
	$h^{\text{LM-cut}}$	OSP			0.25		0.5		0.75		0.25		0.75				
		0.25	0.5	0.75	S	W	S	W	S	W	S	W	S	W	0.25	0.5	0.75
agricola (20)	0	0	0	0	20	20	13	13	1	1	1	0.5	1	0.5	1	1	1
airport (50)	28	28	24	22	35	34	21	21	16	16	0.6	0.81	1	0.61	3.8	2	1.4
barman (34)	4	18	11	4	18	18	4	4	3	4	0.57	0.94	1	0.5	6.9	4.2	2.5
blocks (35)	28	35	28	21	35	35	29	29	26	26	0.15	0.97	0.8	0.64	11	12.4	13.7
childsnack (20)	0	2	0	0	4	4	0	0	0	0	0.34	0.98	-	-	16.8	-	-
data-network (20)	12	13	13	13	20	20	18	18	17	15	0.72	0.73	0.91	0.66	2.1	1.8	1.5
depot (22)	7	16	11	7	16	16	9	10	3	3	0.24	0.96	0.89	0.68	8.3	7	6.5
driverlog (20)	13	15	13	10	15	15	12	12	10	10	0.17	0.98	0.87	0.5	8.1	16.1	11.1
elevators (50)	40	22	22	22	47	48	38	37	27	27	0.35	0.94	0.9	0.67	4.6	5.1	5.9
floortile (36)	13	18	6	2	8	8	2	2	2	2	0.1	0.99	0.96	0.3	316.2	137	45.5
freecell (80)	15	77	30	21	76	76	30	30	18	18	0.31	0.94	0.88	0.76	4	4.3	3.3
ged (20)	15	20	20	20	16	20	10	12	7	7	0.25	0.9	0.58	0.7	13.3	38.7	12.5
grid (5)	2	5	3	2	5	5	3	4	3	3	0.54	0.84	1	0.54	4	2.5	1
gripper (20)	7	11	8	8	5	5	4	4	3	3	0.21	0.98	0.96	0.46	783.5	228	156
hiking (20)	9	19	14	13	20	20	16	17	11	10	0.81	0.69	1	0.63	1.8	1.7	1
logistics (60)	26	27	20	16	15	15	6	6	3	4	0.35	0.95	0.98	0.73	7.2	7.3	2.8
miconic (150)	141	97	66	55	66	64	42	43	35	36	0.3	0.92	0.95	0.61	81.3	38.2	18.8
mprime (35)	22	35	27	24	35	35	35	35	35	35	0.9	0.59	0.94	0.59	1.3	1.2	1.1
mystery (30)	12	29	27	21	30	30	30	30	30	30	0.89	0.61	0.93	0.61	1.3	1.2	1.1
nomystery (20)	14	20	14	10	20	20	12	12	8	8	0.15	0.98	0.87	0.61	20.2	18.5	5.8
openstacks (77)	47	63	56	52	49	43	45	39	42	35	0.03	0.99	0.12	0.98	15.3	14.9	10.3
organic-syn-s (13)	10	8	8	8	8	8	8	8	6	6	0.19	0.96	0.28	0.91	5.3	7.3	8.3
parcprinter (26)	24	26	22	18	10	14	10	14	10	12	0.44	0.98	0.73	0.85	5.6	7.5	4.1
parking (40)	5	25	5	0	17	12	1	1	0	0	0.02	1	-	-	63.9	31	
pathways (30)	5	5	4	4	7	7	5	5	4	4	0.41	0.86	0.91	0.7	11.3	3.8	1.8
pegsol (2)	2	2	2	2	0	2	0	2	0	2	-	-	-	-	7	23.5	64
pipesworld-nt (50)	17	45	30	23	46	46	25	26	15	15	0.31	0.94	0.88	0.66	5	5.6	4.3
pipesworld-t (50)	12	33	20	16	39	40	18	17	13	11	0.35	0.95	0.88	0.65	4	4.2	3.2
Sum (1517)	828	1088	828	705	1026	1005	690	694	522	528							

Local Explanations from MUGS

$\rightarrow$ In response to user question "Why not property $\bigwedge_{g \in A} g$?":

Proposition (Non-rhs-dominated Goal Exclusions from MUGS)

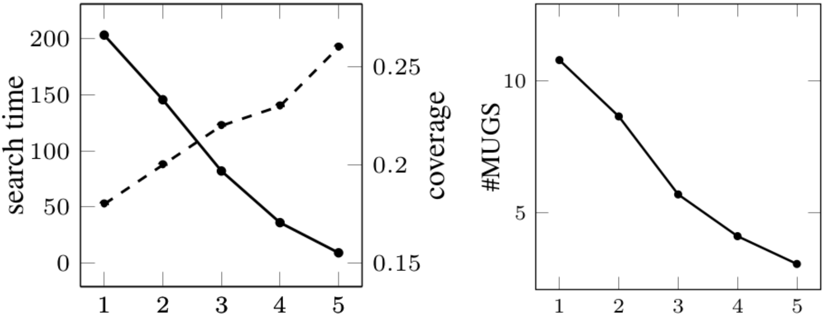
OSP task $\tau = (V, A, c, I, G^{\text{hard}}, G^{\text{soft}}, b)$, Π its set of plans.
Modified task $\tau' := (V, A, c, I, G^{\text{hard}} \cup A, G^{\text{soft}} \setminus A, b)$.
Then: Non-rhs-dominated $\Pi \models \bigwedge_{g \in A} g \Rightarrow \neg \bigwedge_{g \in B} g \Leftrightarrow B$ MUGS in τ'.

This is easier & smaller because: Less soft goals!

- Search tree size worst-case exponential in $|G^{\text{soft}}|$
- $\#$MUGS worst-case exponential in $|G^{\text{soft}}|$

Experiments: Local Explanations

→ Performance and #MUGS as function of question size $|A|$ in "Why not property $\bigwedge_{g \in A} g$?":



Discussion/Literature/Challenges

Many related computations: (for different purposes)

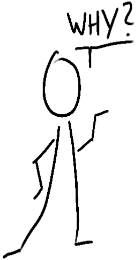
- Minimal unsatisfiable cores [e. g. Chinneck (2007); Laborie (2014)].
- Solvability borderline within a lattice of problem variants [de Kleer (1986); Reiter (1987)]
- MUGS = special case of preferred diagnoses [Grastien *et al.* (2011, 2012)], transfer pruning methods?
- Suggesting goals to drop in oversubscribed situations [Yu *et al.* (2017); Lauffer and Topcu (2019)].

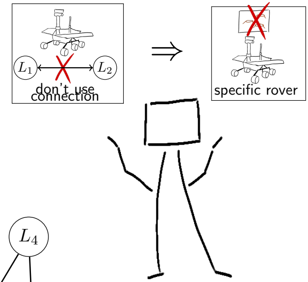
Alternative algorithms to try:

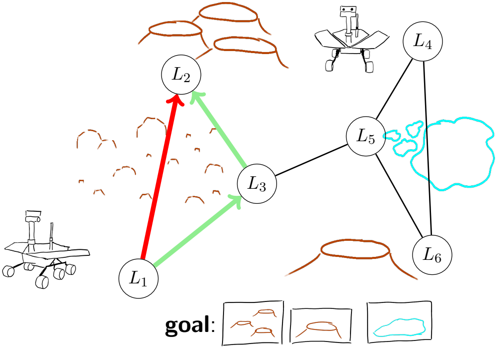
- Run a single search in state space finding all maximal solvable goal subsets. Adapt pruning methods from oversubscription planning [Domshlak and Mirkis (2015)]?
- Represent the plan set Π symbolically (e. g. BDD), use that representation to identify all entailment relations?

We Want More General Plan Properties!

$drive(R_1, L_1, L2)$
$takeImage(I_1, R_1)$
$drive(R_2, L_4, L_5)$
$takeImage(I_3, R_2)$
$drive(R_2, L_5, L_6)$
$takeImage(I_2, R_2)$







Compilation!

→ Compile more general plan properties into (additional variables/actions and) goal facts! For example:

- Precondition and goal formulas, conditional effects [Gazen and Knoblock (1997); Nebel (2000)]
- LTL formulas over plan trajectory [Edelkamp (2006); Baier *et al.* (2009)]
- Initial state uncertainty [Palacios and Geffner (2009)]

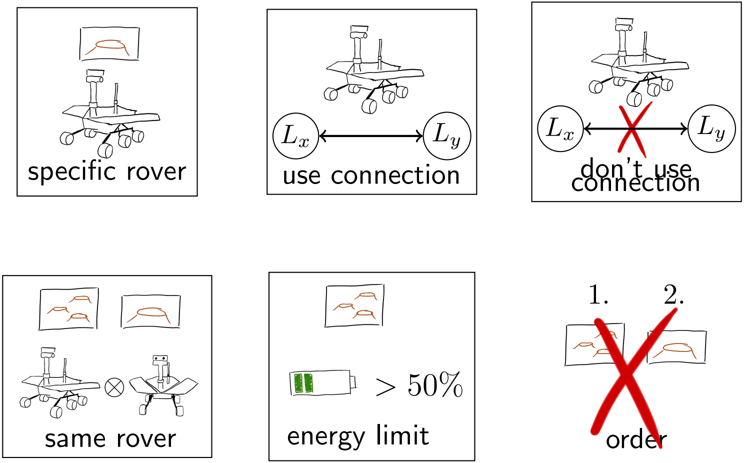
→ Here: effective-to-compile special case of LTL

Action-Set Properties

OSP task $\tau = (V, A, c, I, G^{\text{hard}}, G^{\text{soft}}, b)$, Π its set of plans, $A_1, \dots, A_n \subseteq A$.

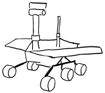
- Action-set property: propositional formula ϕ over atoms $A_1, \dots, A_n$
- $\pi \models \phi$: if ϕ true given truth value assignment $A_i \cap \{a_1, \dots, a_k\} \neq \emptyset$? $A_i \mapsto \mathit{true}$: $A_i \mapsto \mathit{false}$ where $\pi = \langle a_1, \dots, a_k \rangle$

Rovers Action-Set Properties

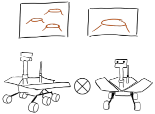


Rovers Action-Set Properties: "Same Rover"

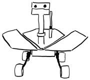
Action sets:



$A_1 = \{takeImage(I_1, R_1), takeImage(I_2, R_1)\}$



same rover



$A_2 = \{takeImage(I_1, R_2), takeImage(I_2, R_2)\}$

Test formula: $$A_1 \otimes A_2$$

Action-Set Property Compilation

Given: τ, Π, and $A_1, \dots, A_n$

Construct: τ'

- Booleans $isUsed_i$, initially $false$, set to $true$ by any action from A_i;
- formula-evaluation state variables and actions evaluating each p_ϕ based on these, setting Boolean flags $isTrue_\phi$;
- separate 1. planning phase vs. 2. formula-evaluation phase, switch action from 1. to 2. enabled when G^{hard} is satisfied.

$\Rightarrow$ planning-phase prefixes in τ' one-to-one Π; given such prefix π, evaluation phase in τ' can achieve $isTrue_\phi$ iff $\pi \models \phi$.

Rovers "Same Rover" Compilation: Illustration

$$A_1 = \{takeImage(I_1, R_1), takeImage(I_2, R_1)\}$$

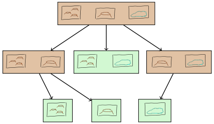
$$A_2 = \{takeImage(I_1, R_2), takeImage(I_2, R_2)\}$$

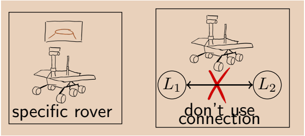
$$A_1 \otimes A_2$$

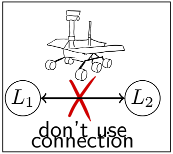
$drive(R_1, L_1, L2)$
$takeImage(I_1, R_1)$
$drive(R_2, L_4, L_6)$
$takeImage(I_2, R_2)$
$drive(R_2, L_6, L_5)$
$\dots$

	used
A_1	✓
A_2	✓

MUGS

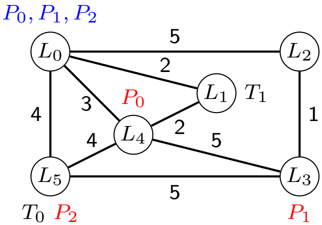






$\Longrightarrow$

Transportation Example (NoMystery)



- Variables V: $T_1, f_1, T_2, f_2, P_0, P_1, P_2$
- Actions A: $drive(T_i, L_x, L_y)$, $load(T_i, P_j, L_x)$, $unload(T_i, P_j, L_x)$ Driving consumes fuel as indicated
- Initial state I: as shown; $I(f_1) = 16, I(f_2) = 7$
- Goal G^{soft}: $at(P_0, L_4), at(P_1, L_3), at(P_2, L_5)$

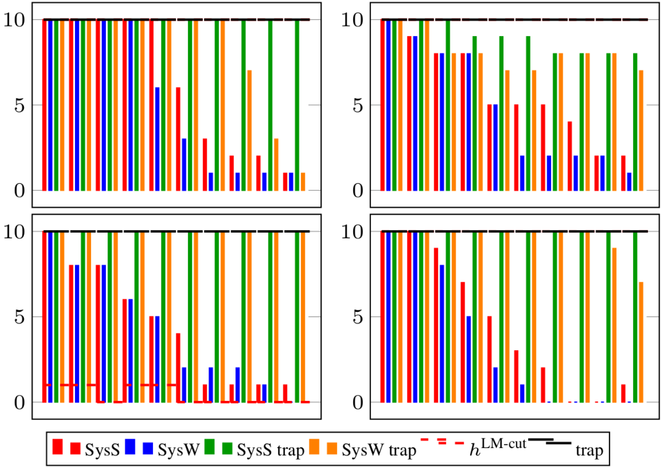
Example action-set property analysis:

- 1. uses T_0 (L_2, L_3); 2. same truck P_1 P_2; 3. uses T_0 (L_4, L_3); 4. same truck P_2 P_0; 5. doesn't use T_0 (L_0, L_5); 6. uses T_1 (L_5, L_4).
- MUGS: 7, each of size 3, including $\{5, 2, 4\}$.

$\rightarrow$ User question "Why do you not avoid the road $L_0 - L_5$ (which has a lot of traffic at the moment)?" "Because if you don't use that road, then you cannot deliver all packages with the same truck."

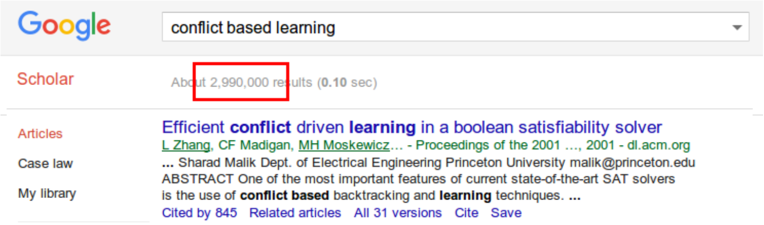
Experiments: Global Explanations

→ Blocksworld, NoMystery, Rovers, TPP (top left to bottom right):



State-Space Conflict Based Learning

Conflict Based Learning: Is all over the place!

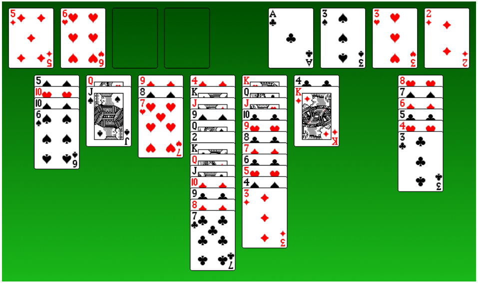


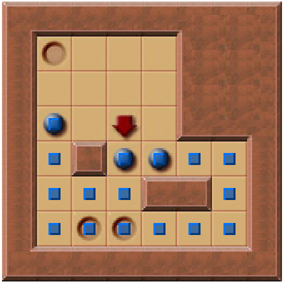
But: "State-space"

- Conflict-based learning is ubiquitous in constraint reasoning.
- Planning/reachability checking: Limited to bounded-length reachability (which is a form of constraint reasoning, easily encoded into e. g. SAT).
- Can we learn from conflicts in unbounded-length state space search?

Conflicts in State Space Search

What is a "conflict" in state space search?





→ Conflict = dead-end state from which the goal is unreachable.

- Planning: took bad decisions (ran out of resources, etc).
- Model checking safety properties: error can't be reached from here.

Learning from Dead-End States

Constraint reasoning: For unsolvable partial assignment α that does not violate the constraints, add a new constraint discarding α.

Basic idea: Constraints $\approx$ sound but incomplete dead-end detector Δ.

$\rightarrow$ For unsolvable state s not detected by Δ, refine Δ to detect s.

What are suitable Δ? E.g. Δ^C, set C of atomic conjunctions :

$$\Delta^C(s,g) = \begin{cases} 0 & g \subseteq s \\ \min_{a:Regress(g,a)\neq\bot} \Delta^C(s, Regress(g,a)) & g \in C \\ \max_{g' \subseteq g, g' \in C} \Delta^C(s,g') & \text{else} \end{cases}$$

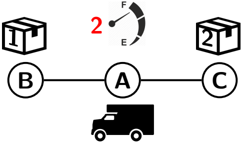
$\rightarrow$ $\Delta^C(s,G) = \infty$: "Goal unreachable even when breaking up conjunctive subgoals into elements of C." For suitable C, Δ^C detects all dead-ends.

Conflict-Learning State Space Search: [Steinmetz and Hoffmann (2016, 2017c)]

- Start with C containing the singleton conjunctions.
- On dead-end s where $\Delta^C(s,G) \neq \infty$, refine Δ^C by adding new atomic conjunctions, i. e., by extending C such that $\Delta^C(s,G) = \infty$.
- Further, learn a clause ϕ where $s' \not\models \phi$ implies $\Delta^C(s',G) = \infty$.

A Simple Transportation Example

Classical Planning Task:



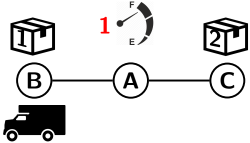
- $V = \{t, f, p_1, p_2\}$ with $D_t = \{A, B, C\}$, $D_f = \{0, 1, 2\}$, $D_{p_i} = \{A, B, C, T\}$.
- $A = \{load(p_i, x), unload(p_i, x), drive(x, x', n)\}$, where e. g.: $pre_{drive(x,x',y)} = \{(t, x), (f, n)\}$ and $eff_{drive(x,x',y)} = \{(t, x'), (f, n-1)\}$.
- $I = \{(t, A), (f, 2), (p_1, B), (p_2, C)\}$. $G = \{(p_1, C), (p_2, B)\}$.

Conflict-Learning State Space Search

1. Forward state space search.
2. Identify dead-end states s.
3. Refine C so that $\Delta^C(s, G) = \infty$.
4. Learn a clause ϕ s.t. $s' \not\models \phi$ implies $\Delta^C(s', G) = \infty$.

Δ^C in the Example

State: $tB, f1, p_1B, p_2C$



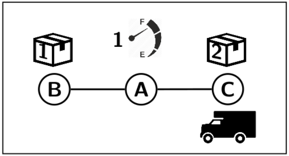
$C := C_1$ containing the singleton conjunctions:

1. $tB,\ f1,\ p_1B,\ p_2C$
2. $drive(B, A, 1) \rightarrow tA,\ load(p_1, B) \rightarrow p_1t$
3. $drive(A, C, 1)\ [pre : tB,\ f1] \rightarrow tC$
4. $unload(p_1, C) \rightarrow p_1C,\ load(p_2, C) \rightarrow p_2t$
5. $unload(p_2, B) \rightarrow p_2B$

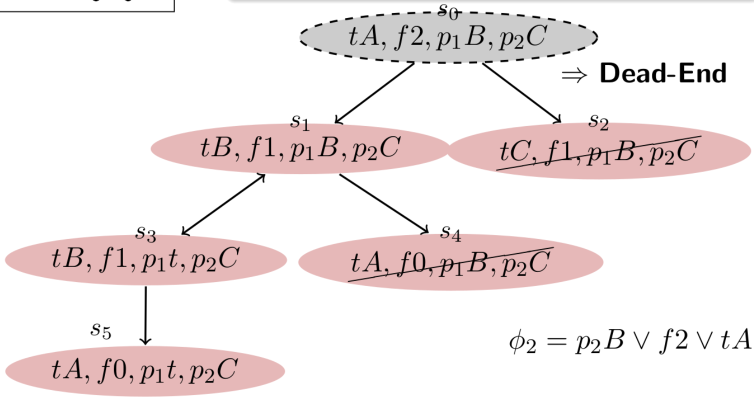
$C := C_1 \cup \{tA \wedge f1\}$:

1. $tB,\ f1,\ p_1B,\ p_2C$
2. $drive(B, A, 1) \rightarrow tA\ [but \not\rightarrow tA \wedge f1],\ load(p_1, B) \rightarrow p_1t$
3. $\cancel{drive(A, C, 1)}\ [pre : tA \wedge f1]$

Conflict-Learning State Space Search in the Example



3. Refine C so that $\Delta^C(s, G) = \infty$.
4. Learn a clause ϕ s.t. $s' \not\models \phi$ implies $\Delta^C(s', G) = \infty$.



$$\phi_2 = p_2B \lor f2 \lor tA$$

$$\phi_1 = p_2B \lor tB \lor f1 \lor f2$$

Discussion/Literature/Challenges

Mature exploration of variants:

- Trap learning [Steinmetz and Hoffmann (2017b)]
- Refining more powerful LP-based dead-end detectors [Steinmetz and Hoffmann (2018)]
- Offline nogood computation [Steinmetz and Hoffmann (2017a)]

Open questions:

- Can we reason over the learned clauses, deducing new knowledge from the already derived one?
- Combine with property-directed reachability (PDR) [Bradley (2011); Suda (2014)]: Combine PDR clauses with clauses from different Δ; use lower-bound heuristic functions for additional pruning in PDR; use heuristic functions for node selection in PDR.
- Apply these methods to model checking and game playing ...

Last Slide

Thanks for your attention. Questions?

References I

Jorge A. Baier, Fahiem Bacchus, and Sheila A. McIlraith. A heuristic search approach to planning with temporally extended preferences. *Artificial Intelligence*, 173(5-6):593–618, 2009.

Aaron R. Bradley. Sat-based model checking without unrolling. In *Proceedings of the 12th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI'11)*, pages 70–87, 2011.

John W. Chinneck. *Feasibility and Infeasibility in Optimization: Algo- rithms and Computational Methods*. Springer-Verlag, 2007.

Johan de Kleer. An assumption-based TMS. *Artificial Intelligence*, 28(2):127–162, March 1986.

Carmel Domshlak and Vitaly Mirkis. Deterministic oversubscription planning as heuristic search: Abstractions and reformulations. *Journal of Artificial Intelligence Research*, 52:97–169, 2015.

Stefan Edelkamp. On the compilation of plan constraints and preferences. In Derek Long and Stephen Smith, editors, *Proceedings of the 16th International Conference on Automated Planning and Scheduling (ICAPS'06)*, pages 374–377, Ambleside, UK, 2006. Morgan Kaufmann.

References II

B. Cenk Gazen and Craig Knoblock. Combining the expressiveness of UCPOP with the efficiency of Graphplan. In S. Steel and R. Alami, editors, *Proceedings of the 4th European Conference on Planning (ECP'97)*, pages 221–233. Springer-Verlag, 1997.

Alban Grastien, Patrik Haslum, and Sylvie Thiébaux. Exhaustive diagnosis of discrete event systems through exploration of the hypothesis space. In *Proc. International Workshop on Principles of Diagnosis (DX)*, 2011.

Alban Grastien, Patrik Haslum, and Sylvie Thiébaux. Conflict-based diagnosis of discrete event systems: Theory and practice. AAAI Press, 2012.

Philippe Laborie. An optimal iterative algorithm for extracting mucs in a black-box constraint network. In Thorsten Schaub, editor, *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI'14)*, pages 1051–1052, Prague, Czech Republic, August 2014. IOS Press.

Niklas Lauffer and Ufuk Topcu. Human-understandable explanations of infeasibility for resource-constrained scheduling problems. In *Proc. 2nd Workshop on Explainable Planning (XAIP'19)*, 2019.

Bernhard Nebel. On the compilability and expressive power of propositional planning formalisms. *Journal of Artificial Intelligence Research*, 12:271–315, 2000.

References III

Hector Palacios and Hector Geffner. Compiling uncertainty away in conformant planning problems with bounded width. *Journal of Artificial Intelligence Research*, 35:623–675, 2009.

Raymond Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–95, April 1987.

David E. Smith. Choosing objectives in over-subscription planning. In Sven Koenig, Shlomo Zilberstein, and Jana Koehler, editors, *Proceedings of the 14th International Conference on Automated Planning and Scheduling (ICAPS'04)*, pages 393–401, Whistler, Canada, 2004. Morgan Kaufmann.

David Smith. Planning as an iterative process. In Jörg Hoffmann and Bart Selman, editors, *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI'12)*, pages 2180–2185, Toronto, ON, Canada, July 2012. AAAI Press.

Marcel Steinmetz and Jörg Hoffmann. Towards clause-learning state space search: Learning to recognize dead-ends. In Dale Schuurmans and Michael Wellman, editors, *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI'16)*. AAAI Press, February 2016.

References IV

Marcel Steinmetz and Jörg Hoffmann. Critical-path dead-end detection vs. nogoods: Offline equivalence and online learning. In *Proceedings of the 27th International Conference on Automated Planning and Scheduling (ICAPS'17)*. AAAI Press, 2017.

Marcel Steinmetz and Jörg Hoffmann. Search and learn: On dead-end detectors, the traps they set, and trap learning. In Carles Sierra, editor, *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*. AAAI Press/IJCAI, 2017.

Marcel Steinmetz and Jörg Hoffmann. State space search nogood learning: Online refinement of critical-path dead-end detectors in planning. *Artificial Intelligence*, 245:1–37, 2017.

Marcel Steinmetz and Jörg Hoffmann. LP heuristics over conjunctions: Compilation, convergence, nogood learning. In Jérôme Lang, editor, *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*, pages 4837–4843. ijcai.org, 2018.

Martin Suda. Property directed reachability for automated planning. *Journal of Artificial Intelligence Research*, 50:265–319, 2014.

References V

Tiago Stegun Vaquero, José Reinaldo Silva, Flavio Tonidandel, and J. Christopher Beck. itsimple: towards an integrated design system for real planning applications. *Knowledge Engineering Review*, 28(2):215–230, 2013.

Peng Yu, Brian Charles Williams, Cheng Fang, Jing Cui, and Patrik Haslum. Resolving over-constrained temporal problems with uncertainty through conflict-directed relaxation. *Journal of Artificial Intelligence Research*, 60:425–490, 2017.