Landmarks Revisited

Silvia Richter[1] Malte Helmert[2] Matthias Westphal[2]

[1]Griffith University & NICTA, Australia

[2]Albert-Ludwigs-Universität Freiburg, Germany

AAAI 2008

Outline

Outline

1. Introduction to SAS$^+$ Planning
2. Landmarks in Previous Work
3. Using Landmarks as Pseudo-Heuristic
4. Extended Landmark Generation

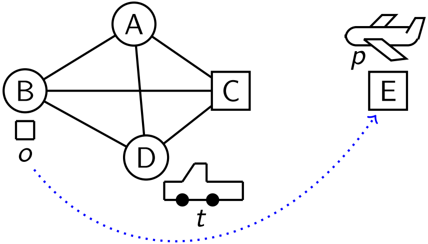
SAS^+ planning task

SAS^+ planning task: $\Pi = \langle \mathcal{V}, \mathcal{A}, s_0, s_\star \rangle$

- $\mathcal{V}$: state variables with finite domain $\mathcal{D}_v$
  Fact: variable-value pair $v \mapsto d$ ($v \in \mathcal{V}$, $d \in \mathcal{D}_v$)
  State: variable assignment for all $v \in \mathcal{V}$
- $\mathcal{A}$: actions $\langle pre, eff \rangle$, with *pre*, *eff* fact sets
  - Action $a = \langle pre, eff \rangle$ applicable in state s if $pre \subseteq s$
  - Applying a in s updates s
- s_0: initial state
- $s_\star$: partial variable assignment called the goal

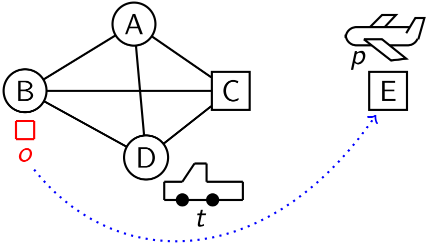
Sequence of actions π a plan iff $s_\star \subseteq s_0[\pi]$.

Encoding of example task



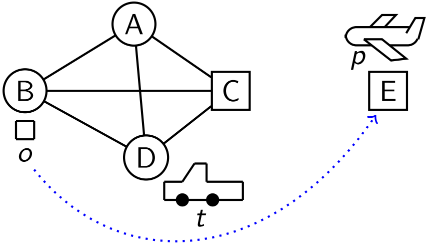
$$
\begin{aligned}
\mathcal{V} &= \{v_o, v_t, v_p\} \\
\mathcal{D}_{v_o} &= \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}, \mathrm{t}, \mathrm{p}\} \quad \mathcal{D}_{v_t} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}\}, \quad \mathcal{D}_{v_p} = \{\mathrm{C}, \mathrm{E}\} \\
\mathcal{A} &= \{\text{drive-t-D-B}, \quad \text{load-o-t-B}, \dots\} \\
s_0 &= \{v_o \mapsto \mathrm{B}, \quad v_t \mapsto \mathrm{D}, \quad v_p \mapsto \mathrm{E}\} \\
s_\star &= \{v_o \mapsto \mathrm{E}\}
\end{aligned}
$$

Encoding of example task



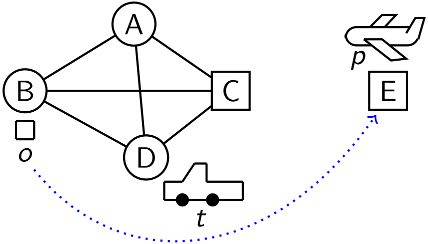
$$
\begin{aligned}
\mathcal{V} &= \{v_o, v_t, v_p\} \\
\mathcal{D}_{v_o} &= \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}, \mathrm{t}, \mathrm{p}\} \quad \mathcal{D}_{v_t} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}\}, \quad \mathcal{D}_{v_p} = \{\mathrm{C}, \mathrm{E}\} \\
\mathcal{A} &= \{\text{drive-t-D-B}, \quad \text{load-o-t-B}, \ldots\} \\
s_0 &= \{v_o \mapsto \mathrm{B}, \quad v_t \mapsto \mathrm{D}, \quad v_p \mapsto \mathrm{E}\} \\
s_\star &= \{v_o \mapsto \mathrm{E}\}
\end{aligned}
$$

Encoding of example task



$$
\begin{aligned}
\mathcal{V} &= \{v_o, v_t, v_p\} \\
\mathcal{D}_{v_o} &= \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}, \mathrm{t}, \mathrm{p}\} \quad \mathcal{D}_{v_t} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}\}, \quad \mathcal{D}_{v_p} = \{\mathrm{C}, \mathrm{E}\} \\
\mathcal{A} &= \{\text{drive-t-D-B}, \quad \text{load-o-t-B}, \ldots\} \\
s_0 &= \{v_o \mapsto \mathrm{B}, \quad v_t \mapsto \mathrm{D}, \quad v_p \mapsto \mathrm{E}\} \\
s_\star &= \{v_o \mapsto \mathrm{E}\}
\end{aligned}
$$

Encoding of example task



$$\mathcal{V} = \{v_o, v_t, v_p\}$$

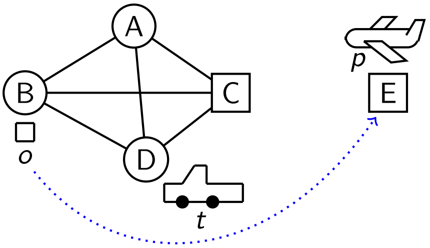
$$\mathcal{D}_{v_o} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}, \mathrm{t}, \mathrm{p}\} \quad \mathcal{D}_{v_t} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}\}, \quad \mathcal{D}_{v_p} = \{\mathrm{C}, \mathrm{E}\}$$

$$\mathcal{A} = \{\text{drive-t-D-B}, \quad \text{load-o-t-B}, \dots\}$$

$$s_0 = \{v_o \mapsto \mathrm{B}, \quad v_t \mapsto \mathrm{D}, \quad v_p \mapsto \mathrm{E}\}$$

$$s_\star = \{v_o \mapsto \mathrm{E}\}$$

Encoding of example task



$$\mathcal{V} = \{v_o, v_t, v_p\}$$

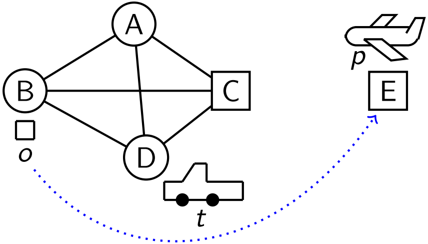
$$\mathcal{D}_{v_o} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}, \mathrm{t}, \mathrm{p}\} \quad \mathcal{D}_{v_t} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}\}, \quad \mathcal{D}_{v_p} = \{\mathrm{C}, \mathrm{E}\}$$

$$\mathcal{A} = \{\text{drive-t-D-B}, \quad \text{load-o-t-B}, \dots\}$$

$$s_0 = \{v_o \mapsto \mathrm{B}, \quad v_t \mapsto \mathrm{D}, \quad v_p \mapsto \mathrm{E}\}$$

$$s_\star = \{v_o \mapsto \mathrm{E}\}$$

Encoding of example task



$$\mathcal{V} = \{v_o, v_t, v_p\}$$

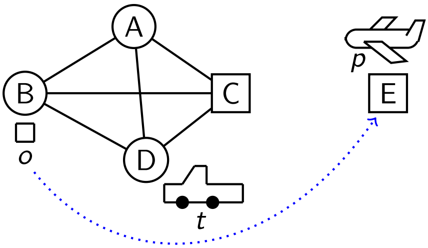
$$\mathcal{D}_{v_o} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}, \mathrm{t}, \mathrm{p}\} \quad \mathcal{D}_{v_t} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}\}, \quad \mathcal{D}_{v_p} = \{\mathrm{C}, \mathrm{E}\}$$

$$\mathcal{A} = \{\text{drive-t-D-B}, \quad \text{load-o-t-B}, \dots\}$$

$$s_0 = \{v_o \mapsto \mathrm{B}, \quad v_t \mapsto \mathrm{D}, \quad v_p \mapsto \mathrm{E}\}$$

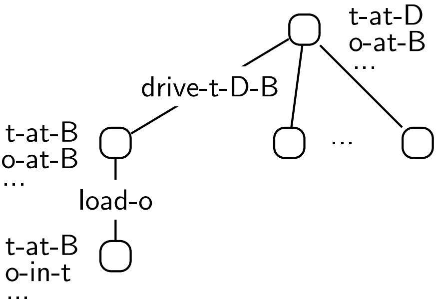
$$s_\star = \{v_o \mapsto \mathrm{E}\} \qquad \text{o-at-E}$$

Encoding of example task cont'd



$$\begin{aligned} \text{load-o-t-B} : \langle \text{Pre} &= \{v_o \mapsto \mathrm{B},\ v_t \mapsto \mathrm{B}\}, \\ \text{Eff} &= \{v_o \mapsto \mathrm{t}\}\rangle \end{aligned}$$

Preferred Operators

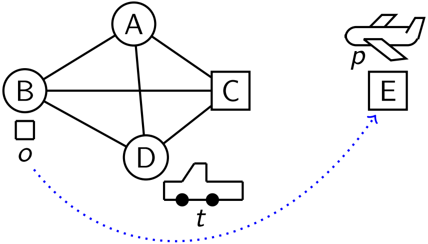


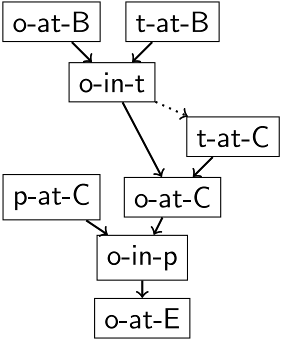
- Improvement of heuristic search approaches (Helmert 2006)
- Idea: prefer actions that are likely to improve heuristic value
- E. g. those which are part of plan for simplified problem

Outline

Landmarks in Previous Work

- Facts that must be true in every plan (Porteous et al. 2001 & 2002; Hoffmann et al. 2004)
- Intuitively helpful to direct seach
- Automatically found, incl. orderings

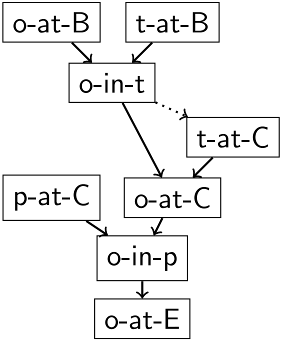




Landmarks in Previous Work cont'd

Find landmarks by backchaining

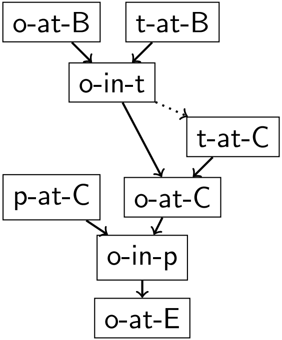
- Every goal is a landmark
- If B is landmark and all actions that first achieve B have A as precondition, then A is a landmark
- Approximation with RPGs: consider all achievers "possibly before" B (Porteous et al. 2002)



- Disjunctive landmarks also possible: (o-in-p_1 $\vee$ o-in-p_2)

Landmarks in Previous Work cont'd

- Use as subgoals, then simply concatenate plans of subtasks ("LM-local")
- Greatly speeds up search in many domains
- But: bad-quality plans, incomplete (dead ends)
- Any base planner possible for subtasks



Outline

Landmark Heuristic + Preferred Operators

Novel usage of landmarks

- Pseudo-Heuristic = #landmarks that still need to be achieved
- Take orderings into account (see paper for details)
- Preferred operators = landmark-achieving operators or operators in relaxed plan to nearest landmark
- Combination with other heuristics through multi-heuristic BFS (Helmert 2006)

Experiments with several heuristics (FF, CG, blind) on all tasks from past planning competitions

Results: %Tasks solved (Average)

	Algorithm		
Base Heuristic	base	LM-local	LM-heur
FF heuristic	87	82	**88**
CG heuristic	74	66	**87**
blind heuristic	25	52	**84**

Note: updated results for LM-local

- With all 3 heuristics, LM-heur dominates other approaches
- LM-local worse than base with CG and blind heuristic (dead ends in 8 domains)
- FF-heuristic: base and LM-local are close. . .

Results: %Tasks solved (Average)

	Algorithm		
Base Heuristic	base	LM-local	LM-heur
FF heuristic	**87**	82	**88**
CG heuristic	74	66	87
blind heuristic	25	52	84

- With all 3 heuristics, LM-heur dominates other approaches
- LM-local worse than base with CG and blind heuristic (dead ends in 8 domains)
- FF-heuristic: base and LM-local are close. . .

Results: #Tasks solved exclusively (FF heuristic)

Domain	FF heuristic base	FF heuristic LM-heur
Airport (50)	**6**	2
Depot (22)	0	**2**
Freecell (80)	1	**3**
Logistics-1998 (35)	0	**2**
Miconic-FullADL (150)	**2**	0
MPrime (35)	0	**3**
Mystery (30)	0	**1**
Pathways (30)	1	**2**
Philosophers (48)	0	**2**
Pipesworld-NoTankage (50)	0	**2**
Pipesworld-Tankage (50)	1	**5**
Schedule (150)	0	**1**
Storage (30)	**1**	0
Total	12	**25**

LM-heur solves twice as many tasks exclusively as base

Outline

Extended Landmark Generation

- Adapted previous procedures to SAS^+ planning
- Admit disjunctive landmarks
- Find additional landmarks through DTGs

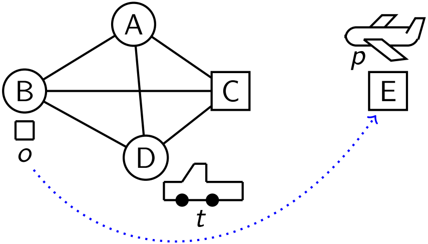
Domain Transition Graphs (DTGs)

The domain transition graph of $v \in \mathcal{V}$ (DTG$_v$) represents how the value of v can change

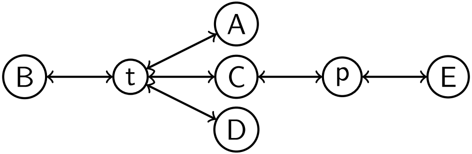
Given: a SAS^+ task $\langle \mathcal{V}, \mathcal{A}, s_0, s_\star \rangle$
DTG$_v$ is a directed graph with nodes $\mathcal{D}_v$ that has arc $\langle d, d' \rangle$ iff

- $d \neq d'$, and
- $\exists$ action with $v \mapsto d'$ as effect, and either
  - $v \mapsto d$ as precondition, or
  - no precondition on v

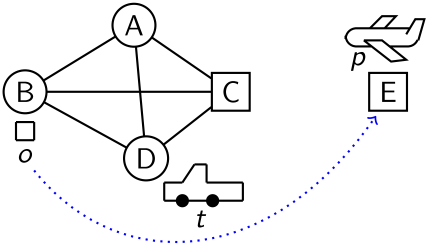
DTG Example



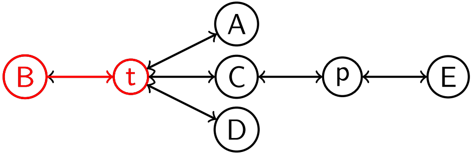
DTG for v_o:



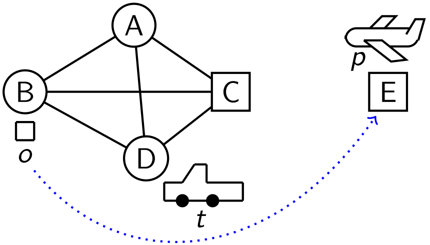
DTG Example



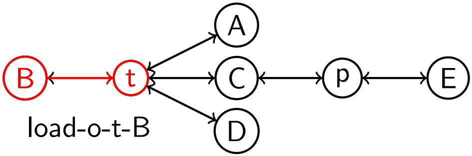
DTG for v_o:



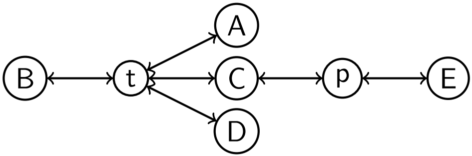
DTG Example



DTG for v_o:



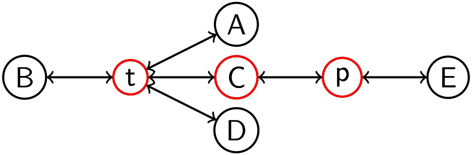
Extended Landmark Generation



- Find additional landmarks through DTGs: if
  - $s_0(v) = d_0$,
  - $v \mapsto d$ landmark, and
  - every path from d_0 to d passes through d',

  then $v \mapsto d'$ landmark
- No further improvement in % solved, but shorter plans

Extended Landmark Generation



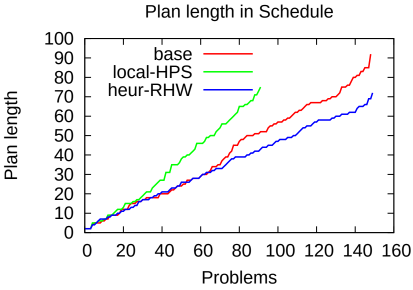
- Find additional landmarks through DTGs: if
    - $s_0(v) = d_0$,
    - $v \mapsto d$ landmark, and
    - every path from d_0 to d passes through d',

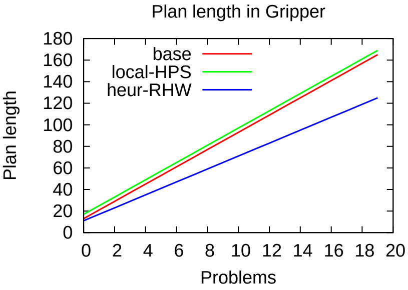
  then $v \mapsto d'$ landmark

- No further improvement in % solved, but shorter plans

Extended landmark generation – Plan length

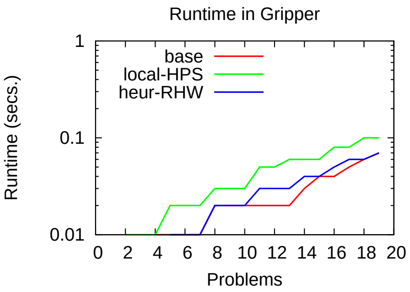
- Local: plans 6% longer than with base
- Heur: plans 1% shorter
- Heur with extended LMs: plans 3% shorter

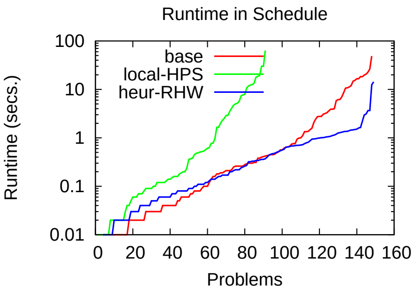




Remarks on Runtime

- LM generation usually < 1 sec. (max. 2 min.)
- During search: slight overhead through landmarks (≤ 18 %)
- Overhead typically outweighed by benefit in larger problems





Summary

- Landmark heuristic significantly improves existing heuristics
- More tasks solved
- Better quality of solutions (plan lengths)
- Complete, unlike previous local search approach
- First approach that handles disjunctive landmarks

Summary

- Landmark heuristic significantly improves existing heuristics
- More tasks solved
- Better quality of solutions (plan lengths)
- Complete, unlike previous local search approach
- First approach that handles disjunctive landmarks

Thank you!