An Admissible Heuristic for SAS$^+$ Planning Obtained from the State Equation

Blai Bonet

ICAPS. Rome, Italy. June 2013.

(to appear also in IJCAI-2013)

UNIVERSIDAD SIMÓN BOLÍVAR

Introduction

Domain-independent optimal planning $=$ A* $+$ heuristic

Most important heuristics are based on (Helmert & Domshlak, 2009):

- delete relaxation: hmax, FF, etc.
- abstractions: PDBs, structural patterns, M&S, etc.
- critical-path heuristics: h^m
- landmark heuristics: LA, LM-cut, etc

We present a new admissible heuristic that

- doesn't belong to such classes; in particular, isn't bounded by h^+
- it is competitive with LM-cut on some domains
- it offers a new framework for further enhancements

Reached Limit of Delete-Relaxation

Claim: *we have reached the limit of delete-relaxation heuristics for optimal planning*

Justifications:

- computing h^+ is NP-hard
- LM-cut approximates h^+ very well; on some domains, LM-cut $= h^+$
- LM-cut is the best (single) known heuristic (since 2009)
- known strenghtenings on LM-cut show marginal improvements and aren't cost effective

Need to go beyond the delete-relaxation!

Abstractions and Critical Paths

Abstraction and critical-path heuristics are not bounded by h^+

Have the potential to dominate others (Helmert & Domshlak, 2009)

This potential has not been met by methods such as

- structural patterns
- Merge-and-shrink (M&S)
- h^m for small $m = 1, 2$
- M&S based on bisimulations
- ....
- semi-relaxed heuristics don't yet perform well for optimal planning (Keyder, Hoffmann & Haslum, 2012)

Contribution

New admissible heuristic h^{SEQ} for optimal planning:

- it is not bounded (a priori) by h^+
- it is computed by solving an LP problem for each state s
- show how the base heuristic can be improved in different ways
- empirical comparison of heuristic across large number of benchmarks

AFAIK, idea was first suggested by Patrik Haslum during a tutorial on Petri Nets in ICAPS-2009

Flows

The heuristic tracks the **flow** (presence) of fluents across the application of actions in potential plans

If p is a **goal** fluent that is **not** initially true, then

$$\#\text{ times is “produced”} \quad - \quad \#\text{ times is “consumed”} \;>\; 0$$

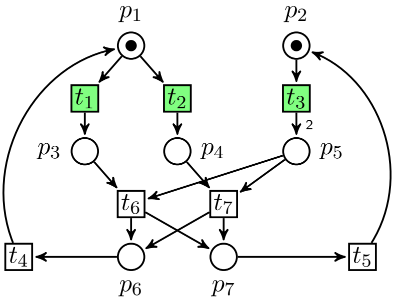
in any plan that solves the task

- fluent p is **produced by action** a if it is added or is prevail
- fluent p is **consumed by action** a if it is deleted or is prevail

Petri Nets

A P/T net is tuple $PN = \langle P, T, F, W, M_0 \rangle$ where

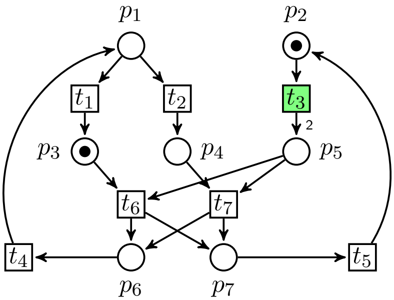
- $P = \{p_1, p_2, \ldots, p_m\}$ is set of places
- $T = \{t_1, t_2, \ldots, t_n\}$ is set of transitions
- $F \subseteq (P \times T) \cup (T \times P)$ is flow relation
- $W : F \to \mathbb{N}$ tells how many items flow in each arc of F
- $M_0 : P \to \mathbb{N}$ is initial marking



Petri Nets

A P/T net is tuple $PN = \langle P, T, F, W, M_0 \rangle$ where

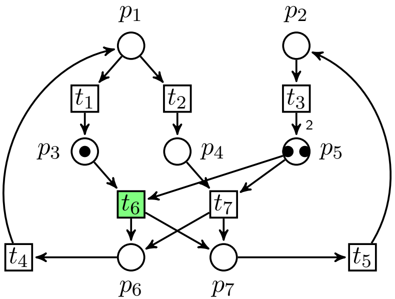
- $P = \{p_1, p_2, \ldots, p_m\}$ is set of places
- $T = \{t_1, t_2, \ldots, t_n\}$ is set of transitions
- $F \subseteq (P \times T) \cup (T \times P)$ is flow relation
- $W : F \to \mathbb{N}$ tells how many items flow in each arc of F
- $M_0 : P \to \mathbb{N}$ is initial marking



Petri Nets

A P/T net is tuple $PN = \langle P, T, F, W, M_0 \rangle$ where

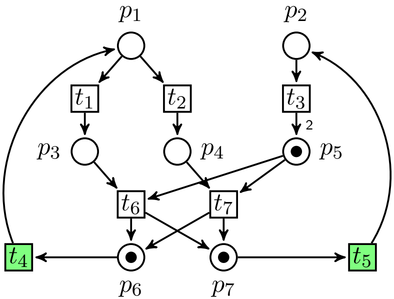
- $P = \{p_1, p_2, \ldots, p_m\}$ is set of places
- $T = \{t_1, t_2, \ldots, t_n\}$ is set of transitions
- $F \subseteq (P \times T) \cup (T \times P)$ is flow relation
- $W : F \to \mathbb{N}$ tells how many items flow in each arc of F
- $M_0 : P \to \mathbb{N}$ is initial marking



Petri Nets

A P/T net is tuple $PN = \langle P, T, F, W, M_0 \rangle$ where

- $P = \{p_1, p_2, \ldots, p_m\}$ is set of places
- $T = \{t_1, t_2, \ldots, t_n\}$ is set of transitions
- $F \subseteq (P \times T) \cup (T \times P)$ is flow relation
- $W : F \to \mathbb{N}$ tells how many items flow in each arc of F
- $M_0 : P \to \mathbb{N}$ is initial marking



State Equation

Incidence matrix A is $n \times m$ (transitions as rows, places as cols)
with entries $a_{ij} = W(t_i, p_j) - W(p_j, t_i)$

$$a_{i,j} = \text{“net change in number of tokens at } p_j \text{ caused by firing } t_i\text{”}$$

If when at marking M transition t_i fires, the result is marking M' where
$M'(p_j) = M(p_j) + a_{i,j}$ for every j

If when at marking M sequence $\sigma = u_1 \cdots u_\ell$ fires, the result is

$$M' \;=\; M + A^T \textstyle\sum_{k=1}^{\ell} u_k \;=\; M + A^T u$$

where u_k is an indicator vector whose i-th entry is 1 iff $u_k = t_i$

The vector $u = \sum_{k=1}^{\ell} u_k$ is called a **firing-count** vector

From SAS$^+$ to Petri Nets

SAS$^+$ problem $P = \langle V, A, s_{init}, s_G, c \rangle$

SAS$^+$ atoms are of the form '$X = x$' for variable X and $x \in D_X$

P/T net associated with problem P is $PN = \langle P, T, F, W, M_0 \rangle$ where

- places are atoms and transitions are actions
- F contains:
  - $(X = x, a)$ if *pre*$(a)[X] = x$ or $X = x$ is prevail
  - $(a, X = x)$ if *post*$(a)[X] = x$ or $X = x$ is prevail
- W assigns 1 to each arc in F
- M_0 is marking $M_{s_{init}}$ associated with state s_{init}

Def: for state s, marking M_s is such that $M_s(X = x) = 1$ iff $s[X] = x$

Necessary Conditions for Plan Existence

Reachable markings in PN **are not** in 1-1 correspondence to reachable states in P. However,

Theorem

Plan π is applicable at s_{init} only if π is a firing sequence at M_0.
If π reaches state s, then π reaches a marking M that covers M_s (i.e., $M_s \leq M$).

Let π be a plan for P; i.e., it reaches a goal state from s_{init}. Then,

$$A^T u_\pi \;=\; M_\pi - M_0 \;\geq\; M_s - M_0 \;\geq\; M_{s_G} - M_0$$

where u_π is firing-count vector for π and M_π is the marking reached by π.

SEQ Heuristic

h^{SEQ} assigns to state s the value $\lceil c^T x^* \rceil$ where x^* is solution of

$$\begin{array}{lrcl} \text{Minimize} & c^T x & & \\ \text{subject to} & A^T x & \geq & M_{s_G} - M_s \\ & x & \geq & 0\,, \end{array}$$

if LP is feasible, and ∞ if not. The case of unbounded solutions is not possible.

Theorem

h^{SEQ} is an admissible heuristic for SAS$^+$ planning.

Features of Heuristic

Strenghts:

- It can account for multiple applications of same action
- It is easy to improve by adding additional constraints

Weaknesses:

- Need to solve an LP for each state encountered during search
- Prevail conditions don't play an active role as they have zero net change

Improvements

Paper proposes three ways to improve the heuristic h^{SEQ}

- **Reformulations:** extend goal with fluents p that must hold concurrently with G. E.g., it happens in `airport` where coverage increases by 72.7% from 22 to 38 problems.

- **Safeness information:** promote inequalities $\geq$ to equalities in LP. It can be done for atoms in a **safe** set S: $p \in S$ implies $M(p) \leq 1$ for each reachable marking M. Safe sets S can computed directly at the planning problem.

- **Landmarks:** if $L = \{a_1, a_2, \ldots, a_k\}$ is an action landmark, then can add the constraint

$$x(a_1) + x(a_2) + \cdots + x(a_k) \geq 1$$

Experimental Results – Coverage I

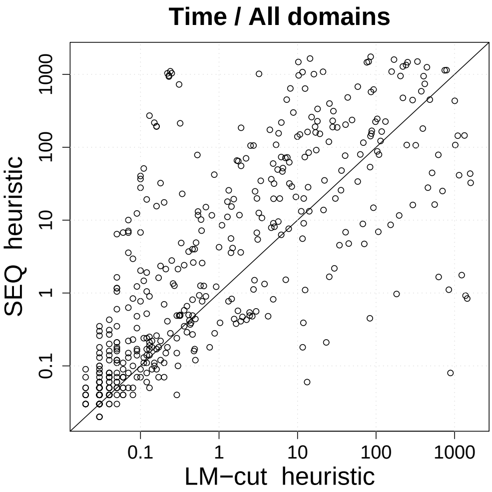
Domain	$h^{\text{LM-cut}}$	$h^{\text{LM-cut}}_{\text{ours}}$	h^{LA}	$h^{\text{M\&S}}$	HSP^*_F	h^{SEQ}	$h^{\text{SEQ}}_{\text{safe}}$
Airport (50)	**38**	35	24	16	15	22	23
Blocks (35)	28	28	20	18	**30**	28	28
Depot (22)	**7**	**7**	**7**	**7**	4	6	6
Driverlog (20)	**14**	**14**	**14**	12	9	11	11
Freecell (80)	15	15	28	15	20	**30**	**30**
Grid (5)	**2**	**2**	**2**	**2**	0	**2**	**2**
Gripper (20)	6	6	6	**7**	6	**7**	**7**
Logistics-2000 (28)	**20**	**20**	**20**	16	16	16	16
Logistics-1998 (35)	**6**	**6**	5	4	3	3	3
Miconic-STRIPS (150)	**140**	**140**	**140**	54	45	50	50
MPrime (35)	25	**24**	21	21	8	21	21
Mystery (19)	**17**	**17**	15	14	9	15	15
Openstacks-STRIPS (30)	**7**	**7**	**7**	**7**	**7**	**7**	**7**
Pathways (30)	**5**	**5**	4	3	4	4	4
Pipesworld-no-tankage (50)	17	17	17	**20**	13	15	15
Pipesworld-tankage (50)	11	11	9	**13**	7	9	9
PSR-small (50)	49	49	48	**50**	**50**	**50**	**50**
Rovers (40)	**7**	**7**	6	6	6	6	6
Satellite (36)	8	**9**	7	6	5	6	6
TPP (30)	6	6	6	6	5	**8**	**8**
Trucks (30)	**10**	9	7	6	9	**10**	**10**
Zenotravel (20)	**12**	**12**	9	11	8	9	9
Airport-modified (50)	na	36	na	na	na	**38**	**38**
Total (w/o Airport-modified)	**450**	446	422	314	279	335	336

Experimental Results – Coverage II

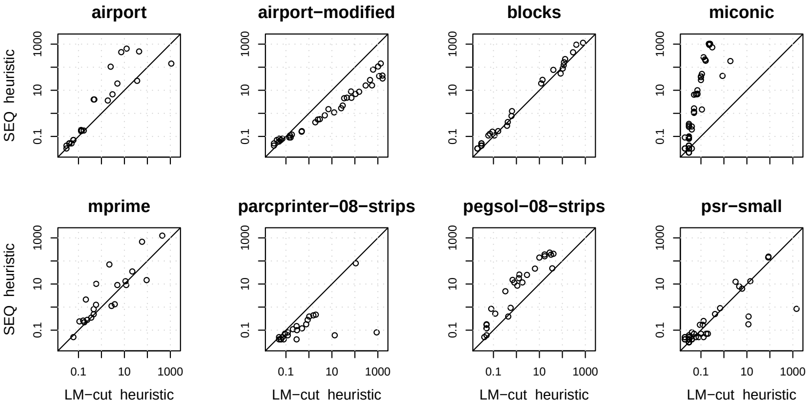
Domain	$h^{\text{LM-cut}}_{\text{ours}}$	h^{SEQ}	$h^{\text{SEQ}}_{\text{safe}}$
Elevators-08-STRIPS (30)	**19**	9	9
Openstacks-08-STRIPS (30)	**19**	16	16
Parcprinter-08-STRIPS (30)	22	**28**	**28**
Pegsol-08-STRIPS (30)	**27**	26	**27**
Scanalyzer-08-STRIPS (30)	**15**	12	12
Sokoban-08-STRIPS (30)	**28**	17	17
Transport-08-STRIPS (30)	**11**	9	9
Woodworking-08-STRIPS (30)	**15**	12	12
Total	**156**	129	130

Domains from IPC-08 that involve actions with different costs

Experimental Results – Time on All Domains

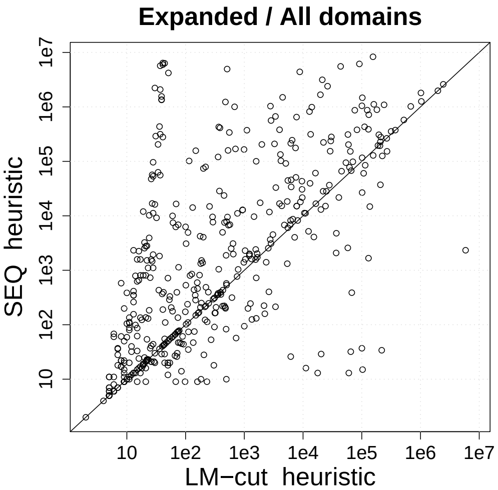


Experimental Results – Time on Selected Domains



Domains with at least 20 instances solved by the two heuristics

Experimental Results – Expansions on All Domains



Conclusions & Future Work

- Defined a new heuristic that is not bounded by h^+
- Vanilla flavor of heuristic is competitive with state-of-the-art heuristics on some domains
- Heuristic can be further improved; some proposals put on the table but need to be tested
- Interestingly, solving an LP for each node is not as bad as it sounds

Future work:

- Add constraints from landmarks
- Try dealing with prevail conditions by using **duplication**: if p is prevail for some action a, introduce two 'copies' of p, p and p', such that a consumes p and produces p'