Sequencing Operator Counts

Toby Davies
Adrian Pearce   Peter Stuckey   Nir Lipovetzky

The University of Melbourne and NICTA

9th June 2015

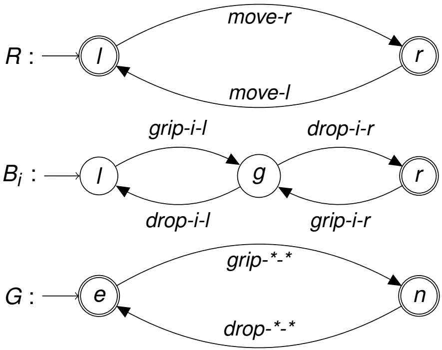
Australian Government
Department of Broadband, Communications and the Digital Economy
Australian Research Council

NICTA Funding and Supporting Members and Partners

The Planning Problem

Find a sequence of operators which:

- Satisfies multiple Domain Transition Graphs (DTGs).
- Has minimum cost.



Algorithms for planning

- Forward (and backward) state-based search
- Planning-as-SAT
- Partial-order planning

Planning-as-SAT

- Encode a number of time-indexed “layers” as a SAT formula.

$$\neg\texttt{holding-b1@t1} \Rightarrow \neg\texttt{drop-b1-in-left@t1}$$

$$\neg\texttt{b1-in-left@t1} \wedge \texttt{b1-in-left@t2} \Rightarrow \texttt{drop-b1-in-left@t1}$$

- Incrementally extend the formula as needed.
- How do you prove optimality?

Heuristic Search

- A* with a relaxation (heuristic) gives a LB.
- By expanding minimum LB state, we can prove optimality.
- How do you handle side constraints?

Operator Counting Heuristics

$$\text{minimize} \sum_{o \in O} c(o) \cdot Y_o$$

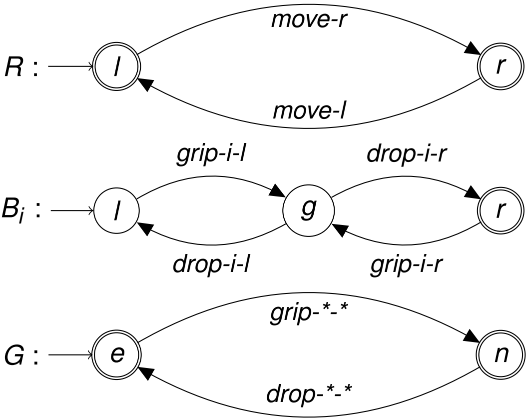
$$s.t.$$

$$\sum_{o \in LM} Y_o \geq 1 \qquad \forall LM$$

$$\sum_{o \in \text{prod}(p)} Y_o - \sum_{o \in \text{cons}(p)} Y_o = \Delta_p \qquad \forall p$$

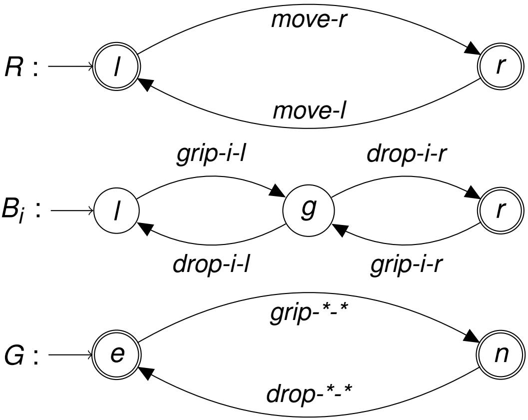
- Use a MIP with Y_o variables which count each operator o.
- Heuristics can be combined, often strictly dominating the components.
- The MIP solution gives a heuristic estimate; and
- **An assignment to the Y_o variables.**

Sequencing operator counts



- $\mathcal{C}(o)$ =
  grip-1-l: 1
  grip-2-l: 1
  move-l: 1
  move-r: 1
  drop-1-r: 1
  drop-2-r: 1
  otherwise: 0

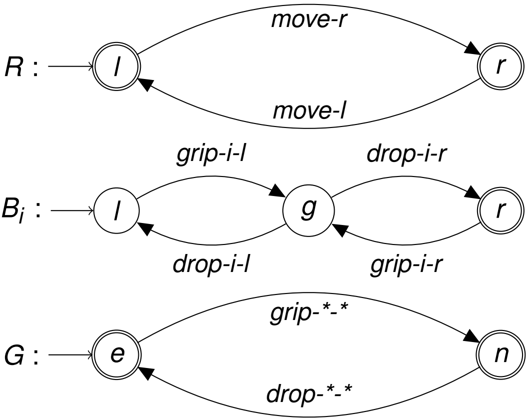
Sequencing operator counts



✗ $\mathcal{C}(o)$ =

grip-1-l: 1
grip-2-l: 1
move-l: 1
move-r: 1
drop-1-r: 1
drop-2-r: 1
otherwise: 0

Sequencing operator counts



✓ $\mathcal{C}(o)$ =
grip-1-l: 1
grip-2-l: 1
move-l: 1
move-r: 2
drop-1-r: 1
drop-2-r: 1
otherwise: 0

Landmarks

A (Disjunctive Action) Landmark is a necessary condition on the set of operators in a plan.

$$Y_1 + \cdots + Y_n \geq 1$$

or

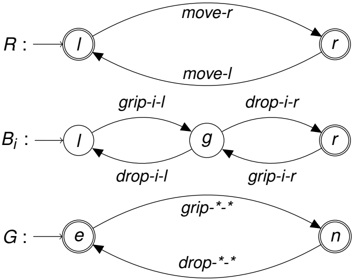
$$[Y_1 \geq 1] \vee \cdots \vee [Y_n \geq 1]$$

"at least one of these operators occurs at least one time"

Generalized Landmarks (GLMs)

We generalize this to:

$$[Y_1 \geq k_1] \vee \cdots \vee [Y_n \geq k_n]$$



The flaw we identified earlier:

$$[Y_{move\text{–}r} \geq 2]$$

Domain Constraints

Bounds literals ($[Y_o \geq k]$) are not built in to MIPs,
To define their relationship with the Y_o variables, we add:

$$[Y_o \geq k] \leq [Y_o \geq k-1]$$

$$Y_o \geq \sum_{i=1}^{\infty} [Y_o \geq i]$$

$$Y_o \leq M[Y_o \geq k] + k - 1$$

- $[Y_o \geq k] \Rightarrow [Y_o \geq k-1]$
- n bounds literals are set, then $Y_o \geq n$;
- if k or more operators occur, $[Y_o \geq k]$ must be set.

We then lazily create the bounds literals when they are mentioned in a GLM.

Theorem

There exists a set of generalized landmark constraints such that solving a MIP with these constraints will compute $h^*(s_0)$.

Proof.

With optimal operator count $\mathcal{C}$, either:

- We have found a plan projection; or
- We can add

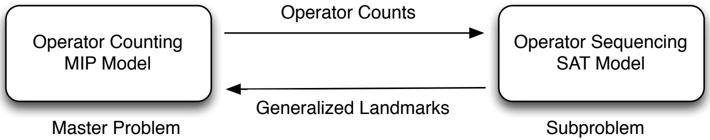
$$\sum_{o \in O} [Y_o \geq \mathcal{C}(o) + 1] \geq 1$$

  and re-optimise to get a new count.

□

Logic-Based Benders Decomposition

$$\mathcal{C} : O \rightarrow \mathbb{N}$$



$$\sum_{o \in L} [Y_o \geq \mathcal{C}(o) + 1] \geq 1$$

Using SAT to get better cuts

We use the at-most-k constraint $\leq_k$ encoded into SAT.
Add **assumptions** to SAT-planning model for each upper-bound $Y_o \leq \mathcal{C}(o)$:

$$\neg[Y_o \geq \mathcal{C}(o) + 1]$$

When UNSAT is proved, the solver identifies a subset of the assumptions responsible for failure.

Using SAT to get better cuts

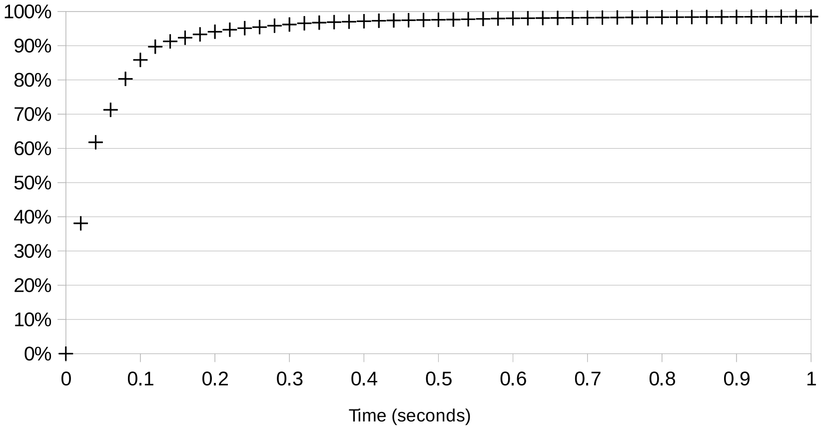
$$[Y_{grip-1-r} \geq 1] + [Y_{drop-1-l} \geq 1] + [Y_{grip-1-l} \geq 2] + [Y_{drop-1-r} \geq 2] +$$
$$[Y_{grip-2-l} \geq 2] + [Y_{drop-2-r} \geq 2] + [Y_{grip-2-r} \geq 1] + [Y_{drop-2-l} \geq 1] +$$
$$[Y_{move-l} \geq 2] + [Y_{move-r} \geq 2] \geq 1$$

vs

$$[Y_{grip-1-r} \geq 1] + [Y_{drop-1-l} \geq 1] + [Y_{move-r} \geq 2] + [Y_{drop-2-l} \geq 1] +$$
$$[Y_{grip-2-r} \geq 1] + [Y_{move-l} \geq 2] + [Y_T \geq 7] \geq 1$$

NB: Y_T is the count of a "fake" operator T: the total operator count.

Generating GLMs is surprisingly efficient



Benchmark	*OpSeq* C	*OpSeq* =	*OpSeq* Q	*Hpp* C	*Hpp* =	*Hpp* Q	*SymBA*-2* C	*SymBA*-2* =	*SymBA*-2* Q
barman	0	0	9.37	0	0	9.14	**11**	**20**	**20.00**
elevators	11	11	19.38	0	0	16.47	**19**	**20**	**20.00**
nomystery	5	10	18.33	5	8	8.00	**15**	**18**	**19.82**
openstacks	0	0	5.52	0	0	5.52	**20**	**20**	**20.00**
parcprinter	**20**	**20**	**20.00**	**20**	**20**	**20.00**	17	17	18.63
pegsol	2	5	15.97	0	0	12.43	**19**	**20**	**20.00**
scanalyzer	1	3	7.99	3	**14**	**18.93**	**9**	10	14.32
sokoban	0	2	10.70	1	2	11.27	**20**	**20**	**20.00**
transport	5	13	**19.47**	0	0	12.41	**11**	**14**	17.81
visitall	**14**	**20**	**20.00**	5	13	19.21	12	12	15.70
woodworking	**20**	**20**	**20.00**	18	18	19.95	19	19	19.74
Total	78	104	166.74	52	75	153.33	**172**	**189**	**206.02**

Coverage (C)

Number of best bounds (=)

Dual quality scores (Q)

Conclusion and Future Work

This is a fundamentally new approach to planning, splitting planning into an operator counting problem, and a sequencing problem.

Any **explaining** constraint or theory can be added to the sub-problem, and can be re-written into the assumptions

This has applications in:

- Temporal planning.
- Planning with resources.
- Hybrid planning/scheduling problems.