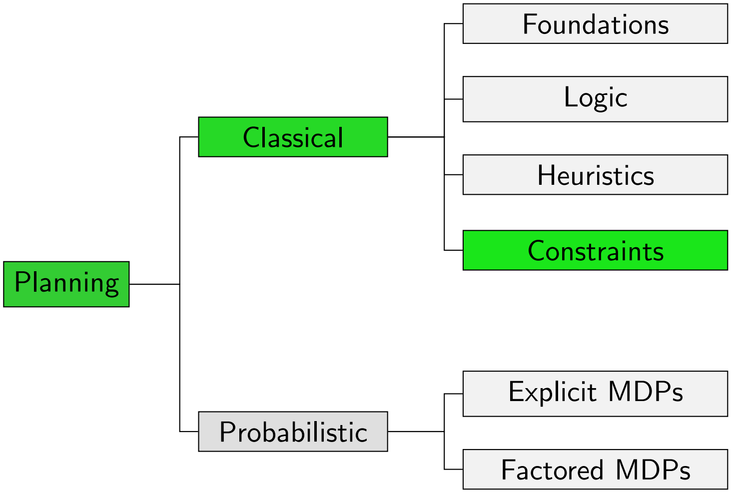
Planning and Optimization

E8. Operator Counting

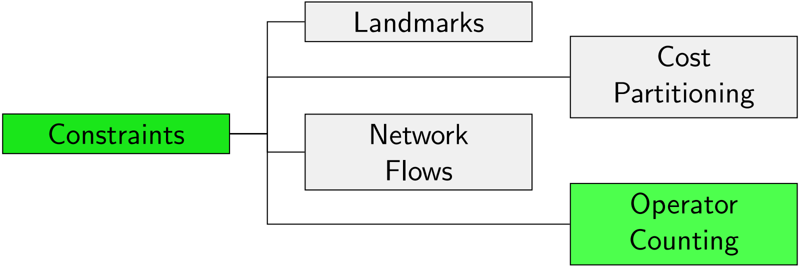
Malte Helmert and Gabriele Röger

Universität Basel

Content of this Course



Content of this Course: Constraints



Introduction

Reminder: Flow Heuristic

In the previous chapter, we used flow constraints to describe how often operators must be used in each plan.

Example (Flow Constraints)

Let Π be a planning problem with operators $\{o_{\text{red}}, o_{\text{green}}, o_{\text{blue}}\}$.
The flow constraint for some atom a is the constraint

$$1 + Count_{o_{\text{green}}} = Count_{o_{\text{red}}}.$$

In natural language, this flow constraint expresses that

Reminder: Flow Heuristic

In the previous chapter, we used flow constraints to describe how often operators must be used in each plan.

Example (Flow Constraints)

Let Π be a planning problem with operators $\{o_{\text{red}}, o_{\text{green}}, o_{\text{blue}}\}$.
The flow constraint for some atom a is the constraint

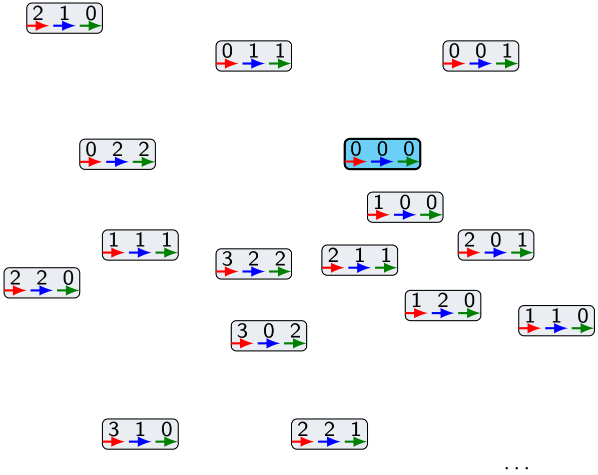
$$1 + \mathit{Count}_{o_{\text{green}}} = \mathit{Count}_{o_{\text{red}}}.$$

In natural language, this flow constraint expresses that

every plan uses o_{red} once more than o_{green}.

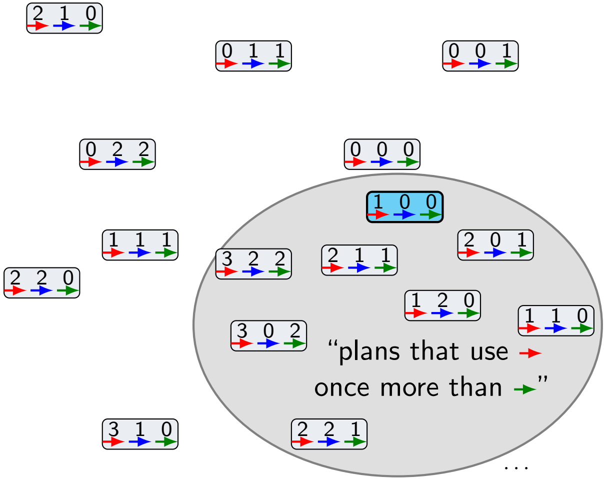
Reminder: Flow Heuristic

Let us now observe how each flow constraint alters the operator count solution space.



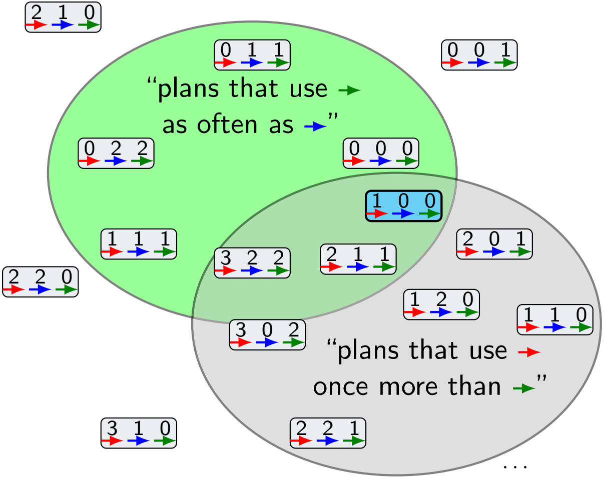
Reminder: Flow Heuristic

Let us now observe how each flow constraint alters the operator count solution space.



Reminder: Flow Heuristic

Let us now observe how each flow constraint alters the operator count solution space.



Operator-counting Framework

Operator Counting

Operator counting

- generalizes this idea to a framework that allows to admissibly combine different heuristics.
- uses linear constraints ...
- ... that describe number of occurrences of an operator ...
- ... and must be satisfied by every plan.
- provides declarative way to describe knowledge about solutions.
- allows reasoning about solutions to derive heuristic estimates.

Operator-counting Constraint

Definition (Operator-counting Constraints)

Let Π be a planning task with operators O and let s be a state. Let $\mathcal{V}$ be the set of integer variables Count_o for each $o \in O$.

A linear inequality over $\mathcal{V}$ is called an operator-counting constraint for s if for every plan π for s setting each Count_o to the number of occurrences of o in π is a feasible variable assignment.

Operator-counting Heuristics

Definition (Operator-counting IP/LP Heuristic)

The operator-counting integer program IP_C for a set C of operator-counting constraints for state s is

$$\text{Minimize} \quad \sum_{o \in O} cost(o) \cdot \mathrm{Count}_o \quad \text{subject to}$$

$$C \text{ and } \mathrm{Count}_o \geq 0 \text{ for all } o \in O,$$

where O is the set of operators.

The IP heuristic h_C^{IP} is the objective value of IP_C,
the LP heuristic h_C^{LP} is the objective value of its LP-relaxation.

If the IP/LP is infeasible, the heuristic estimate is ∞.

Operator-counting Constraints

- Adding more constraints can only remove feasible solutions
- Fewer feasible solutions can only increase objective value
- Higher objective value means better informed heuristic

Operator-counting Constraints

- Adding more constraints can only remove feasible solutions
- Fewer feasible solutions can only increase objective value
- Higher objective value means better informed heuristic

Are there operator-counting constraints other than flow constraints?

Reminder: Minimum Hitting Set for Landmarks

Variables

Non-negative variable Applied_o for each operator o

Objective

Minimize $\sum_o cost(o) \cdot \text{Applied}_o$

Subject to

$$\sum_{o \in L} \text{Applied}_o \geq 1 \text{ for all landmarks } L$$

Operator Counting with Disjunctive Action Landmarks

Variables

Non-negative variable Count_o for each operator o

Objective

Minimize $\sum_o cost(o) \cdot \mathrm{Count}_o$

Subject to

$$\sum_{o \in L} \mathrm{Count}_o \geq 1 \text{ for all landmarks } L$$

New: Post-hoc Optimization Constraints

For set of abstractions $\{\alpha_1, \ldots, \alpha_n\}$:

Variables

Non-negative variables Count_o for all operators $o \in O$

$\mathrm{Count}_o \cdot \mathit{cost}(o)$ is cost incurred by operator o

Objective

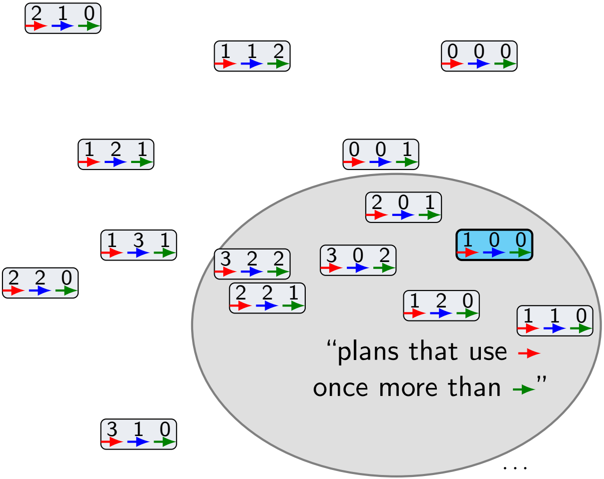
Minimize $\sum_{o \in O} \mathit{cost}(o) \cdot \mathrm{Count}_o$

Subject to

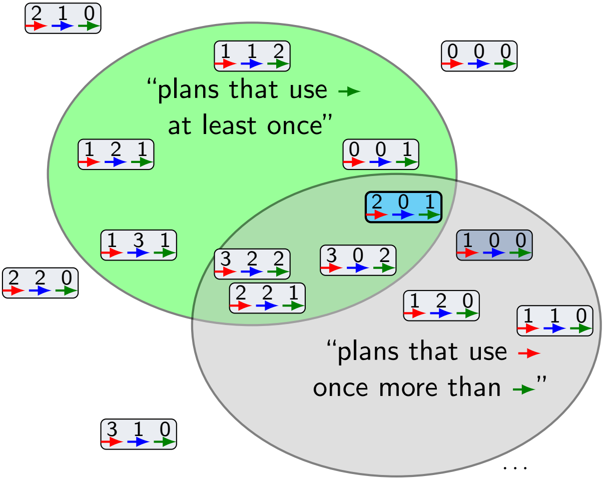
$$\sum_{o \in O : o \text{ affects } \mathcal{T}^\alpha} \mathit{cost}(o) \cdot \mathrm{Count}_o \geq h^\alpha(s) \quad \text{for } \alpha \in \{\alpha_1, \ldots, \alpha_n\}$$

$$\mathit{cost}(o) \cdot \mathrm{Count}_o \geq 0 \quad \text{for all } o \in O$$

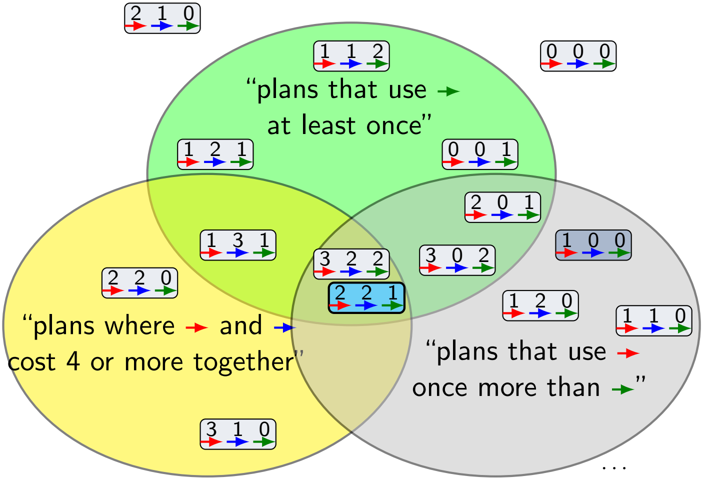
Example


2 1 0
1 1 2
0 0 0
1 2 1
0 0 1
2 0 1
1 3 1
1 0 0
3 2 2
3 0 2
2 2 0
2 2 1
1 2 0
1 1 0
"plans that use once more than"
3 1 0
...

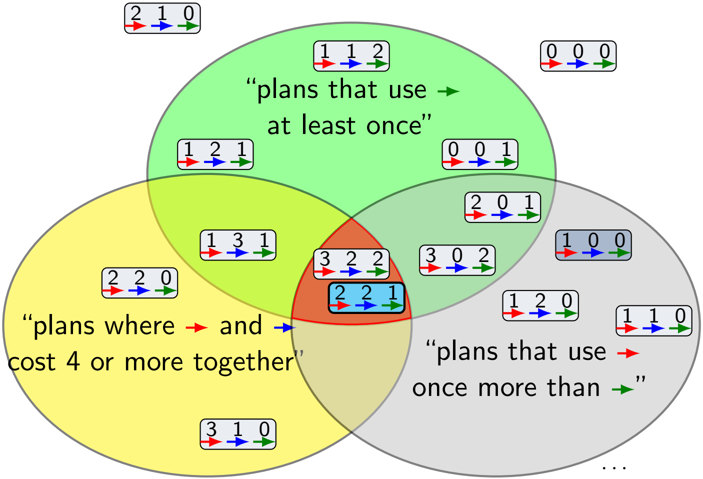
Example


2 1 0
1 1 2
0 0 0
"plans that use
at least once"
1 2 1
0 0 1
2 0 1
1 3 1
1 0 0
3 2 2
3 0 2
2 2 0
2 2 1
1 2 0
1 1 0
"plans that use
once more than "
3 1 0
...

Example


2 1 0
1 1 2
0 0 0
"plans that use at least once"
1 2 1
0 0 1
2 0 1
1 3 1
1 0 0
3 2 2
3 0 2
2 2 0
2 2 1
1 2 0
1 1 0
"plans where and cost 4 or more together"
"plans that use once more than"
3 1 0
...

Example


2 1 0
1 1 2
0 0 0
"plans that use → at least once"
1 2 1
0 0 1
2 0 1
1 3 1
1 0 0
3 2 2
3 0 2
2 2 0
2 2 1
1 2 0
1 1 0
"plans where → and → cost 4 or more together"
"plans that use → once more than →"
3 1 0
. . .

Further Examples?

- The definition of operator-counting constraints can be extended to groups of constraints and auxiliary variables.
- With this extended definition we could also cover more heuristics, e.g., the perfect delete-relaxation heuristic h^+.

Properties

Admissibility

Theorem (Operator-counting Heuristics are Admissible)

The IP and the LP heuristic are admissible.

Proof.

Let C be a set of operator-counting constraints for state s and π be an optimal plan for s. The number of operator occurrences of π are a feasible solution for C. As the IP/LP minimizes the total plan cost, the objective value cannot exceed the cost of π and is therefore an admissible estimate. □

Dominance

Theorem

Let C and C' be sets of operator-counting constraints for s and let $C \subseteq C'$. Then $\mathrm{IP}_C \leq \mathrm{IP}_{C'}$ and $\mathrm{LP}_C \leq \mathrm{LP}_{C'}$.

Proof.

Every feasible solution of C' is also feasible for C. As the LP/IP is a minimization problem, the objective value subject to C can therefore not be larger than the one subject to C'. □

Adding more constraints can only improve the heuristic estimate.

Heuristic Combination

Operator counting as heuristic combination

- Multiple operator-counting heuristics can be combined by computing $h_{\mathcal{C}}^{\mathrm{LP}} / h_{\mathcal{C}}^{\mathrm{IP}}$ for the union of their constraints.
- This is an admissible combination.
  - Never worse than maximum of individual heuristics
  - Sometimes even better than their sum
- We already know a way of admissibly combining heuristics: cost partitioning.

  $\Rightarrow$ How are they related?

Connection to Cost Partitioning

Theorem

Let $C_1, \ldots, C_n$ be sets of operator-counting constraints for s and $\mathcal{C} = \bigcup_{i=1}^{n} C_i$. Then $h^{\mathrm{LP}}_{\mathcal{C}}$ is the optimal general cost partitioning over the heuristics $h^{\mathrm{LP}}_{C_i}$.

Proof ommitted.

Comparison to Optimal Cost Partitioning

- some heuristics are more compact
  if expressed as operator counting
- some heuristics cannot be expressed as operator counting
- operator counting IP even better than
  optimal cost partitioning
- Cost partitioning maximizes, so heuristics must be encoded perfectly to guarantee admissibility.
  Operator counting minimizes, so missing information just makes the heuristic weaker.

Summary

Summary

- Many heuristics can be formulated in terms of operator-counting constraints.
- The operator counting heuristic framework allows to combine the constraints and to reason on the entire encoded declarative knowledge.
- The heuristic estimate for the combined constraints can be better than the one of the best ingredient heuristic but never worse.
- Operator counting is equivalent to optimal general cost partitioning over individual constraints.

Literature (1)

Florian Pommerening, Gabriele Röger and Malte Helmert.
Getting the Most Out of Pattern Databases for Classical Planning.
Proc. IJCAI 2013, pp. 2357–2364, 2013.
Introduces post-hoc optimization and points out relation to canonical heuristic.

Blai Bonet.
An Admissible Heuristic for SAS+ Planning Obtained from the State Equation.
Proc. IJCAI 2013, pp. 2268–2274, 2013.
Suggests combination of flow constraints and landmark constraints.

Literature (2)

- Tatsuya Imai and Alex Fukunaga.
  A Practical, Integer-linear Programming Model for the Delete-relaxation in Cost-optimal Planning.
  Proc. ECAI 2014, pp. 459–464, 2014.
  IP formulation of h^+.

- Florian Pommerening, Gabriele Röger, Malte Helmert and Blai Bonet.
  LP-based Heuristics for Cost-optimal Planning.
  Proc. ICAPS 2014, pp. 226–234, 2014.
  Systematic introduction of operator-counting framework.