LP-based Heuristics for Cost-optimal Classical Planning

3. Operator Counting

Florian Pommerening   Gabriele Röger   Malte Helmert

ICAPS 2015 Tutorial

June 7, 2015

Operator-counting Framework

Operator Counting

Reminder:

Idea 2: Operator Counting Constraints

- linear constraints whose variables denote number of occurrences of a given operator
- must be satisfied by every plan that solves the task

Examples:

- $Y_{o_1} + Y_{o_2} \geq 1$ "must use o_1 or o_2 at least once"
- $Y_{o_1} - Y_{o_3} \leq 0$ "cannot use o_1 more often than o_3"

Motivation:

- declarative way to represent knowledge about solutions
- allows reasoning about solutions to derive heuristic estimates

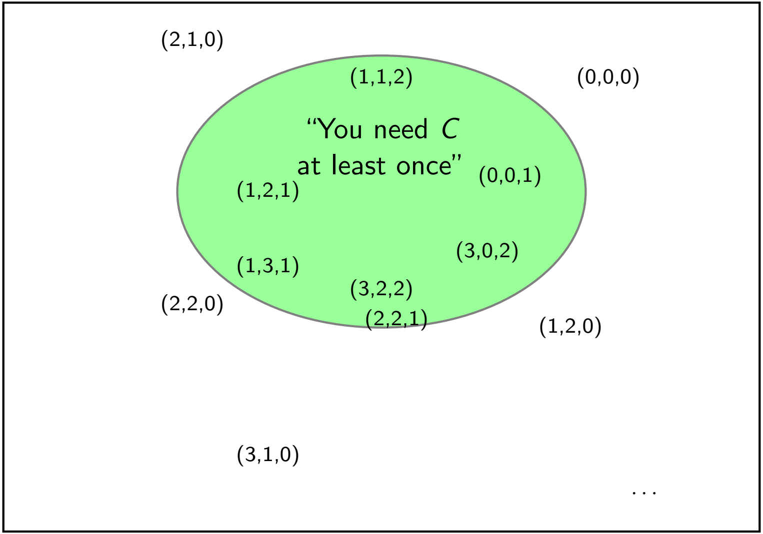
Operator Counting Heuristics

Operator occurrences in potential plans

(2,1,0)
(1,1,2)
(0,0,0)
(0,0,1)
(1,2,1)
(3,0,2)
(1,3,1)
(3,2,2)
(2,2,0)
(2,2,1)
(1,2,0)
(3,1,0)
...

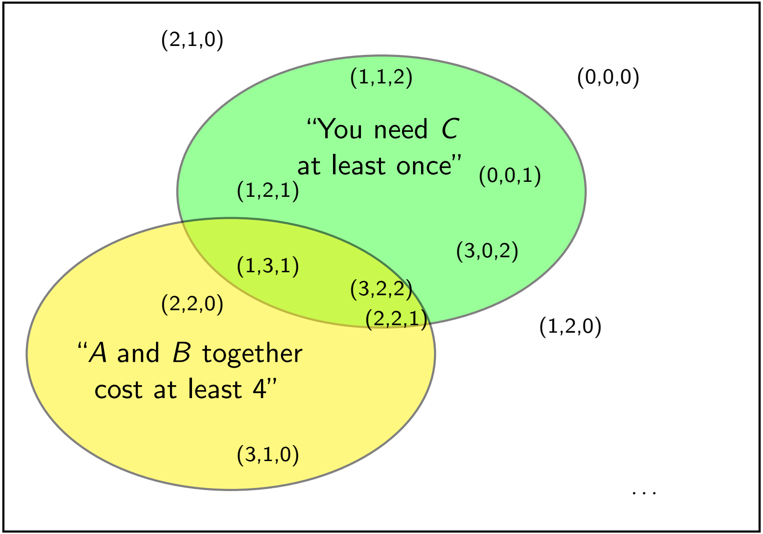
Operator Counting Heuristics

Operator occurrences in potential plans



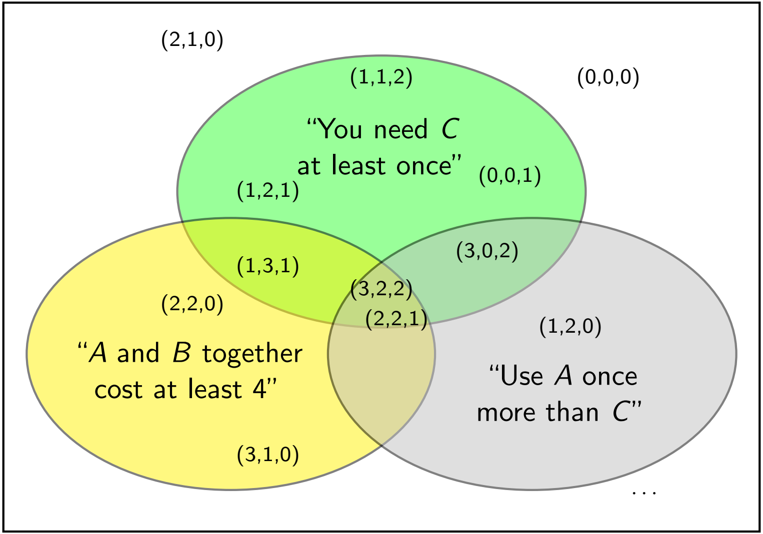
Operator Counting Heuristics

Operator occurrences in potential plans



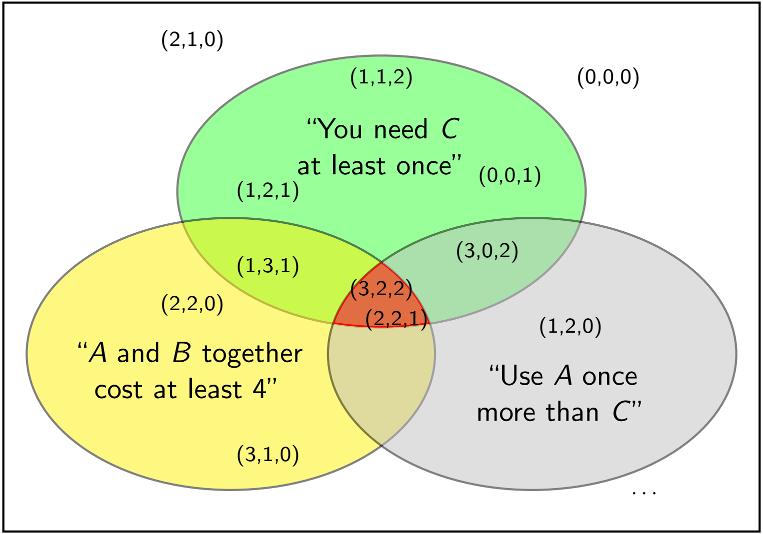
Operator Counting Heuristics

Operator occurrences in potential plans



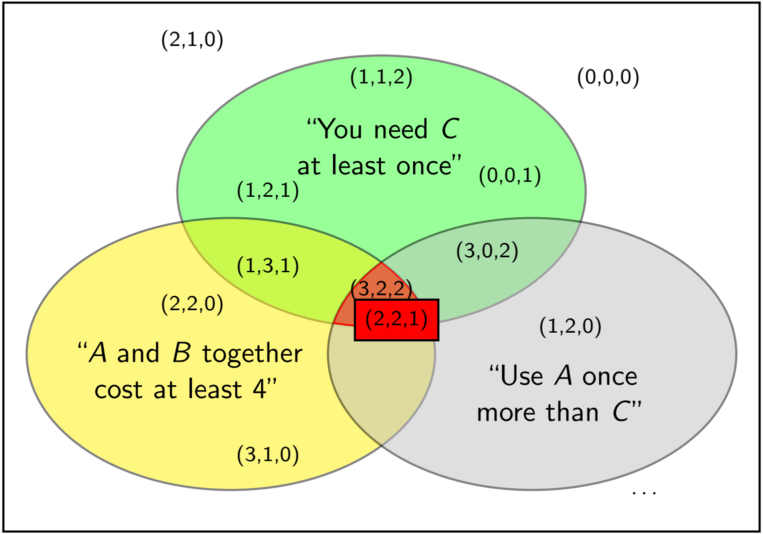
Operator Counting Heuristics

Operator occurrences in potential plans



Operator Counting Heuristics

Operator occurrences in potential plans



Operator-counting Heuristics

Operator-counting IP/LP Heuristic

$$\text{Minimize} \sum_{o} Y_o \cdot cost(o) \text{ subject to}$$

$Y_o \geq 0$ and some operator-counting constraints

Operator-counting constraint

- Set of linear inequalities
- For every plan π there is an LP-solution where Y_o is the number of occurrences of o in π .

Properties of Operator-counting Heuristics

Admissibility

Operator-counting (IP and LP) heuristics are admissible.

Computation time

Operator-counting LP heuristics are solvable in polynomial time.

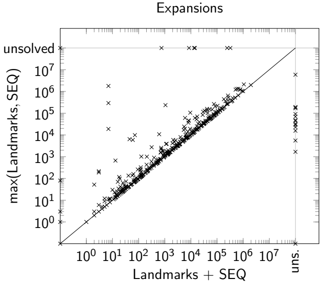
Adding constraints

Adding constraints can only make the heuristic more informed.

Combining heuristics

Combination of two heuristics

- Use both operator-counting constraints
- Combination always dominates individual heuristics
- Positive interaction between constraints



Examples

Example 1: Disjunctive Action Landmarks

We have already seen one example

- Optimal cost partitioning for landmarks
- Use one landmark constraint per landmark

Landmark constraint for landmark L

$$\sum_{o \in L} Y_o \geq 1$$

Connection to cost partitioning is no coincidence

- More details later

Example 2: Post-hoc Optimization Heuristic

Post-hoc optimization heuristic (Pommerening et al. 2013)

- Some operators are irrelevant for a heuristic computation
  - E.g., operator does not affect variables in projection
- Admissible heuristic is a lower bound for the cost induced by relevant operators

Post-hoc optimization constraint for heuristic h

$$\sum_{o \text{ is relevant for } h} Y_o cost(o) \geq h(s)$$

Example 3: State-equation Heuristic

Also known as

- Order-relaxation heuristic (van den Briel et al. 2007)
- State-equation heuristic (Bonet 2013)
- Flow-based heuristic (van den Briel and Bonet 2014)

Main idea:

- Facts can be produced (made true) or consumed (made false) by an operator
- Number of producing and consuming operators must balance out for each fact

Example 3: State-equation Heuristic

Net-change constraint for fact f

$$G(f) - S(f) = \sum_{f \in eff(o)} Y_o - \sum_{f \in pre(o)} Y_o$$

Remark:

- Assumes transition normal form (not a limitation)
  - Operator mentions same variables in precondition and effect
  - General form of constraints more complicated

⇝ presentation: Tuesday, first afternoon session

Fluent Merging in the State Equation Heuristic

- State equation heuristic has one constraint per fact
  - Only limited interaction with other variables captured

Solution: Fluent merging (van den Briel et al. 2007)

- Consider tuple of facts as one new fact
- Dynamic merging strategy (van den Briel and Bonet 2014)
  - Repeatedly remove inaccuracies in LP solution

Example 4: Relaxed planning heuristic

Relaxed Planning

- Ignores all delete effects
- No operator needed more than once

Can be expressed as operator-counting constraint
(Imai and Fukunaga 2014)

- Binary variables U_o express if an operator is used
- Connection to operator-counting variables: $U_o \leq Y_o$
- Encoding uses time steps for first achievers

Connection to Cost Partitioning

Operator-counting Heuristics and General Cost Partitioning

Theorem (Pommerening et al. 2015)

Combining operator-counting heuristics in one LP
is equivalent to
computing their optimal general cost partitioning.

Use the Theorem to Combine Heuristics

- Easy way to compute cost partitioning of heuristics
  - LP can be more compact (variable elimination)
  - No need for one variable per operator and subproblem
- Even better combination of heuristics with IP heuristic
  - Considers that operator cannot be used 1.5 times
  - But computation is no longer polynomial

Use the Theorem to Analyze Heuristics

Analyze operator counting heuristics

1. Group linear constraints into operator-counting constraints
2. Figure out what heuristic is computed with just one such operator-counting constraint
3. Your original operator-counting heuristic computes the optimal general cost partition of those component heuristics

Use the Theorem to Analyze Heuristics

Analyze operator counting heuristics

Example: state equation heuristic

1. Group linear constraints into operator-counting constraints
2. Figure out what heuristic is computed with just one such operator-counting constraint
3. Your original operator-counting heuristic computes the optimal general cost partition of those component heuristics

Use the Theorem to Analyze Heuristics

Analyze operator counting heuristics

Example: state equation heuristic

1. Group linear constraints into operator-counting constraints
   - One group of net change constraints per variable
2. Figure out what heuristic is computed with just one such operator-counting constraint
3. Your original operator-counting heuristic computes the optimal general cost partition of those component heuristics

Use the Theorem to Analyze Heuristics

Analyze operator counting heuristics

Example: state equation heuristic

1. Group linear constraints into operator-counting constraints
   - One group of net change constraints per variable
2. Figure out what heuristic is computed with just one such operator-counting constraint
   - Minimizing total cost while respecting flow in projection to one variable
   - Shortest path in projection
3. Your original operator-counting heuristic computes the optimal general cost partition of those component heuristics

Use the Theorem to Analyze Heuristics

Analyze operator counting heuristics

Example: state equation heuristic

1. Group linear constraints into operator-counting constraints
   - One group of net change constraints per variable
2. Figure out what heuristic is computed with just one such operator-counting constraint
   - Minimizing total cost while respecting flow in projection to one variable
   - Shortest path in projection
3. Your original operator-counting heuristic computes the optimal general cost partition of those component heuristics
   - State equation heuristic = gOCP(atomic projection heuristics)

Other Examples

What about the rest of our examples?

- Landmark constraints
  - gOCP(individual landmark heuristics)
- Post-hoc optimization heuristic
  - gOCP(heuristics that spend a minimum cost on relevant ops)
  - Also: cost partitioning over atomic projection heuristics
    - Operator costs not independent
    - Scale with one factor per projection
- Relaxed planning heuristic
  - Still unclear. Constraints cannot be easily factored (variables for time steps tie everything together)

Using Operator Counts

Using Operator Counts

- Operator counting so far:
  - Set up one LP
  - Solve it for each state encountered during search
  - Use estimated plan cost as heuristic value
  - Discard all other information
- Individual operator counts carry important information as well
- Sequencing Operator Counts (Davies et al. 2015)
  - Try to bring operators in applicable sequence
  - If that fails, generate new constraints
  - Best Paper Award

  ⇝ presentation: Tuesday, first afternoon session

Tutorial Structure

1. ~~Introduction and Overview~~
2. ~~Cost Partitioning~~
3. ~~Operator Counting~~
4. Potential Heuristics