LP-based Heuristics for Cost-optimal Classical Planning

1. Introduction and Overview

Florian Pommerening   Gabriele Röger   Malte Helmert

ICAPS 2015 Tutorial

June 7, 2015

Background: Linear Programs

Linear Programs and Integer Programs

Linear Program

A linear program (LP) consists of:

- a finite set of real-valued variables V
- a finite set of linear inequalities (constraints) over V
- an objective function, which is a linear combination of V
- which should be minimized or maximized.

Integer program (IP): ditto, but with integer-valued variables

Linear Program: Example

Example:

$$\begin{aligned} \text{maximize} \quad & 2x - 3y + z \quad \text{subject to} \\ & x + 2y + z \leq 10 \\ & x - z \leq 0 \\ & x \geq 0, \quad y \geq 0, \quad z \geq 0 \end{aligned}$$

$\rightsquigarrow$ unique optimal solution:
$x = 5$, $y = 0$, $z = 5$ (objective value 15)

Solving Linear Programs and Integer Programs

Complexity:

- LP solving is a polynomial-time problem.
- Finding solutions for IPs is NP-complete.

Common idea:

- Approximate IP solution with corresponding LP (LP relaxation).

Three Key Ideas in This Tutorial

Cost Partitioning

Idea 1: Cost Partitioning

- create copies $\Pi_1, \dots, \Pi_n$ of planning task Π
- each has its own operator cost function $cost_i$ (otherwise identical to Π)
- for all o: require $cost_1(o) + \dots + cost_n(o) \leq cost(o)$

⇝ sum of solution costs in copies is admissible heuristic: $h^*_{\Pi_1} + \dots + h^*_{\Pi_n} \leq h^*_{\Pi}$

Motivation:

- method for obtaining additive admissible heuristics
- very general and powerful

Operator Counting Constraints

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

Potential Heuristics

Idea 3: Potential Heuristics

Heuristic design as an optimization problem:

- Define simple numerical state features $f_1, \dots, f_n$.
- Consider heuristics that are linear combinations of features:

$$h(s) = w_1 f_1(s) + \cdots + w_n f_n(s)$$

  with weights (potentials) $w_i \in \mathbb{R}$
- Find potentials for which h is admissible and well-informed.

Motivation:

- declarative approach to heuristic design
- heuristic very fast to compute if features are

Cost Partitioning

Cost Partitioning

Idea 1: Cost Partitioning

- create copies $\Pi_1, \dots, \Pi_n$ of planning task Π
- each has its own operator cost function $cost_i : \mathcal{O} \to \mathbb{R}_0^+$ (otherwise identical to Π)
- for all o: require $cost_1(o) + \dots + cost_n(o) \leq cost(o)$

⇝ sum of solution costs in copies is admissible heuristic: $h^*_{\Pi_1} + \dots + h^*_{\Pi_n} \leq h^*_{\Pi}$

Cost Partitioning

- for admissible heuristics $h_1, \ldots, h_n$,
  $h(s) = h_{1,\Pi_1}(s) + \cdots + h_{n,\Pi_n}(s)$
  is an admissible estimate
- $h(s)$ can be better or worse than any $h_{i,\Pi}(s)$
  $\rightarrow$ depending on cost partitioning
- strategies for defining cost-functions
  - uniform: $cost_i(o) = cost(o)/n$
  - zero-one: full operator cost in one copy, zero in all others
  - ...

Can we find an optimal cost partitioning?

Optimal Cost Partitioning

Optimal Cost Partitioning

Optimal Cost Partitioning with LPs

- Use variables for cost of each operator in each task copy
- Express heuristic values with linear constraints
- Maximize sum of heuristic values subject to these constraints

LPs known for

- abstraction heuristics
- landmark heuristic

Optimal Cost Partitioning for Abstractions

Abstractions

- Simplified versions of the planning task, e.g. projections
- Cost of optimal abstract plan is admissible estimate

How to express the heuristic value as linear constraints?

Optimal Cost Partitioning for Abstractions

Abstractions

- Simplified versions of the planning task, e.g. projections
- Cost of optimal abstract plan is admissible estimate

How to express the heuristic value as linear constraints?
$\rightsquigarrow$ Shortest path problem in abstract transition system

LP for Shortest Path in State Space

Variables

Distance_s for each state s,
GoalDist

Objective

Maximize GoalDist

Subject to

$$\begin{aligned} \text{Distance}_{s_I} &= 0 && \text{for the initial state } s_I \\ \text{Distance}_{s'} &\leq \text{Distance}_s + cost(o) && \text{for all transition } s \xrightarrow{o} s' \\ \text{GoalDist} &\leq \text{Distance}_{s_\star} && \text{for all goal states } s_\star \end{aligned}$$

Optimal Cost Partitioning for Abstractions I

Variables

For each abstraction α:
$\text{Distance}^{\alpha}_{s}$ for each abstract state s,
cost^{α}_{o} for each operator o,
GoalDist^{α}

Objective

Maximize $\sum_{\alpha} \text{GoalDist}^{\alpha}$

...

Optimal Cost Partitioning for Abstractions II

Subject to

for all operators o

$$\sum_\alpha \mathrm{Cost}_o^\alpha \leq cost(o)$$

$$\mathrm{Cost}_o^\alpha \geq 0 \qquad \text{for all abstractions } \alpha$$

and for all abstractions α

$$\mathrm{Distance}_{s_I}^\alpha = 0 \qquad \text{for the abstract initial state } s_I$$

$$\mathrm{Distance}_{s'}^\alpha \leq \mathrm{Distance}_s^\alpha + \mathrm{Cost}_o^\alpha \qquad \text{for all transition } s \xrightarrow{o} s'$$

$$\mathrm{GoalDist}^\alpha \leq \mathrm{Distance}_{s_\star}^\alpha \qquad \text{for all abstract goal states } s_\star$$

Optimal Cost Partitioning for Landmarks

Disjunctive action landmark

- Set of operators
- Every plan uses at least one of them
- Landmark cost = cost of cheapest operator

Optimal Cost Partitioning for Landmarks

Variables

Cost_L for each landmark L

Objective

Maximize $\sum_L \text{Cost}_L$

Subject to

$$\sum_{L:o \in L} \text{Cost}_L \leq cost(o) \qquad \text{for all operators } o$$

$$\text{Cost}_L \geq 0 \qquad \text{for all landmarks } L$$

Caution

A word of warning

- optimization for every state gives best-possible cost partitioning
- but takes time

Better heuristic guidance often does not outweigh the overhead.

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)

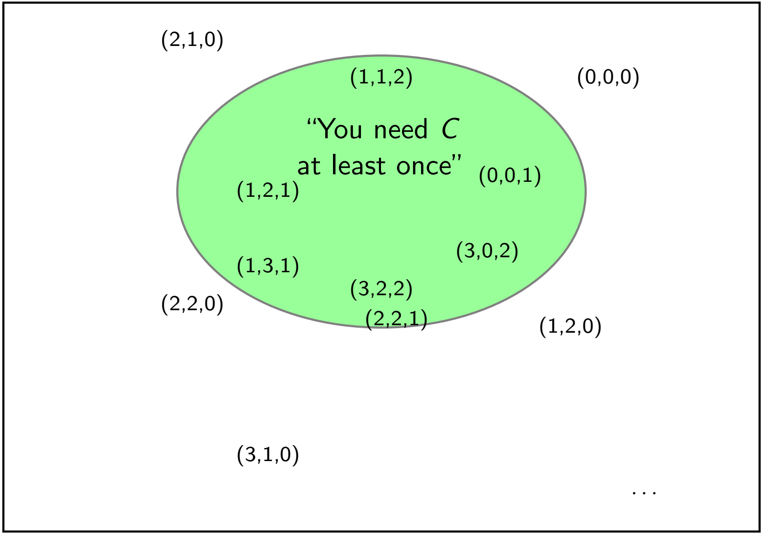
(2,2,1) (1,2,0)

(3,1,0)

. . .

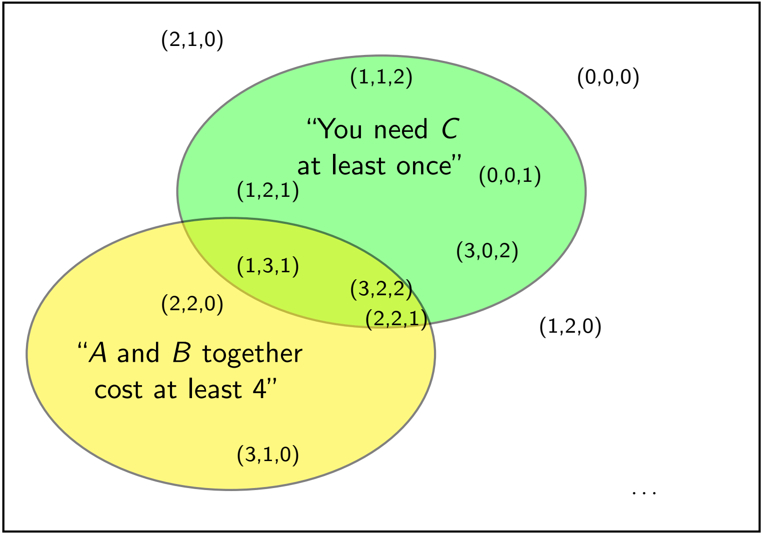
Operator Counting Heuristics

Operator occurrences in potential plans



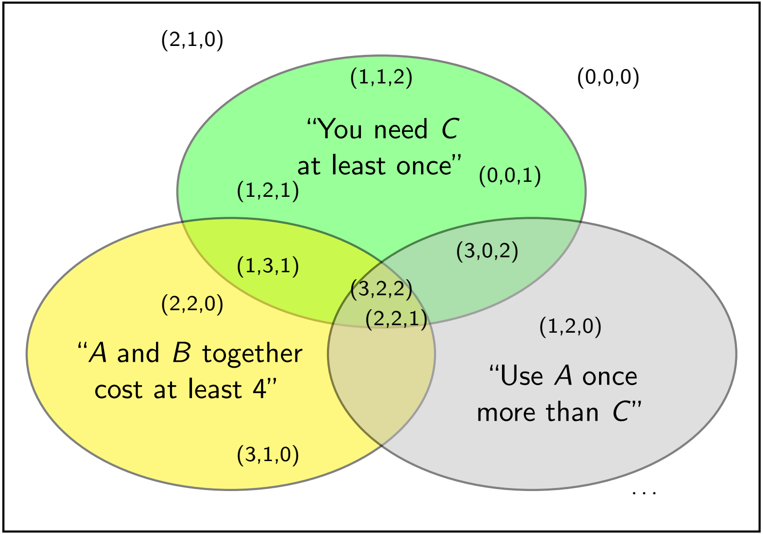
Operator Counting Heuristics

Operator occurrences in potential plans



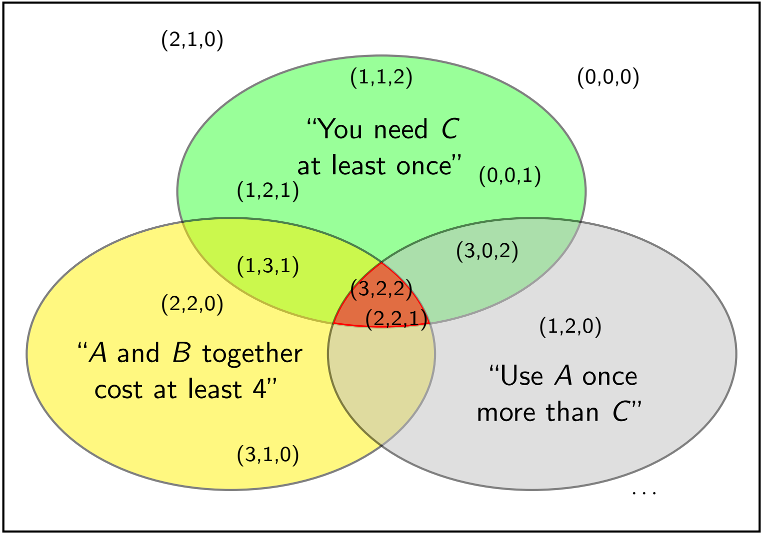
Operator Counting Heuristics

Operator occurrences in potential plans



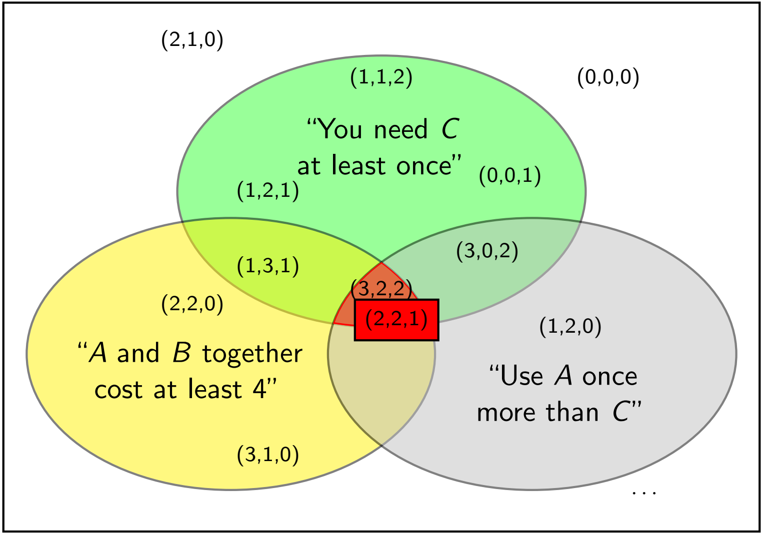
Operator Counting Heuristics

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.

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

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

Remark:

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

$\rightsquigarrow$ presentation: Tuesday, first afternoon session

Potential Heuristics

Reminder:

Idea 3: Potential Heuristics

Heuristic design as an optimization problem:

- Define simple numerical state features $f_1, \dots, f_n$.
- Consider heuristics that are linear combinations of features:

$$h(s) = w_1 f_1(s) + \dots + w_n f_n(s)$$

  with weights (potentials) $w_i \in \mathbb{R}$
- Find potentials for which h is admissible and well-informed.

Motivation:

- declarative approach to heuristic design
- heuristic very fast to compute if features are

Comparison to Previous Parts (1)

What is the same as in operator-counting constraints:

- We again use LPs to compute (admissible) heuristic values (spoiler alert!)

Comparison to Previous Parts (2)

What is different from operator-counting constraints (computationally):

- With potential heuristics, solving one LP defines the entire heuristic function, not just the estimate for a single state.
- Hence we only need one LP solver call, making LP solving much less time-critical.

Comparison to Previous Parts (3)

What is different from operator-counting constraints (conceptually):

- axiomatic approach for defining heuristics:
  - What should a heuristic look like mathematically?
  - Which properties should it have?
- define a space of interesting heuristics
- use optimization to pick a good representative

Potential Heuristics

Features

Definition (feature)

A (state) feature for a planning task is a numerical function defined on the states of the task: $f : S \to \mathbb{R}$.

Potential Heuristics

Definition (potential heuristic)

A potential heuristic for a set of features $\mathcal{F} = \{f_1, \dots, f_n\}$ is a heuristic function h defined as a linear combination of the features:

$$h(s) = w_1 f_1(s) + \cdots + w_n f_n(s)$$

with weights (potentials) $w_i \in \mathbb{R}$.

$\rightsquigarrow$ cf. evaluation functions for board games like chess

Atomic Potential Heuristics

Atomic features test if some proposition is true in a state:

Definition (atomic feature)

Let $X = x$ be an atomic proposition of a planning task.

The atomic feature $f_{X=x}$ is defined as:

$$f_{X=x}(s) = \begin{cases} 1 & \text{if variable } X \text{ has value } x \text{ in state } s \\ 0 & \text{otherwise} \end{cases}$$

- We only consider atomic potential heuristics, which are based on the set of all atomic features.
- Example for a task with state variables X and Y:

$$h(s) = 3f_{X=a} + \tfrac{1}{2}f_{X=b} - 2f_{X=c} + \tfrac{5}{2}f_{Y=d}$$

Finding Good Potential Heuristics

How to Set the Weights?

We want to find good atomic potential heuristics:

- admissible
- consistent
- well-informed

How to achieve this? Linear programming to the rescue!

Admissible and Consistent Potential Heuristics

Constraints on potentials characterize (= are necessary and sufficient for) admissible and consistent atomic potential heuristics:

Goal-awareness (i.e., $h(s) = 0$ for goal states)

$$\sum_{\text{goal facts } f} w_f = 0$$

Consistency

$$\sum_{\substack{f \text{ consumed} \\ \text{by } o}} w_f - \sum_{\substack{f \text{ produced} \\ \text{by } o}} w_f \leq cost(o) \quad \text{for all operators } o$$

Remarks:

- assumes transition normal form (not a limitation)
- goal-aware and consistent = admissible and consistent

Well-Informed Potential Heuristics

How to find a well-informed potential heuristic?

⇝ encode quality metric in the objective function
and use LP solver to find a heuristic maximizing it

Examples:

- maximize heuristic value of a given state (e.g., initial state)
- maximize average heuristic value of all states
(including unreachable ones)
- maximize average heuristic value of some sample states
- minimize estimated search effort

⇝ see Seipp et al. presentation (joint ICAPS/SoCS session)

Optimal Cost Partitioning

- Michael Katz, Carmel Domshlak.
  Optimal Additive Composition of Abstraction-based Admissible Heuristics. ICAPS 2008
  - optimal cost partitioning for abstractions
- Erez Karpas, Carmel Domshlak.
  Cost-optimal Planning with Landmarks. IJCAI 2009
  - optimal cost partitioning for landmarks
  - we showed a simplified version with fewer variables which can be traced back to Keyder, Richter, and Helmert (2010)
- Blai Bonet, Malte Helmert.
  Strengthening Landmark Heuristics via Hitting Sets.
  ECAI 2010
  - optimal cost partitioning for landmarks (dual)

Operator Counting

- Florian Pommerening, Gabriele Röger,
  Malte Helmert, Blai Bonet.
  LP-based Heuristics for Cost-optimal Planning. ICAPS 2014
  - operator-counting framework
- Florian Pommerening, Gabriele Röger, Malte Helmert.
  Getting the Most Out of Pattern Databases for Classical Planning. IJCAI 2013
  - post-hoc optimization
- Tatsuyai Imai, Alex Fukunaga.
  A Practical, Integer-linear Programming Model for the Delete-relaxation in Cost-optimal planning. ECAI 2014
  - operator-counting constraints for relaxed planning
- Toby Davies, Adrian R. Pearce,
  Peter J. Stuckey, Nir Lipovetzky.
  Sequencing Operator Counts. ICAPS 2015
  - new constraints if operator counts do not correspond to a plan

State-equation Heuristic

- Menkes van den Briel, J. Benton, Subbarao Kambhampati, Thomas Vossen. An LP-Based Heuristic for Optimal Planning. CP 2007
  - state equation heuristic
- Blai Bonet. An Admissible Heuristic for SAS+ Planning Obtained from the State Equation. IJCAI 2013
  - state equation heuristic
- Blai Bonet, Menkes van den Briel. Flow-based Heuristics for Optimal Planning: Landmarks and Merges. ICAPS 2014
  - state equation heuristic with dynamic fluent merging

Connections and Potential Heuristic

- Florian Pommerening, Malte Helmert,
  Gabriele Röger, Jendrik Seipp.
  From Non-Negative to General Operator Cost Partitioning.
  AAAI 2015
  - general cost partitioning
  - potential heuristic
  - connection of the three concepts
- Jendrik Seipp, Florian Pommerening, Malte Helmert.
  New Optimization Functions for Potential Heuristics.
  ICAPS 2015
  - quality objectives for potential heuristics