Lagrangian Decomposition for Classical Planning (Extended Abstract)

Florian Pommerening[1], Gabriele Röger[1], Malte Helmert[1], Hadrien Cambazard[2], Louis-Martin Rousseau[3], Domenico Salvagnin[4]

[1]University of Basel, Basel, Switzerland [2]Univ. Grenoble Alpes, Grenoble INP, G-SCOP, 38000 Grenoble, France

[3]Polytechnique Montreal, Montreal, Canada [4]University of Padua, Padua, Italy

Main Results

- Understand cost partitioning as Lagrangian decomposition
- Develop anytime algorithm for improving cost partitioning

Lagrangian Decomposition

Problem P

$\text{Min } c\, x \text{ s.t.}$
$A_1 x \geq b_1$
$\dots$
$A_k x \geq b_k$
$x \geq 0$

→ rewrite →

Problem P

$\text{Min } c\, x \text{ s.t.}$
$A_i x_i \geq b_i \quad \forall i$
$x = x_i \quad \forall i$
$x, x_i \geq 0 \quad \forall i$

→ relax →

Relaxed Problem $P(\lambda)$

$\text{Min } c\, x + \sum_i \lambda_i (x_i - x) \text{ s.t.}$
$A_i x_i \geq b_i \quad \forall i$
$x, x_i \geq 0 \quad \forall i$

→ decompose →

Subproblem $P_0(\lambda)$

$\text{Min } \left(c - \sum_i \lambda_i\right) x \text{ s.t.}$
$x \geq 0$

Subproblem $P_i(\lambda)$

$\text{Min } \lambda_i x_i \text{ s.t.}$
$A_i x_i \geq b_i$
$x_i \geq 0$

Implicit Constraint

$$\sum_i \lambda_i \leq c$$

Lagrangian Dual Problem

$\text{Max } \sum_i P_i(\lambda) \text{ s.t.}$
$$\sum_i \lambda_i \leq c$$

Relation to Cost Partitioning

original problem P ⇔ operator-counting LP (multiple abstractions)

Lagrangian dual ⇔ cost partitioning

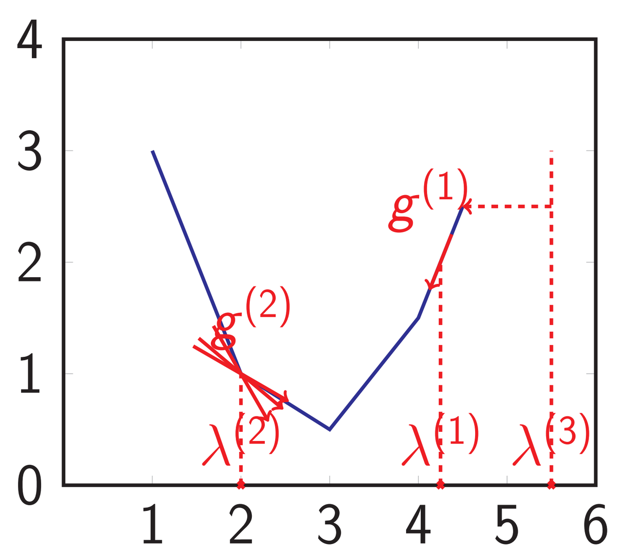
subproblem $P_i(\lambda)$ ⇔ operator-counting LP (single abstraction)

objective coefficients c ⇔ original cost function

Lagrangian multipliers λ_i ⇔ partitioned cost functions

LP variables x, x_i ⇔ operator-counting variables

Projected Subgradient Method



Solving the Lagrangian dual:

- choose point $\lambda^{(1)}$
- repeat for $t = 1, 2 \dots$
  - find subgradient $g^{(t)}$ at $\lambda^{(t)}$
  - compute step length $\eta(t)$
  - set $\lambda^{(t+1)} = \text{proj}((\lambda^{(t)} + \eta(t) g^{(t)})$

Relation to Cost Partitioning

In general

Subgradient g at λ ⇔ optimal solution of subproblem $P_i(\lambda)$

Projection ⇔ projection of arbitrary cost functions to a cost partitioning

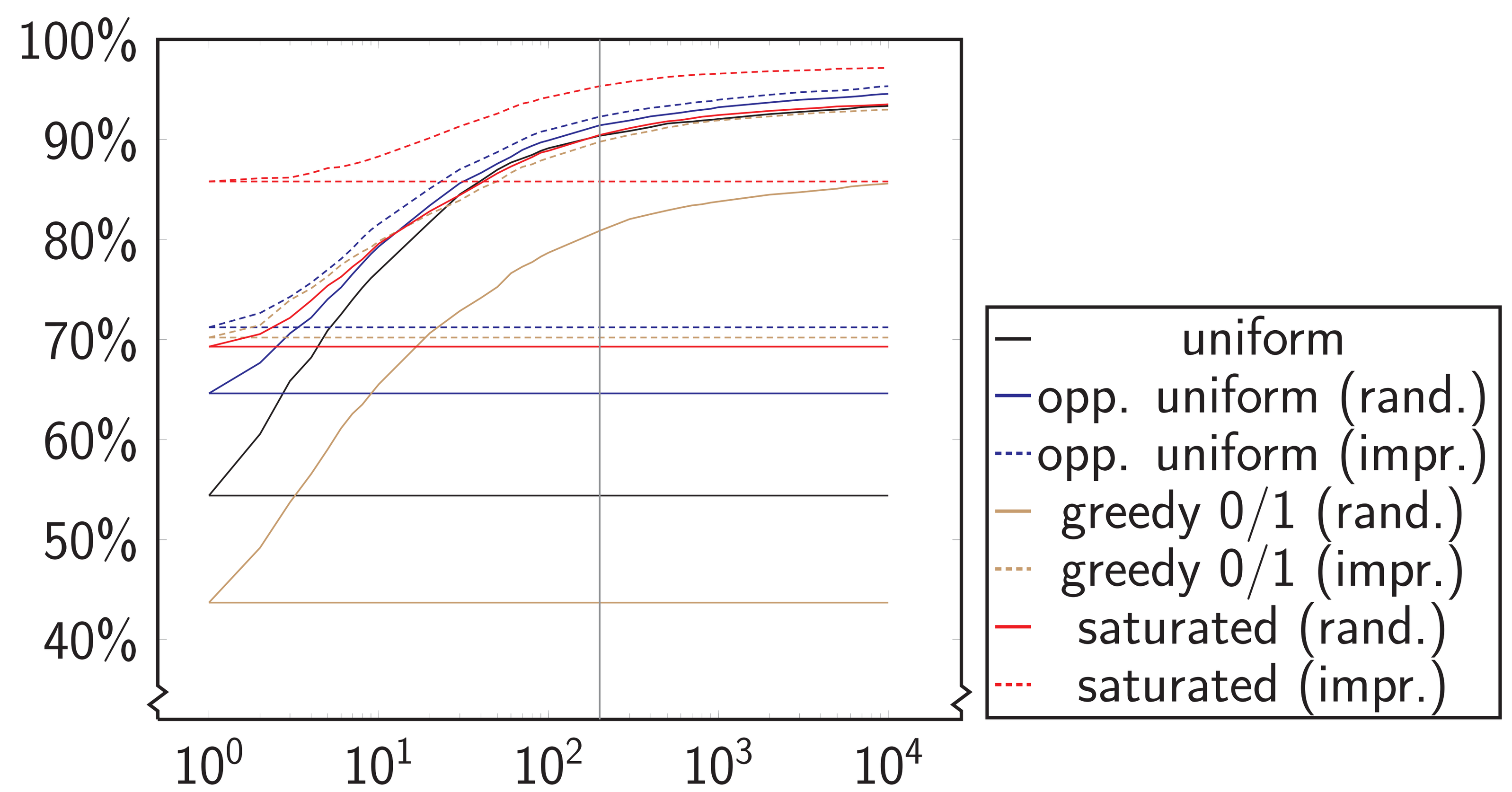
In the context of abstraction heuristics

Subgradient g at λ ⇔ number of times each operator is used in a cheapest plan under cost λ

Anytime Algorithm for Cost Partitioning

- choose any cost partitioning $cost^{(1)}$
- repeat for $t = 1, 2 \dots$
  - for each abstraction i
    - find optimal solution π^* under $cost_i^{(t)}$
    - set $cost_i^{(t+1)}(o) = cost_i^{(t)}(o) + \eta(t) occurrences(o, \pi^*)$
  - project $cost^{(t+1)}$ to a cost partitioning

Heuristic Quality per Iteration



Time to Run 200 Iterations