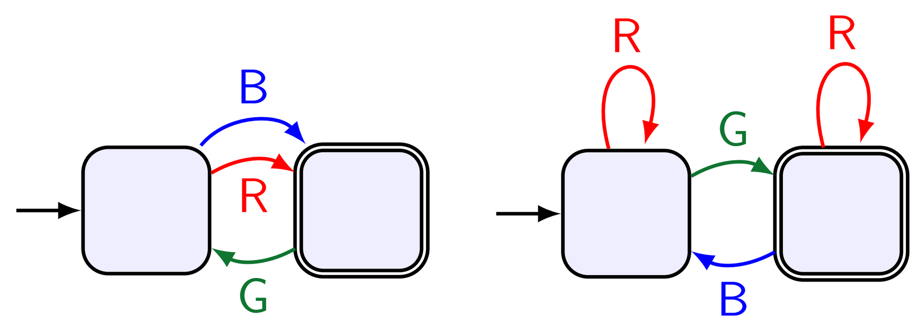
Dantzig-Wolfe decomposition **speeds up cost partitioning** and the involved LPs have **intuitive interpretations.**

Example



Assume we already have these candidate cost functions available.

	Abstr. 1	Abstr. 2
cost(R)	1	0
cost(B)	1	0
cost(G)	0	1
h	1	1

The best combination uses each column once for a heuristic value of 2.

In the dual Master Problem each column corresponds to one constraint.

Dual Master Problem

$$\text{Minimize } y_R + y_B + y_G \text{ s.t.}$$
$$y_R + y_B \geq 1$$
$$y_G \geq 1$$
$$y_R \geq 0 \quad y_B \geq 0 \quad y_G \geq 0$$

Optimal Solution

$$y_R = 1 \quad y_B = 0 \quad y_G = 1$$

The dual solution is used in the objective of the pricing problems.

Pricing Problem

$$\text{Minimize } c(R) + c(G) - h \text{ s.t.}$$
$$h \leq \text{heuristic } i \text{ under cost } c$$

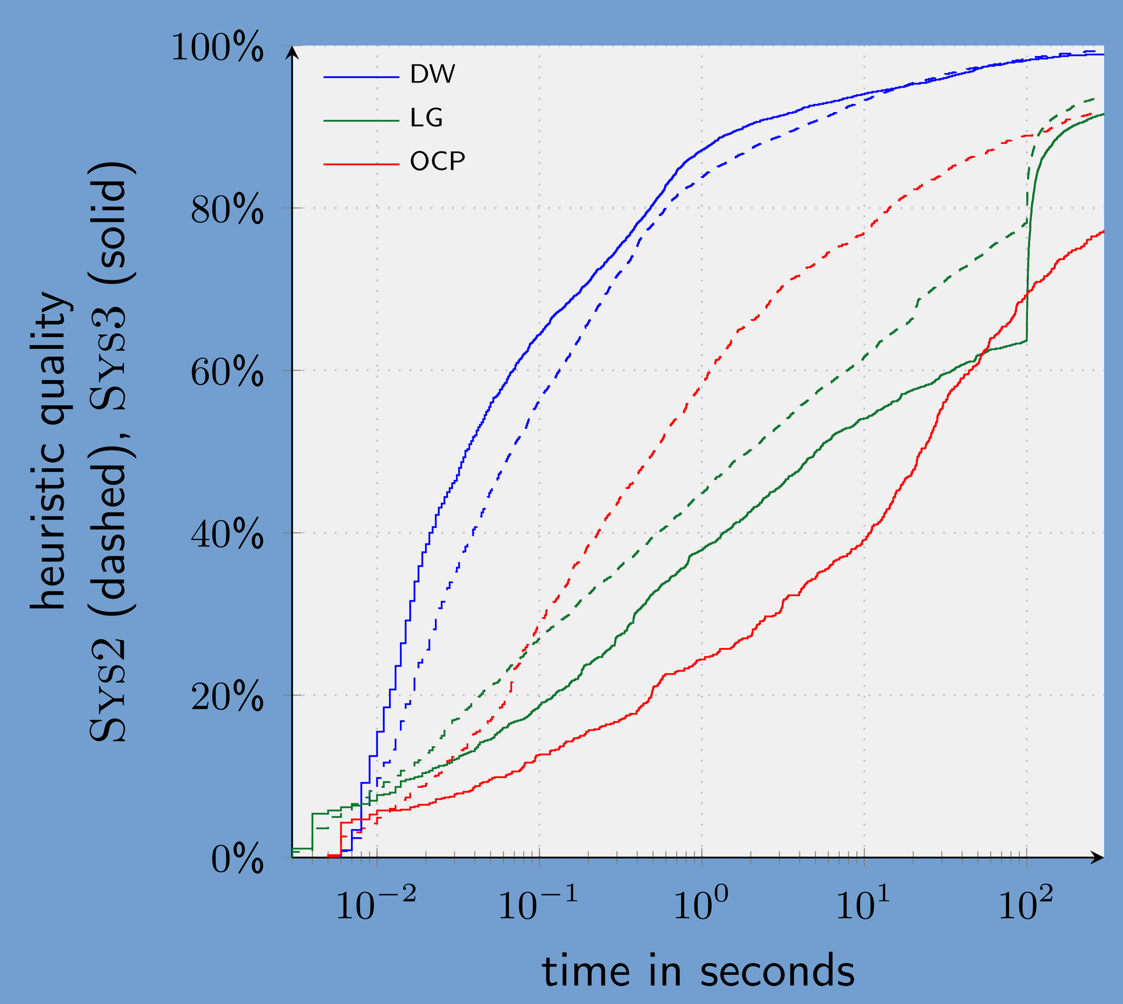
Optimal Solution for First Abstraction

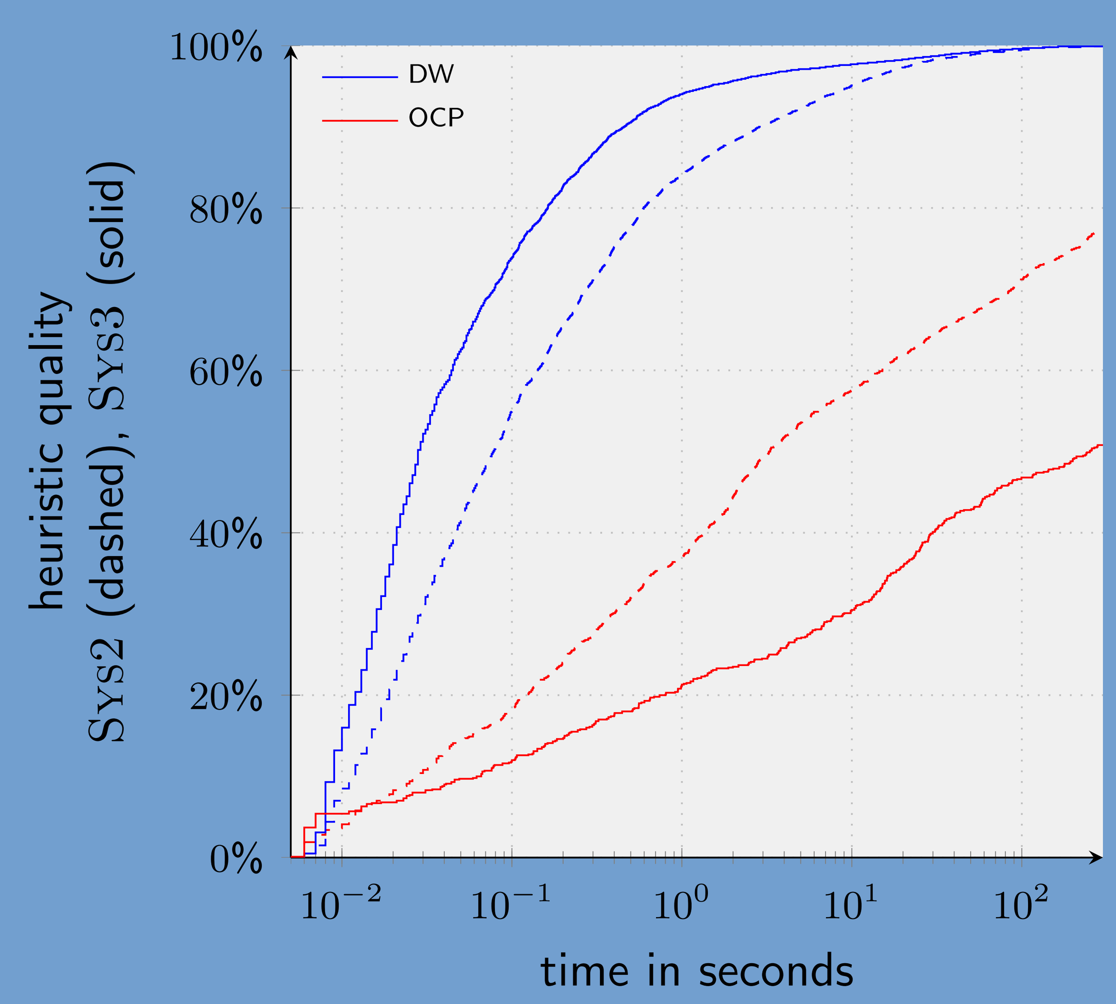
$$c(R) = 1 \quad c(B) = 1 \quad c(G) = -1 \quad h = 1$$

The solution of the pricing problem is added as a new column.

	Abstr. 1		Abstr. 2
cost(R)	1	1	0
cost(B)	1	1	0
cost(G)	0	-1	1
h	1	1	1

The best combination now uses the last column twice and the middle column once for a total heuristic value of 3.





Dantzig-Wolfe Decomposition for Cost Partitioning

Florian Pommerening, Thomas Keller, Valentina Halasi, Jendrik Seipp, Silvan Sievers, Malte Helmert

University of Basel

LINKÖPING UNIVERSITY

PDF & Talk

Extensions

- Combine equivalent labels
- Consider only interesting patterns
- Incrementally add subproblems

Relaxations

- Stop adding subproblems
- Stop adding columns
- Stop evaluating the master problem

Interpretations

Master Problem:
Combine candidate cost funtions.

$$\text{Maximize } \sum_i \sum_j \lambda_{ij} h_{ij} \text{ s.t.}$$
$$\sum_i \sum_j \lambda_{ij} c_{ij}(o) \leq \mathit{cost}(o) \quad \text{for all } o$$
$$\lambda_{ij} \geq 0 \quad \text{for all } i, j$$

Dual Master Problem:
Posthoc optimization heuristic.

$$\text{Minimize } \sum_o y_o \mathit{cost}(o) \quad \text{s.t.}$$
$$\sum_o y_o c_{ij}(o) \geq h_{ij} \quad \text{for all } i, j$$
$$y_o \geq 0 \quad \text{for all } o$$

Pricing Problem:
Balance cost of operator count *y* and cheapest abstract plan.
Return saturated cost function.

$$\text{Minimize } c(y) - h \text{ s.t.}$$
$$h \leq \text{heuristic } i \text{ under cost } c$$

Dual Pricing Problem:
Generate column iff given operator count induces no flow.

$$\text{Maximize } 0 \text{ s.t.}$$
$$\sum_{t = s \xrightarrow{o} s'} f_t = y_o \quad \text{for all } o$$
$$f \text{ is a flow in abstraction } i$$