Column generation: stabilization

Discrete Optimization

Giovanni Righini

UNIVERSITÀ DEGLI STUDI DI MILANO

Column generation convergence

Column generation guarantees a valid dual bound only at the end of the algorithm, when the optimal solution of the master problem has been found (no negative reduced cost columns exist).

Convergence is typically very slow: many iterations with little or no improvement of the optimal value may be required to prove optimality.

The reason is primal degeneracy, that implies multiple dual optima.

Dual variables oscillate instead of converging to the optimal values.

Stabilization techniques

Boxstep method (Marsten et al, 1975). At each iteration the dual variables are constrained to lie in a box centered in the previous dual solution.

Du Merle, Villeneuve, Desrosiers and Hansen (1999): dual stabilization method specifically devised for column generation.

Key observations:

- The columns of the optimal solution are often generated in the last iterations, when the dual variables are near-optimal.
- Initially, wild oscillations of dual variables tend to produce "extreme columns", that are not part of any optimal solution.

So, one should try to generate columns using dual variables near a stability center, i.e. the current best guess for the optimal dual values.

Stabilization techniques

Stabilizing function: it penalizes dual solutions far from the stability center.

The stability center is iteratively updated, until it converges to an optimal dual solution.

The stabilizing function is iteratively updated, tending to zero.

This requires to introduce additional bounded variables in the restricted linear master problem.

Stabilization techniques

$$\begin{aligned} \text{minimize } z = & c^T x \\ \text{s.t. } Ax &= b \\ x &\geq 0 \end{aligned}$$

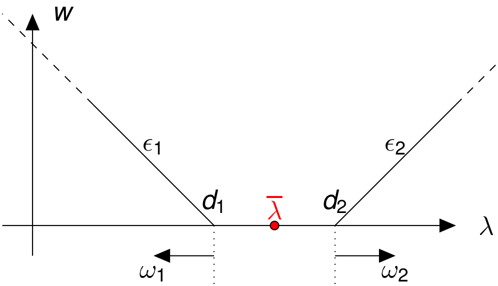
$$\begin{aligned} \text{maximize } w = & b^T \lambda \\ \text{s.t. } A^T\lambda &\leq c \end{aligned}$$

Stabilized master problem:

$$\begin{aligned} \text{minimize } z = & c^T x - d_1 y_1 + d_2 y_2 \\ \text{s.t. } Ax - y_1 + y_2 &= b \\ y_1 &\leq \epsilon_1 \\ y_2 &\leq \epsilon_2 \\ x, y_1, y_2 &\geq 0 \end{aligned}$$

$$\begin{aligned} \text{maximize } w = & b^T \lambda - \omega_1 \epsilon_1 - \omega_2 \epsilon_2 \\ \text{s.t. } A^T\lambda &\leq c \\ d_1 - \omega_1 \leq \lambda &\leq d_2 + \omega_2 \\ \omega_1, \omega_2 &\geq 0 \end{aligned}$$

Stabilized master problem



The current interval is always centered around the stability center $\overline{\lambda}$.

- If λ falls out of the interval $[d_1, d_2]$, enlarge the interval.
- If λ falls within the interval $[d_1, d_2]$, restrict the interval and decrease the penalties.

Drawbacks: this method enlarges the LP and requires parameter tuning.

Single parameter stabilization

A different technique (Pessoa, Uchoa and Poggi, 2008) uses a single parameter.

Key idea: the dual values λ used to solve the pricing problem are not the current optimal dual values λ^*, but a convex combination between λ^* and the current stability center $\overline{\lambda}$:

$$\lambda = \alpha\overline{\lambda} + (1 - \alpha)\lambda^*,$$

with $0 \leq \alpha \leq 1$.

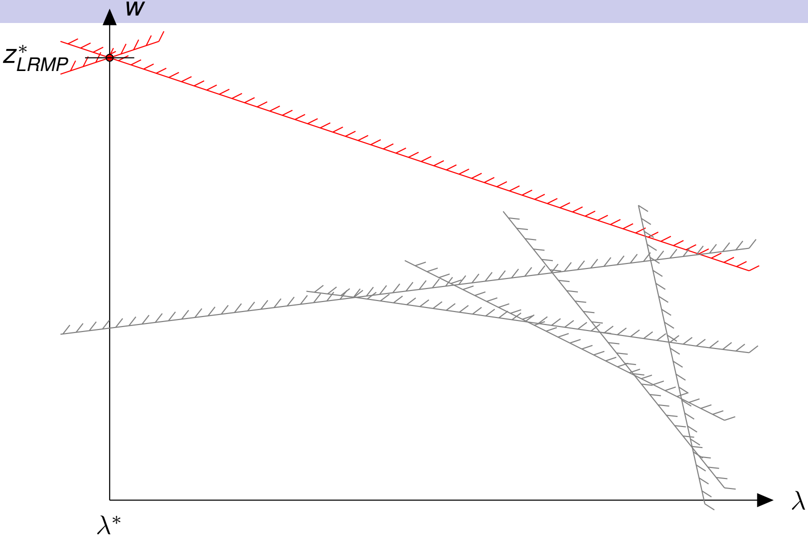
If the solution of the pricing problem with λ does not yield a column whose reduced cost is negative with respect to λ^*, then

$$z^*_{LR}(\lambda) \geq z^*_{LR}(\overline{\lambda}) + \alpha(z^*_{LRMP} - z^*_{LR}(\overline{\lambda})).$$

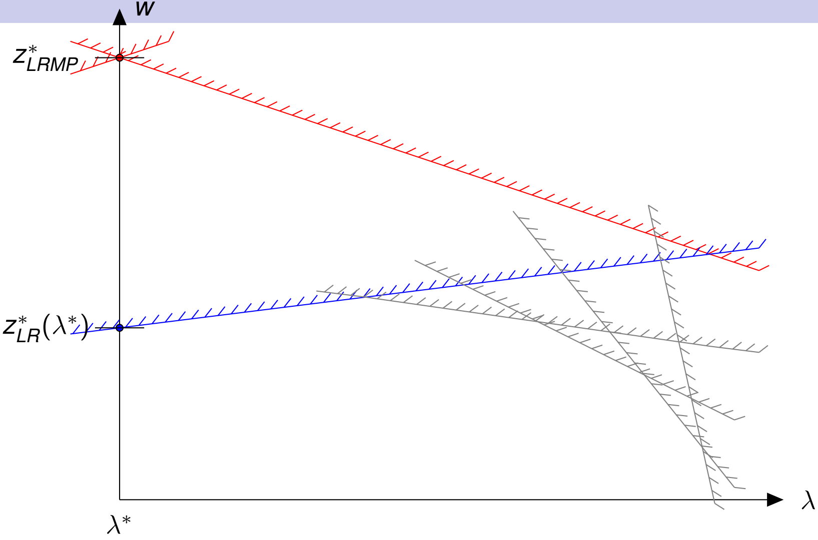
Mis-pricing is not wasted time: the lower bound improves.
The method converges in a finite number of iterations.

Single parameter stabilization

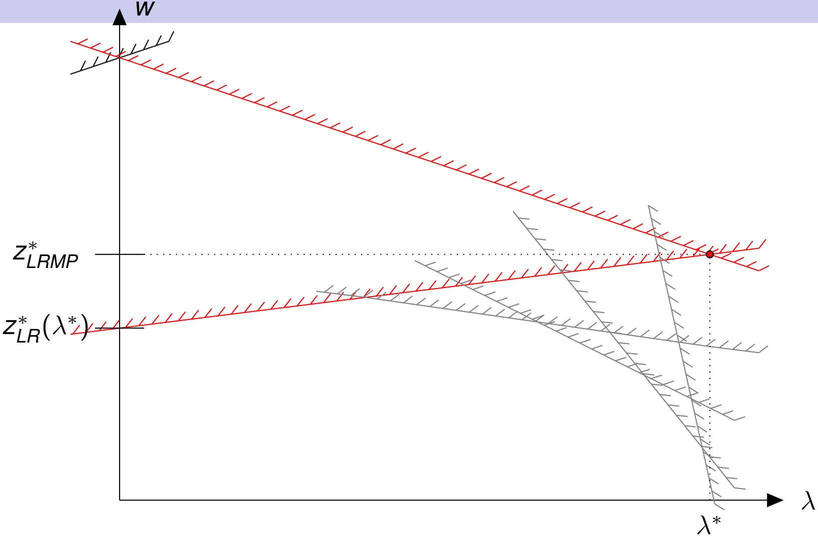
Input: α, ϵ.
$\overline{\lambda} \leftarrow 0$
repeat
  Solve *LRMP* $\rightarrow \lambda^*, z^*_{LRMP}$
  $\lambda \leftarrow \alpha\overline{\lambda} + (1 - \alpha)\lambda^*$
  Solve pricing problem with $\lambda \rightarrow$ column A_j
  if $z^*_{LR}(\lambda) > z^*_{LR}(\overline{\lambda})$ **then**
    $\overline{\lambda} \leftarrow \lambda$
  end if
  if $\overline{c}_j(\lambda^*) < 0$ **then**
    Insert A_j in the *LRMP*
  end if
until $z^*_{LRMP} - z^*_{LR}(\overline{\lambda}) \leq \epsilon$



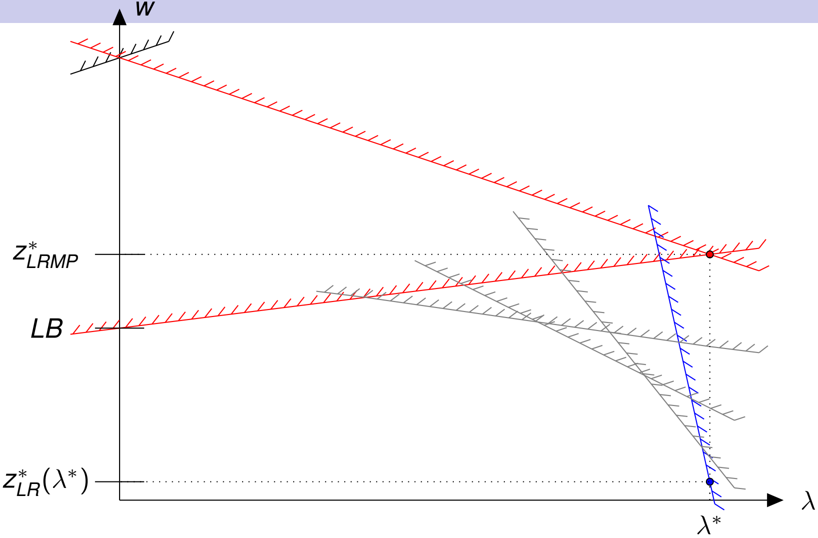
Iteration 1 (master): dummy initial basis.



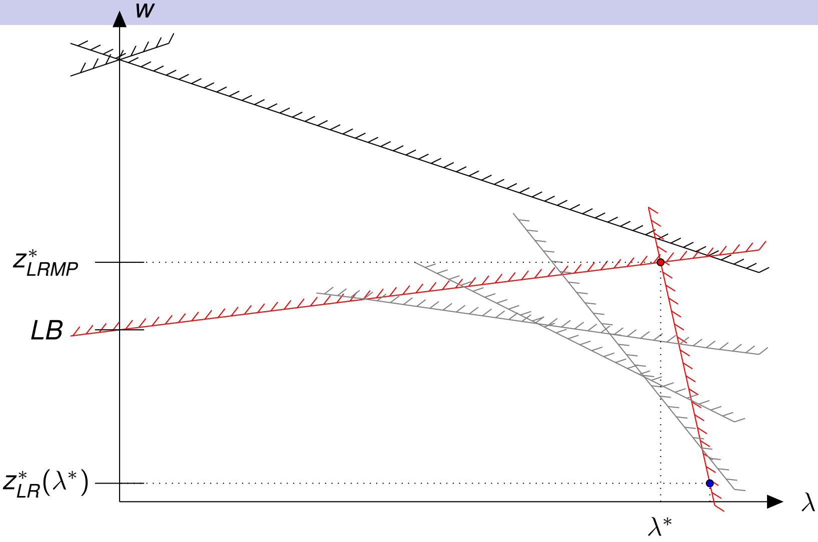
Iteration 1 (pricing): new column generated. A valid Lagrangean lower bound is obtained.



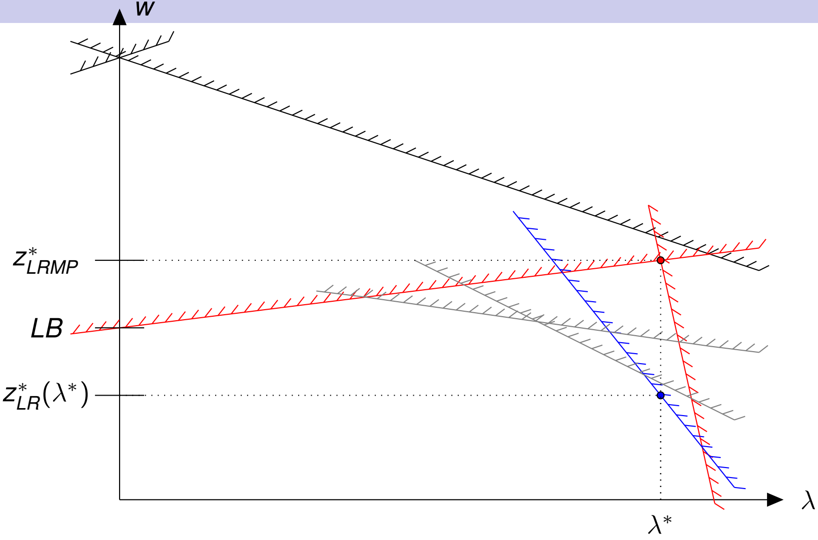
Iteration 2 (master): a large change of λ^* occurs.



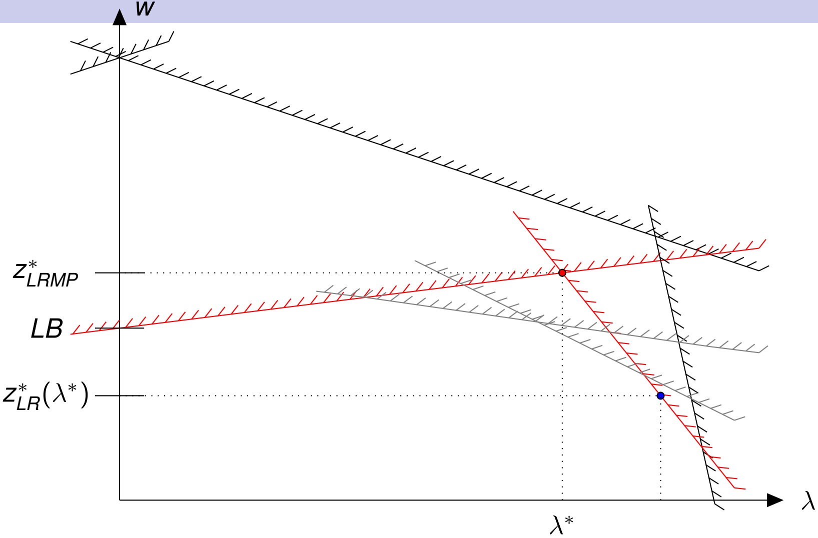
Iteration 2 (pricing): The Lagrangean lower bound is worse.



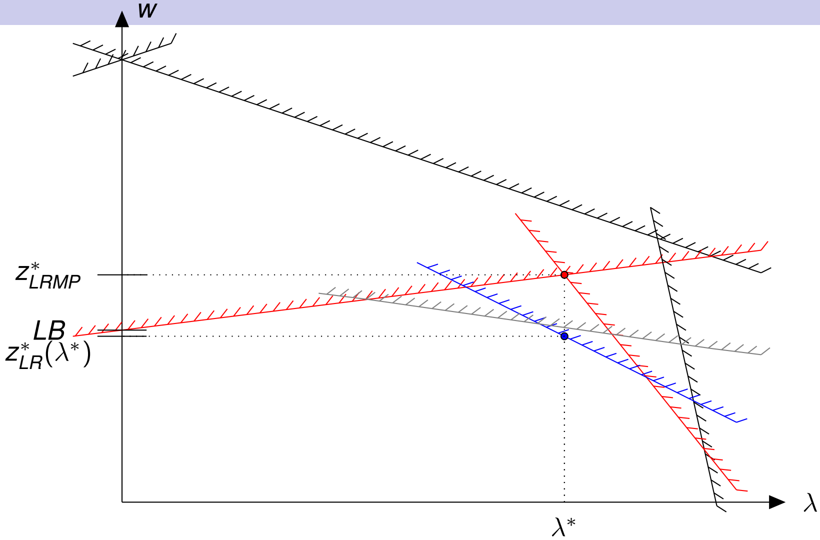
Iteration 3 (master): z^*_{LRMP} keeps decreasing monotonically.



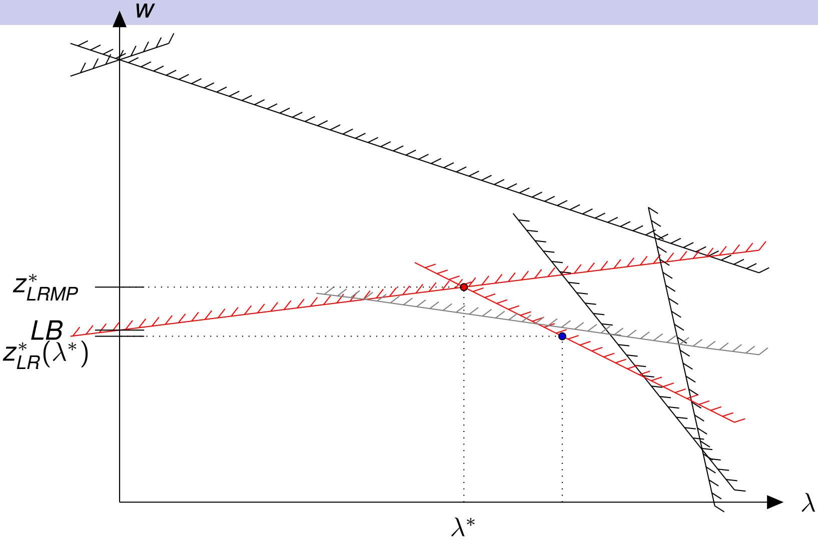
Iteration 3 (pricing): $z^*_{LR}(\lambda^*)$ remains below the current best LB.



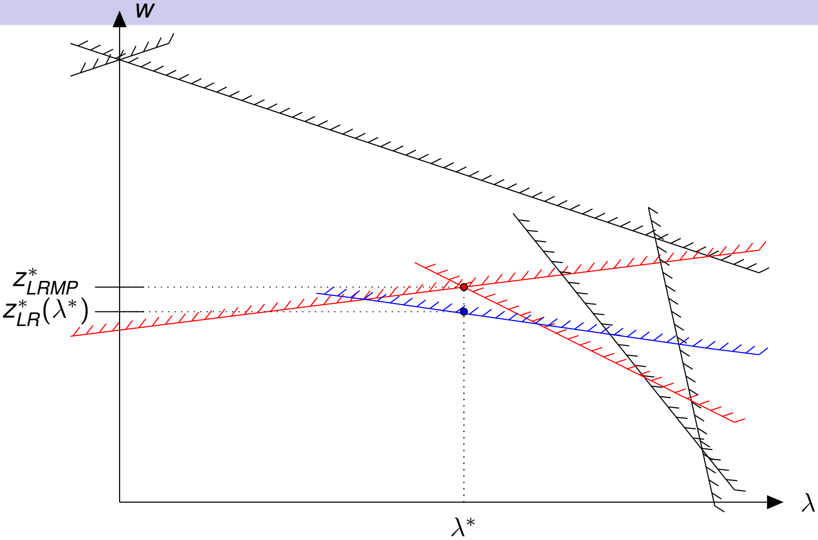
Iteration 4 (master): the change of λ is larger than the previous one.



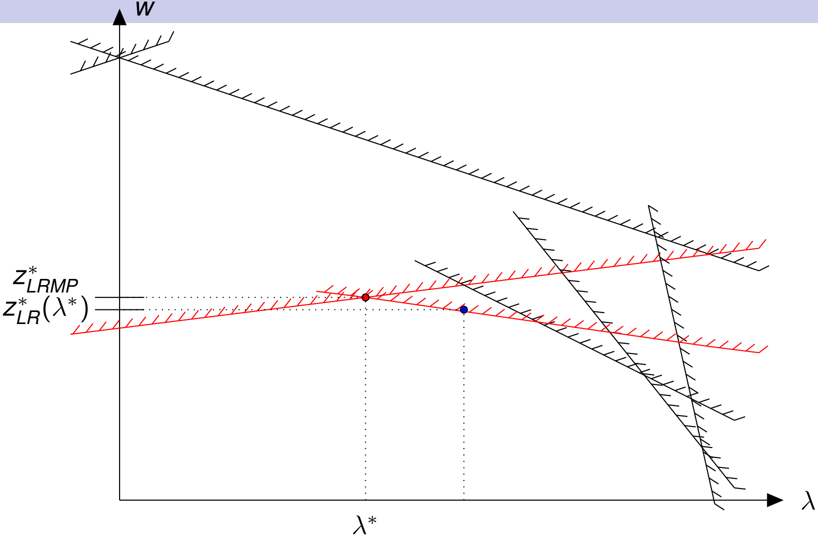
Iteration 4 (pricing): still no improvement of the best current *LB*.



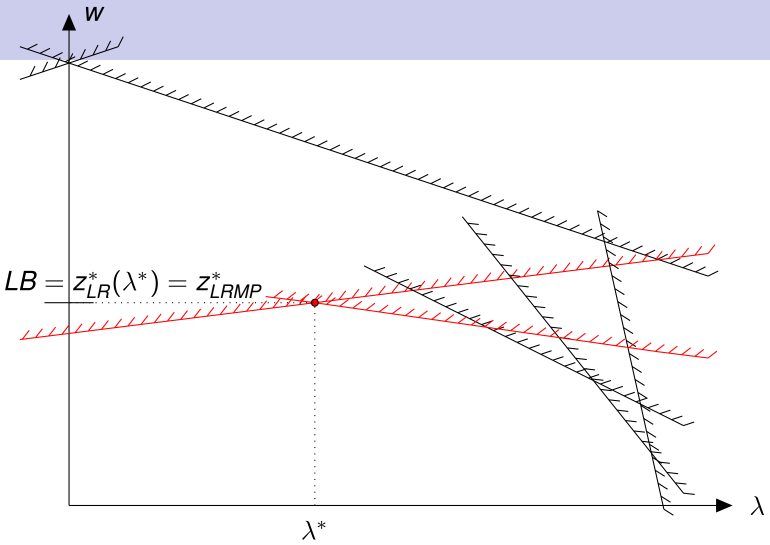
Iteration 5 (master): the master is more and more relaxed; its dual is more and more constrained.



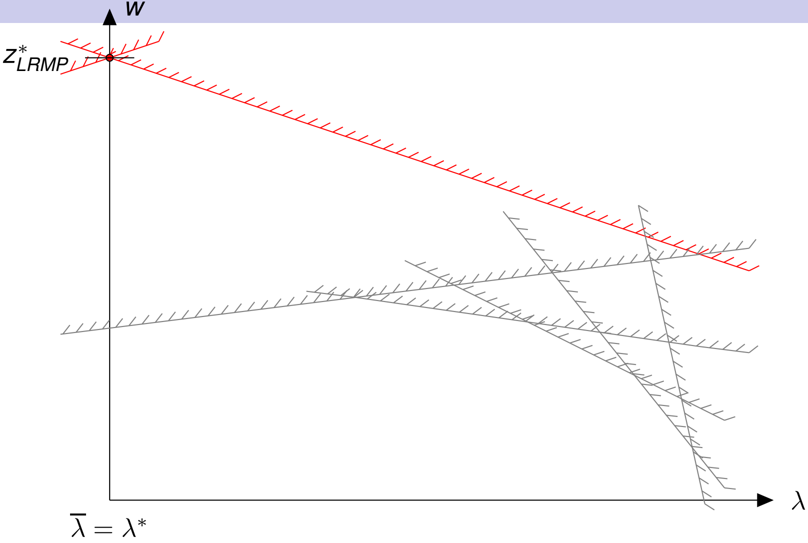
Iteration 5 (pricing): improvement of the current best *LB* (finally!).



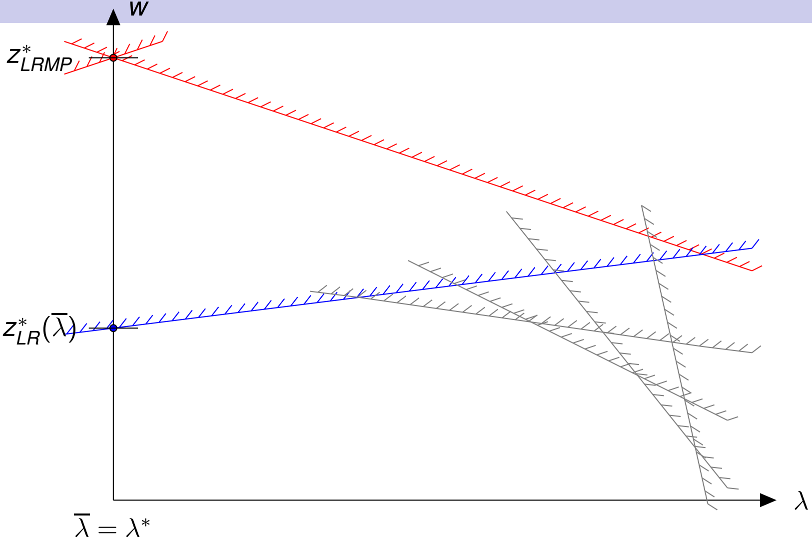
Iteration 6 (master): we have the optimal solution but not the optimality guarantee.



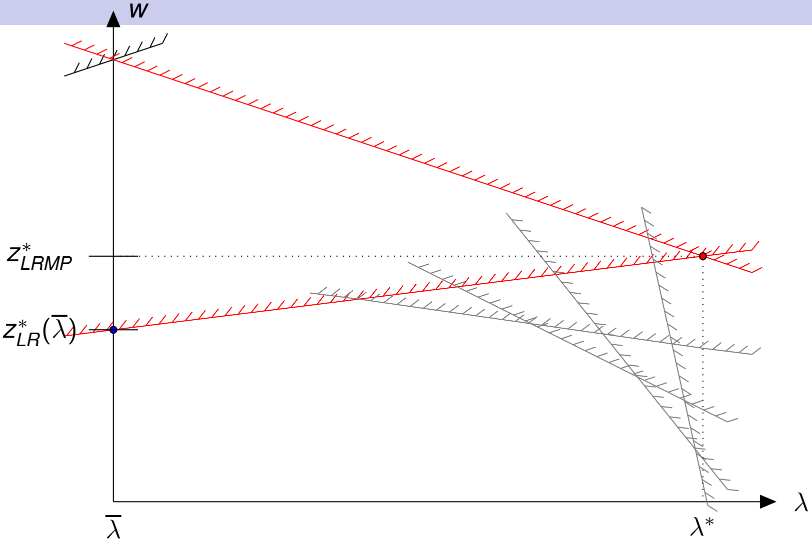
Iteration 6 (pricing): No improving columns. Optimality is proven.
$LB = z^*_{LRMP}$.



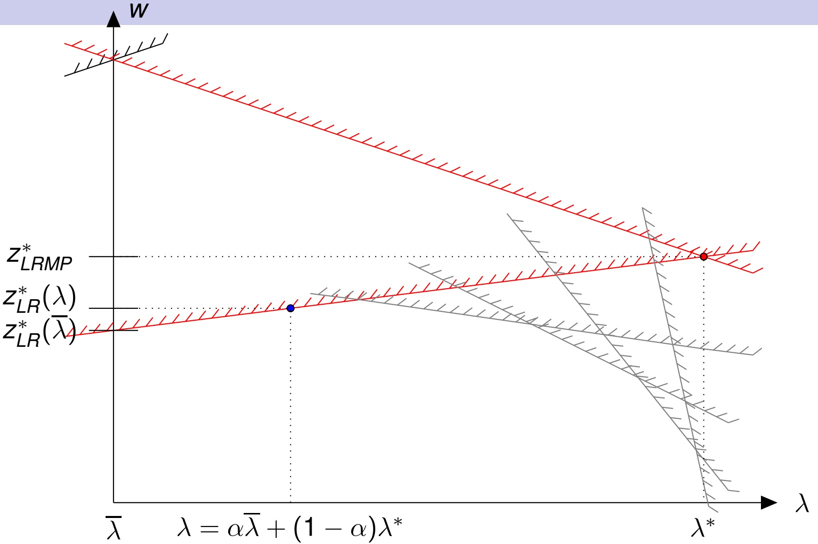
Iteration 1 (master): dummy initial basis. Assume $\overline{\lambda} = 0$, $\alpha = 0.3$.



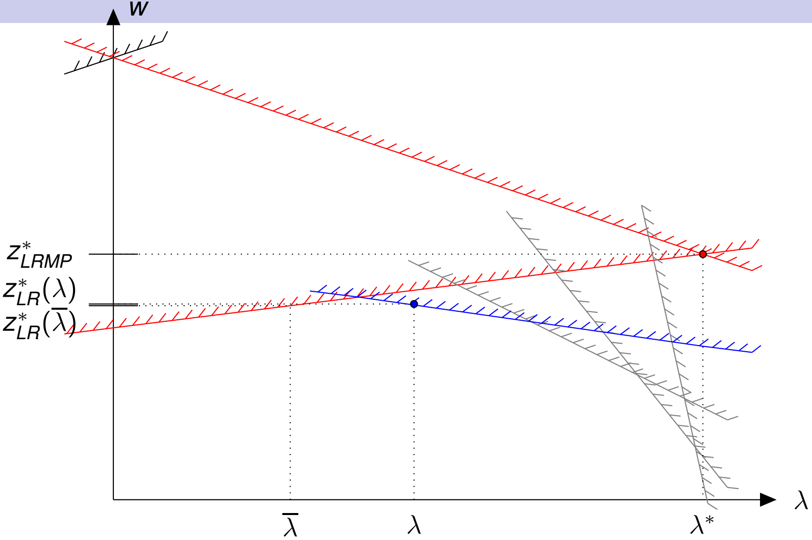
Iteration 1 (pricing): the first iteration is the same as before.



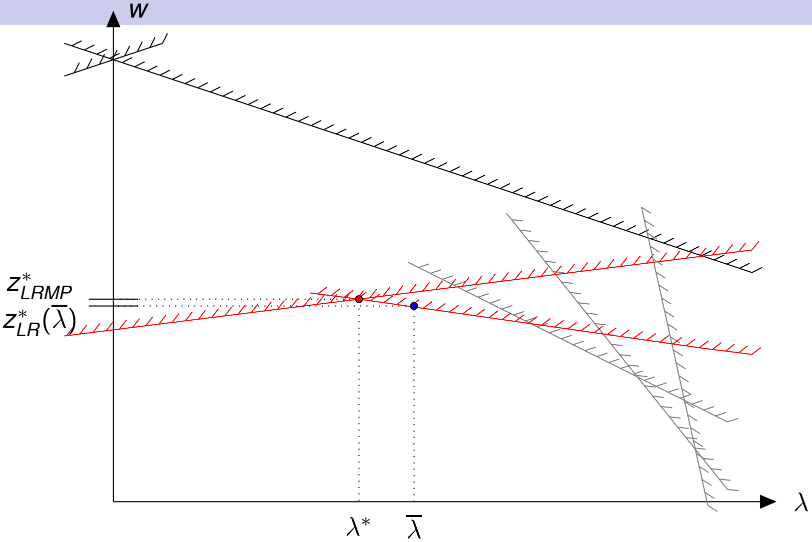
Iteration 2 (master): the iteration is again the same as before.



Iteration 2 (pricing): a convex combination of $\overline{\lambda}$ and λ^* is used.
Misprice: the LB improves by at least $\alpha(z^*_{LRMP} - z^*_{LR}(\overline{\lambda}))$.



Iteration 2b (pricing): the stability is center is updated. A new column is found.



Iteration 3 (master): optimal solution found, but with no guarantee yet.
Stop when $z^*_{LRMP} - z^*_{LR}(\overline{\lambda}) < \epsilon$.