Accelerating Benders decomposition by level stabilization

Sofia Zaourar-Michel

INRIA Grenoble

ROADEF, February 2014

Joint work with Jérôme Malick (CNRS, Grenoble)

Outline

Outline

Generalized Benders decomposition

The problem

$$\begin{array}{ll} \min & f(x, y) \\ \text{s.t.} & G(x, y) \leq 0 \\ & x \in X,\ y \in Y \end{array}$$

where y is a complicating variable

The idea: variable decomposition [Geoffrion 1972]

$$\min_{y \in Y} v(y) \quad \text{with} \quad v(y) := \min_{x \in X, G(x,y) \leq 0} f(x, y)$$

$$v : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$$

Example: network design problems

Fixed charged uncapicitated network design

- Variables
  - x : flow quantity
  - y : network links
- Formulation

$$
\begin{aligned}
\min \quad & \sum_{ijk} c_{ijk} x_{ijk} + \sum_{ij} f_{ij} y_{ij} \\
\text{s.t.} \quad & \sum_{j} x_{ijk} - \sum_{j} x_{jik} = \begin{cases} d_k & i = O(k) \\ -d_k & i = D(k) \\ 0 & \text{otherwise} \end{cases} \\
& x_{ijk} \leq d_k y_{ij} \\
& x \geq 0, \quad y \in \{0,1\}^{n^2}
\end{aligned}
$$

Fixing y leads to K independent network flow subproblems

General case of Benders decomposition

Assumptions

1. v **is convex** (and we can compute a subgradient)

$$v(y) := \min_{x \in X, G(x,y) \leq 0} f(x, y)$$

2. There is **no duality gap** for the subproblem

$$v(y) = \max_{u \geq 0} \theta_y(u) = \max_{u \geq 0} \min_{x \in X} \{ f(x, y) + u^\top G(x, y) \}$$

Benders master problem

- Cutting-plane model

$$\begin{aligned} \check{v}_k(y) &:= \max_{i \in I_k} \{ v(y_i) + g_i^\top (y - y_i) \} &\leq v(y) \\ \check{V}_k &:= \{ y \in Y : \beta_j^\top y \leq \alpha_j,\ j \in J_k \} &\supseteq \operatorname{dom} v \end{aligned}$$

- Benders **relaxed master**: (MILP)

$$y_{k+1} \in \operatorname{argmin} \check{v}_k(y) \quad \text{s.t.} \quad y \in \check{V}_k$$

Benders method

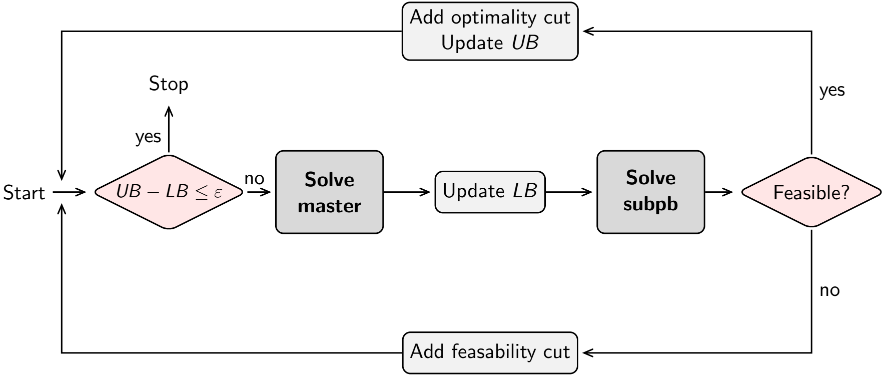


Illustration of Benders method

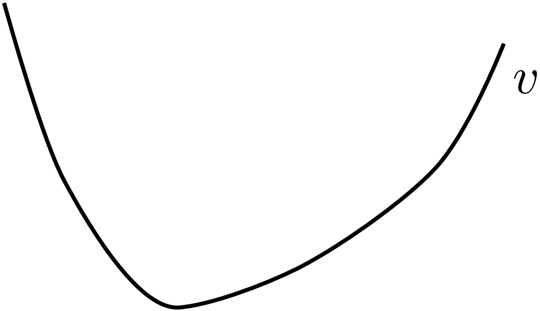


Illustration of Benders method

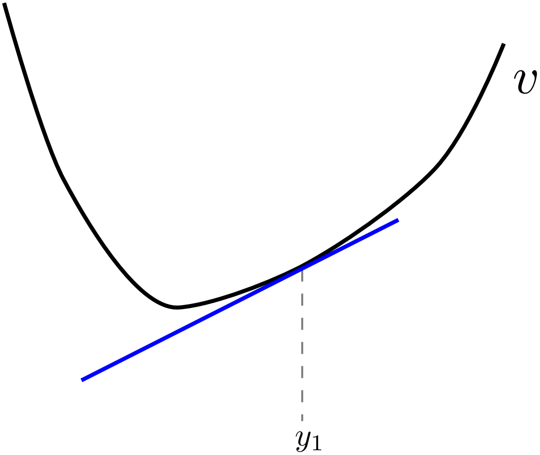


Illustration of Benders method

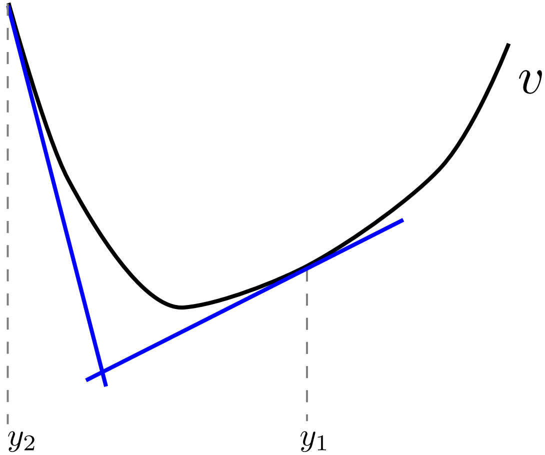


Illustration of Benders method

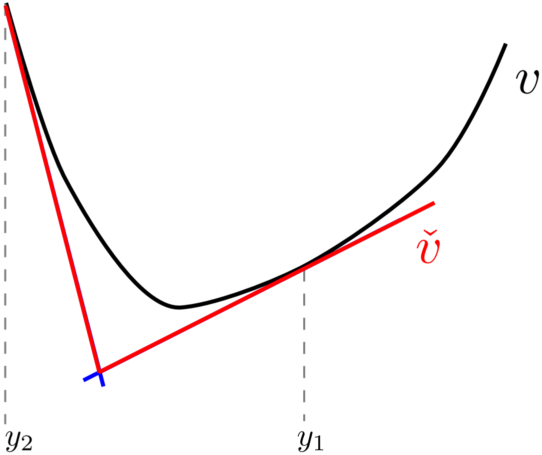


Illustration of Benders method

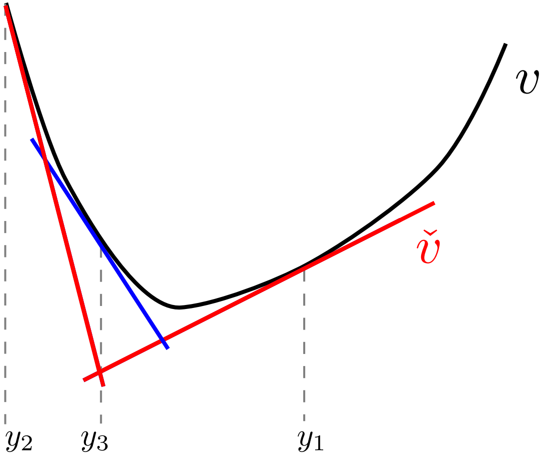


Illustration of Benders method

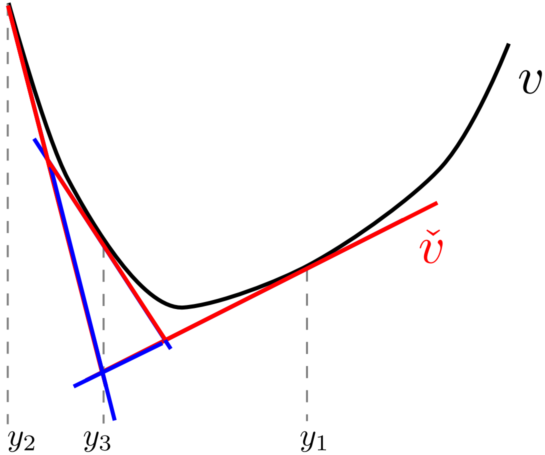


Illustration of Benders method

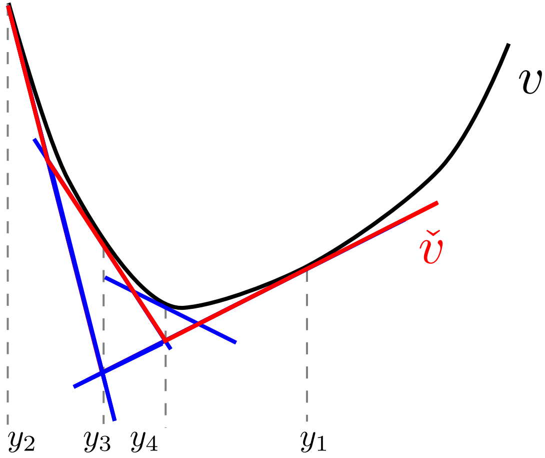


Illustration of Benders method

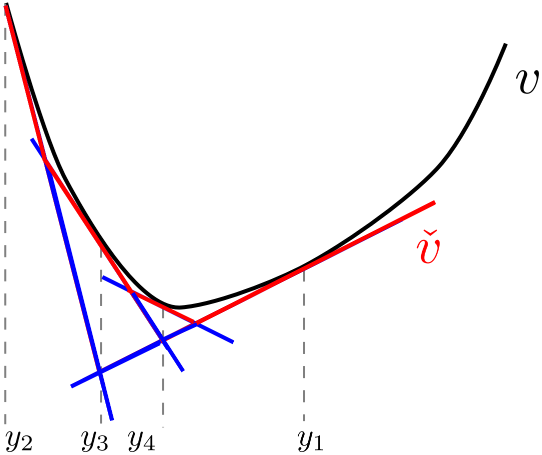


Illustration of Benders method

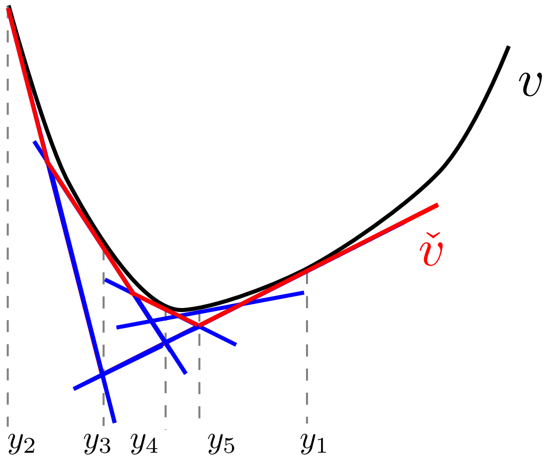


Illustration of Benders method

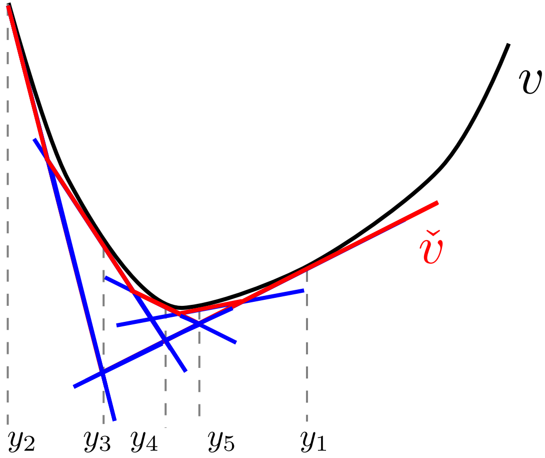
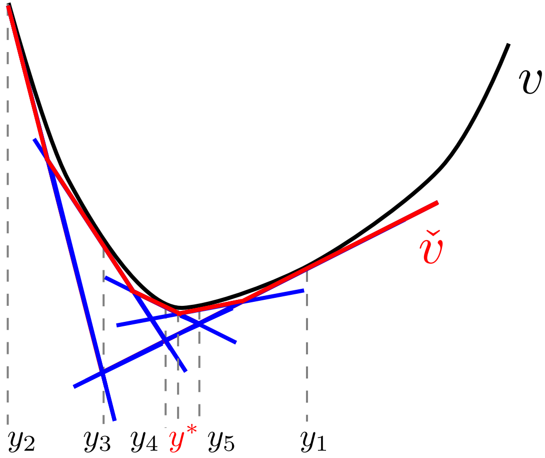
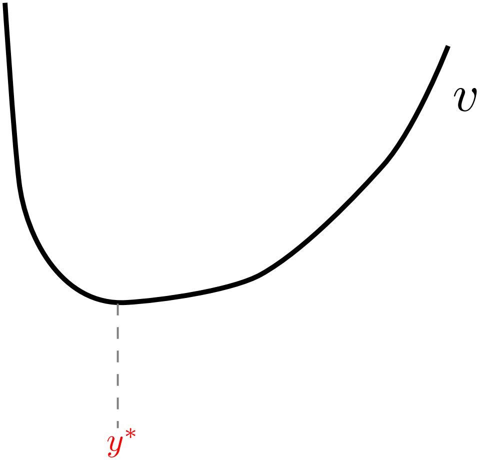


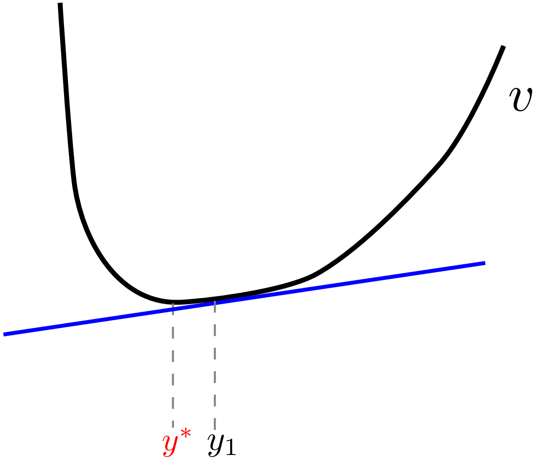
Illustration of Benders method



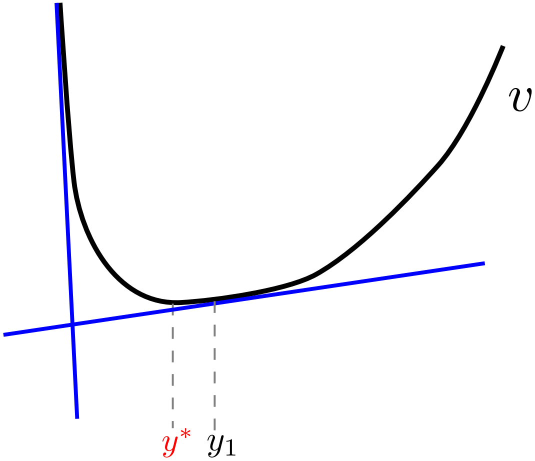
Instability of Benders method



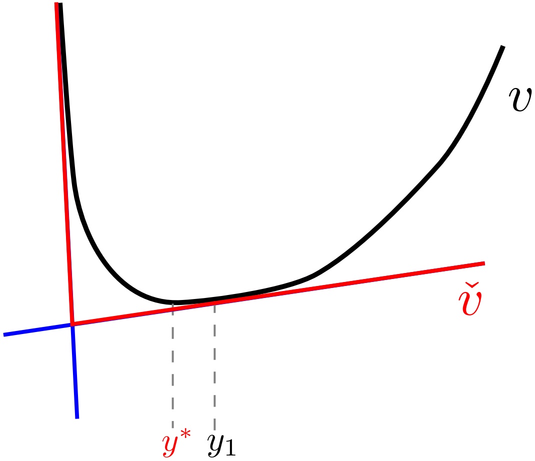
Instability of Benders method



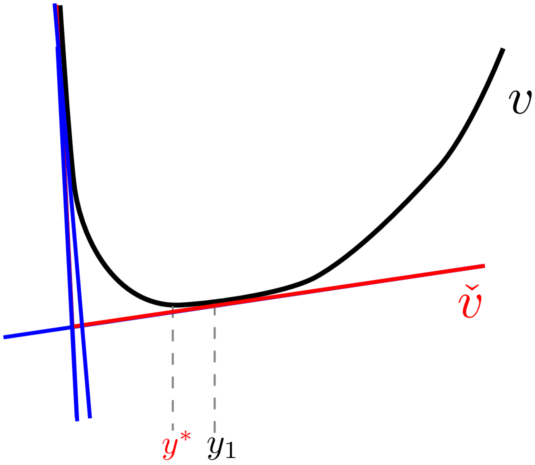
Instability of Benders method



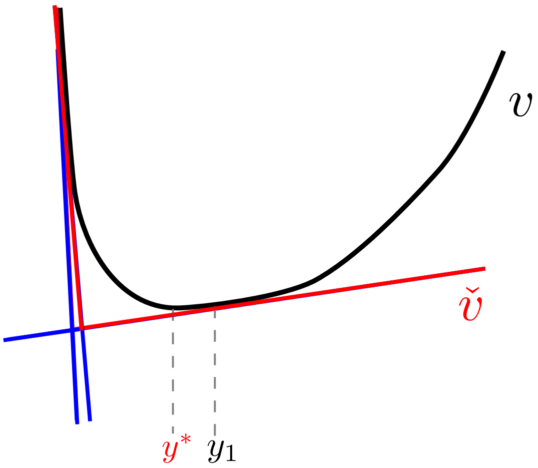
Instability of Benders method



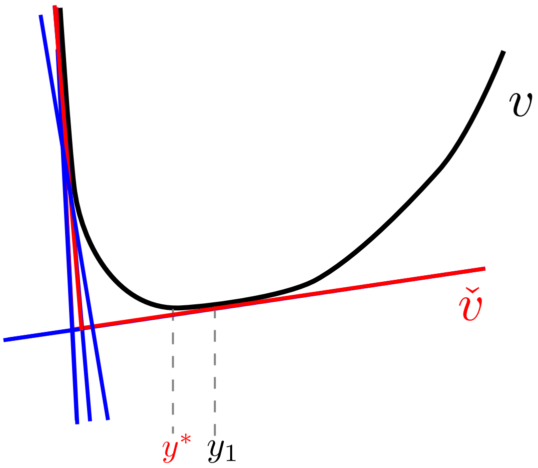
Instability of Benders method



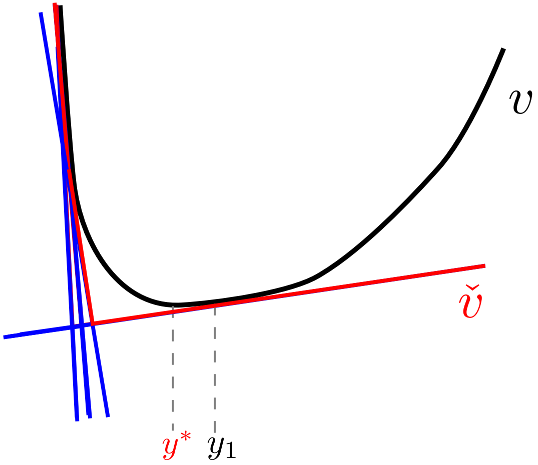
Instability of Benders method



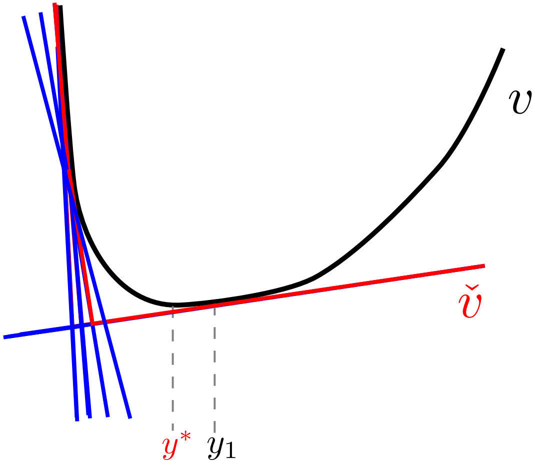
Instability of Benders method



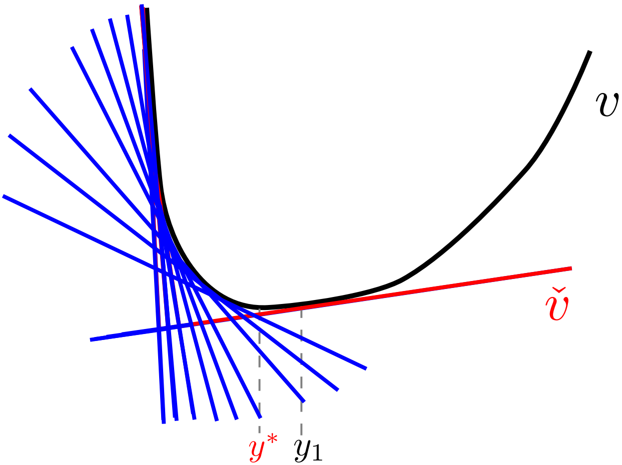
Instability of Benders method



Instability of Benders method



Instability of Benders method



Outline

Stabilizations in the literature

The idea

Make the iterates stay close to the best one $\hat{y}_k$ (where we trust the model)

In nonsmooth convex optimization

- Equivalent of Benders method: Kelley method [Kelley 1960]
- Different stabilizations [Hiriart-Urruty and Lemaréchal 1993]

Proximal bundle	$y_{k+1} \in \arg\min_{y \in Y} \{ \check{v}_k(y) + 1/2t \, \|y - \hat{y}_k\|^2 \}$
Trust region	$y_{k+1} \in \arg\min_{y \in Y} \{ \check{v}_k(y) \text{ s.t. } \|y - \hat{y}_k\|^2 \leq R \}$
Level bundle	$y_{k+1} \in \arg\min_{y \in Y} \{ 1/2 \, \|y - \hat{y}_k\|^2 \text{ s.t. } \check{v}_k(y) \leq L \}$

In MILP, for Dantzig-Wolfe decomposition

Successful stabilization by proximal bundle method [Briant et al 2008]

Our proposal: level version of Benders method

Level master subproblem

$$\begin{array}{ll} \min & \check{v}_k(y) \\ \text{s.t.} & y \in \check{V}_k \end{array} \quad \rightsquigarrow \quad \begin{array}{ll} \min & \frac{1}{2}\|y - \hat{y}_k\|_2^2 \\ \text{s.t.} & \check{v}_k(y) \leq \text{lev}_k \\ & y \in \check{V}_k \end{array}$$

What has changed?

- Integer Linear Program $\rightsquigarrow$ convex Integer Quadratic Program
- But: solvers are now good with convex IQPs
- Optimal value of the master no longer a lower bound
- But: when infeasible, *lev* is a lower bound of the optimal value

Illustration of level-Benders method

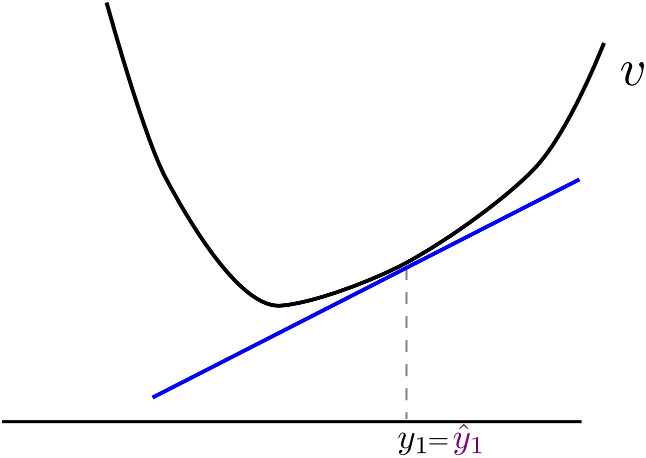


Illustration of level-Benders method

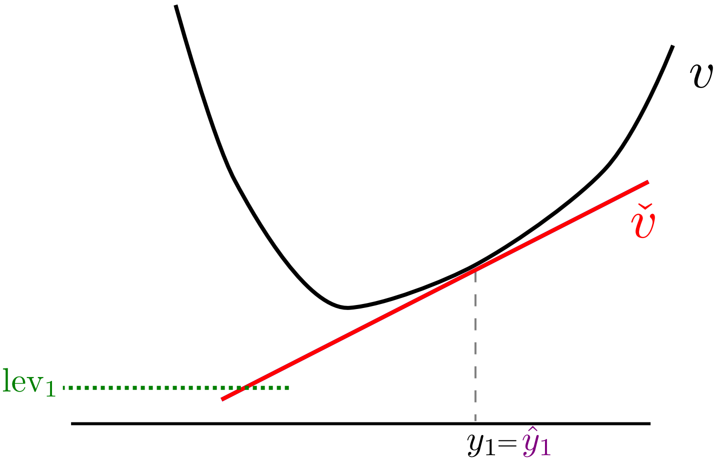


Illustration of level-Benders method

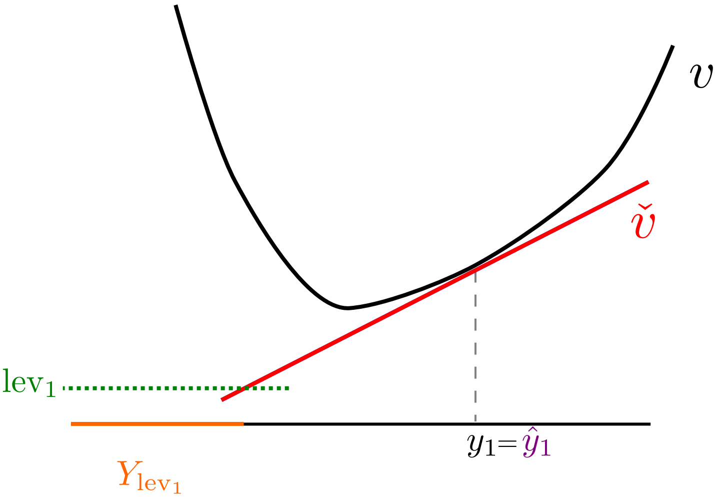


Illustration of level-Benders method

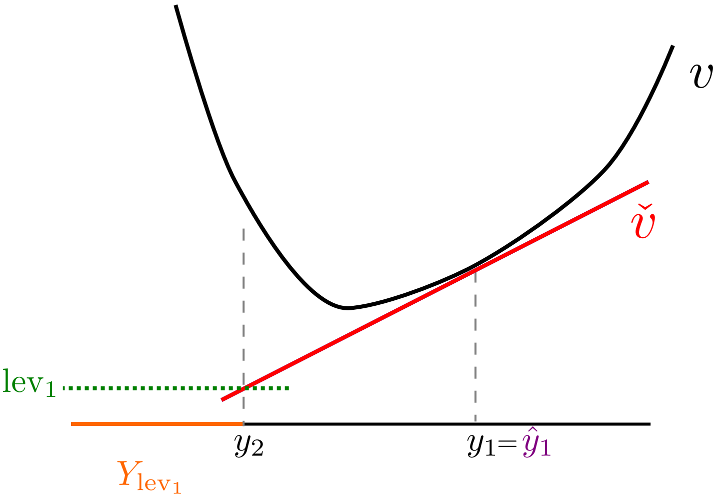


Illustration of level-Benders method

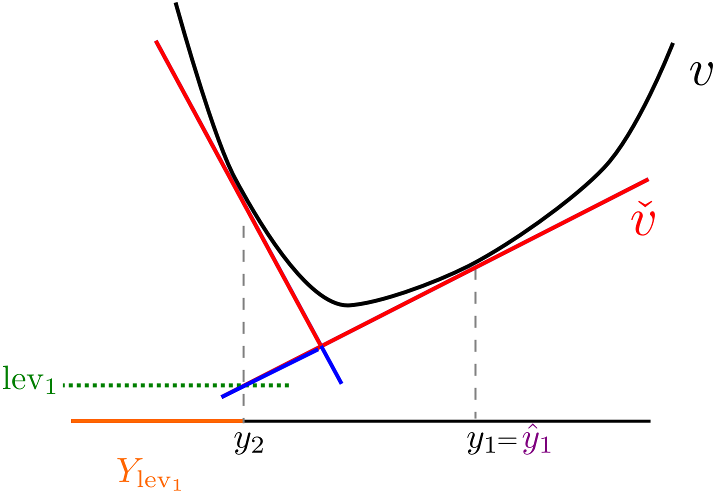


Illustration of level-Benders method

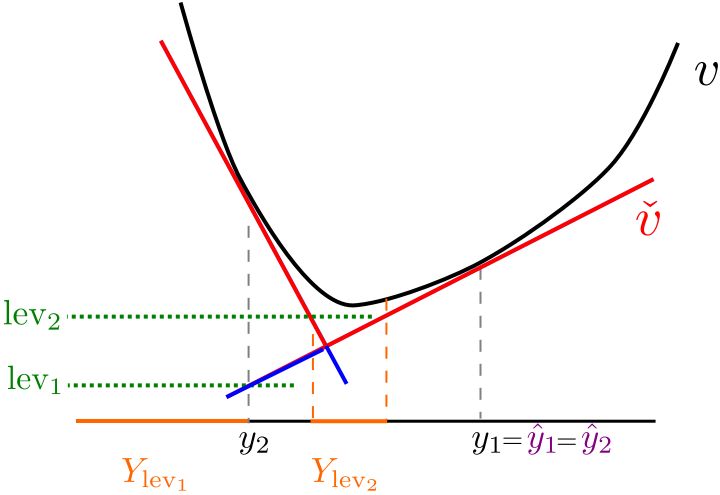


Illustration of level-Benders method

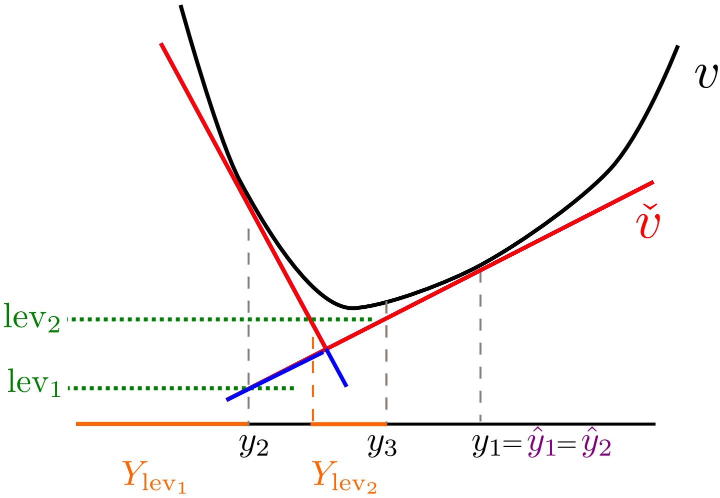


Illustration of level-Benders method

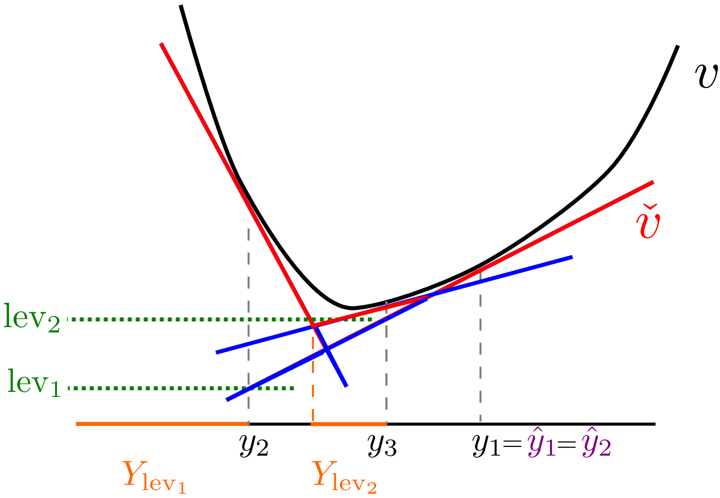


Illustration of level-Benders method

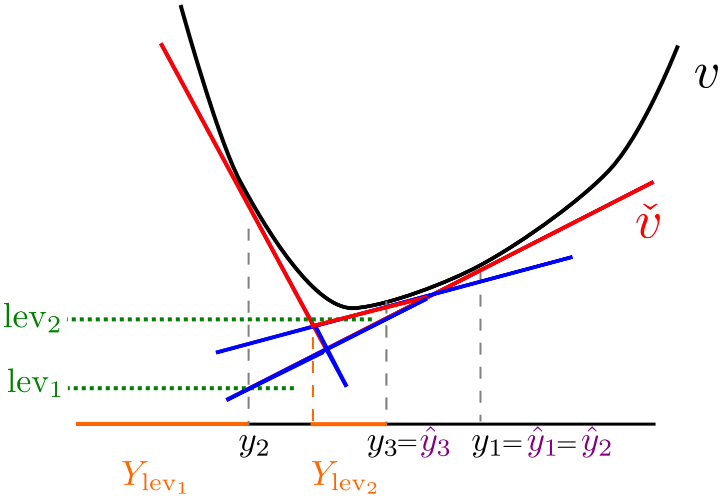
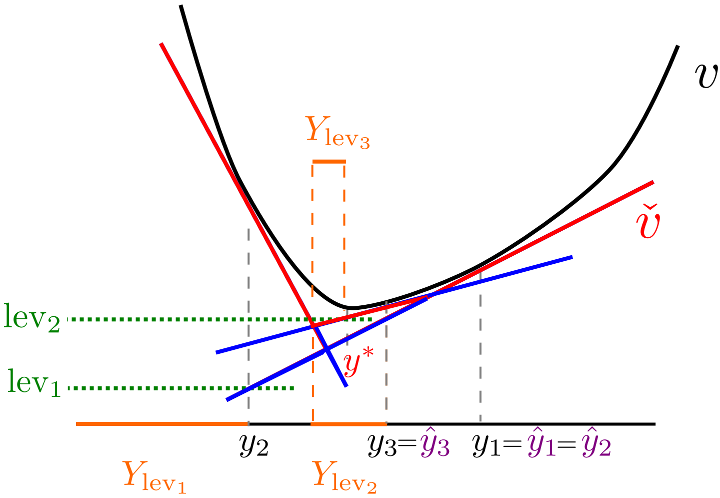
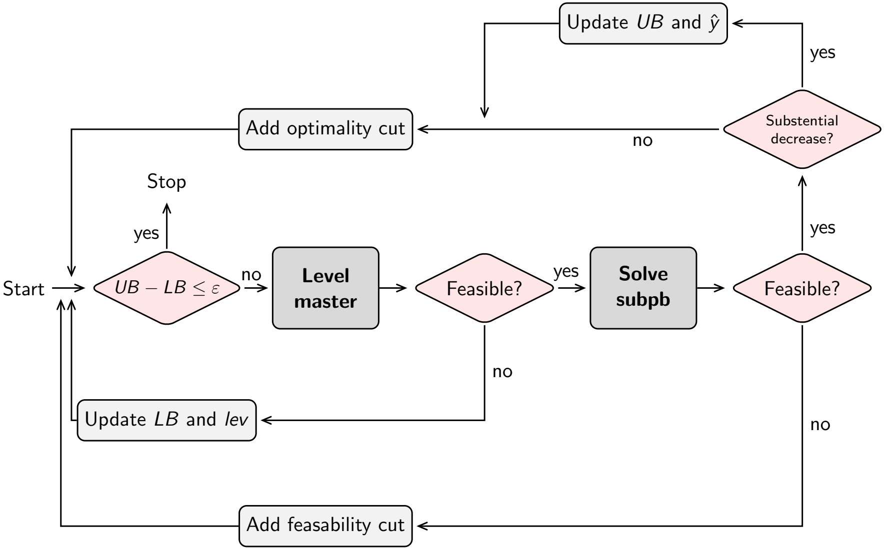


Illustration of level-Benders method



Level Benders algorithm



Convergence analysis

Convergence results are similar to GBD original results.

Theorem 1

If Y compact $\subset \text{dom}v$ then the algorithm asymptotically converges:

$$v(\hat{y}_k) \xrightarrow[k \to \infty]{} v^*$$

Theorem 2

If Y is finite then the algorithm converges in a finite number of iterations:

$$\exists k \;\text{ such that }\; v(\hat{y}_k) = v^*$$

Proof idea

Handle the 4 possible infinite loops:
infeasible master / infeasible subproblem / descent steps / null steps

Outline

Illustration on uncapacited network design problems

Instances

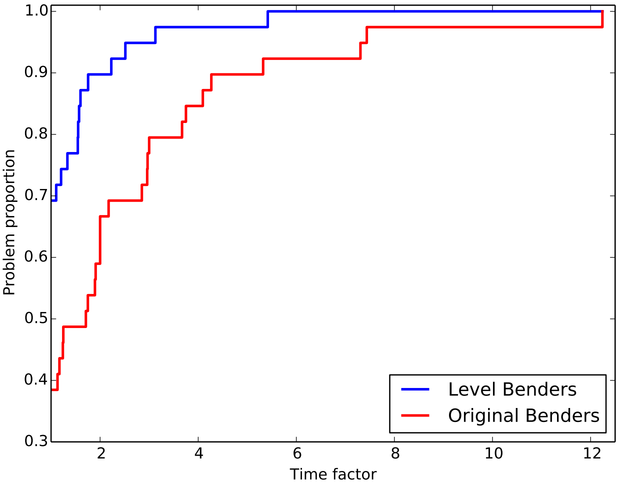
- 40 randomly generated instances
- Number of nodes: $N \leq 15$, number of commodities: $K \leq 15$

Typical results

Instances		Time (s)		Iterations	
N	K	Benders	LB	Benders	LB
5	3	0.11	0.17	14	25
8	5	1.24	1.37	36	33
10	5	8.15	3.75	75	43
10	8	216.20	40.61	218	86
15	5	88.28	11.87	135	46

Preliminary results

Performance profiles of Benders and level-Benders methods



Conclusion

Summary

- Level-benders method:
  - Exploits the MIQP great improvements
  - Same theoretical convergence as original Benders methods
  - Acceleration in practice
- Other nice practical features:
  - Inexact solving of subproblem
  - Limited memory

Compatible further accelerations

- Inexact (multi-)cuts
- Embedding the method in a branch and cut framework

Conclusion

Summary

- Level-benders method:
  - Exploits the MIQP great improvements
  - Same theoretical convergence as original Benders methods
  - Acceleration in practice
- Other nice practical features:
  - Inexact solving of subproblem
  - Limited memory

Compatible further accelerations

- Inexact (multi-)cuts
- Embedding the method in a branch and cut framework

Thank you!