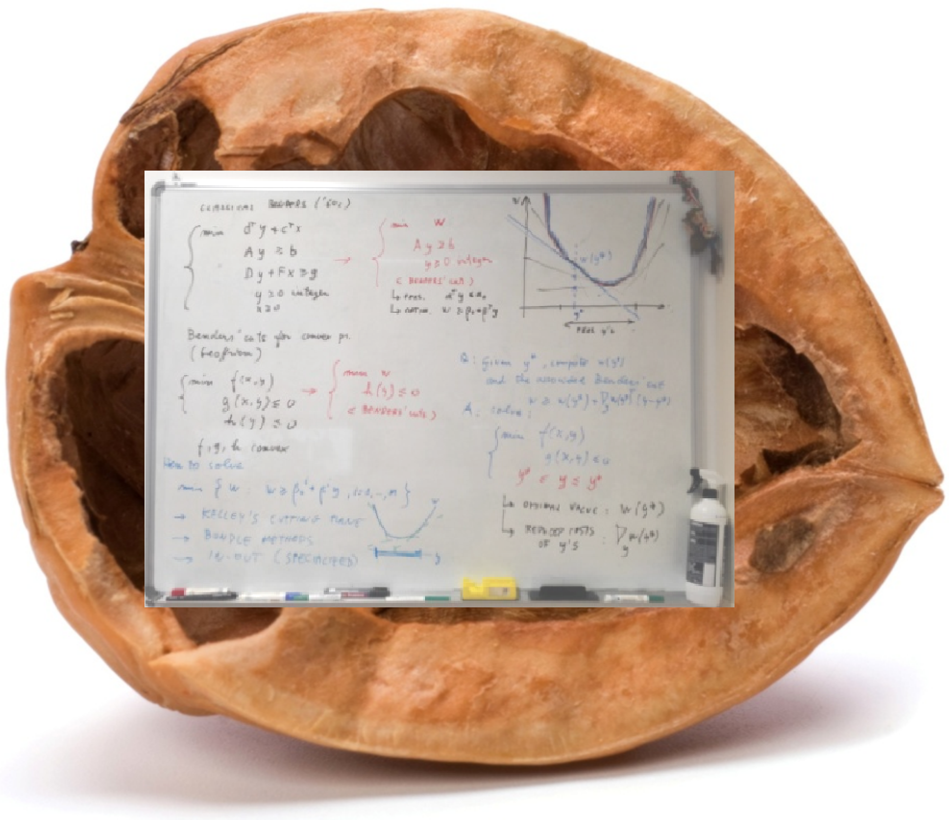
Benders revised

Matteo Fischenders, University of Padova



Montreal, 12 December 2017

Benders iterative method

- Mixed-integer **convex** problem of interest

$$\min f(x,y) \quad g(x,y) \le 0 \quad Ay \le b \quad y \ \text{integer}$$

$\rightarrow$

$$\min \eta \quad f(x,y) \le \eta \quad g(x,y) \le 0 \quad Ay \le b \quad y \ \text{integer}$$

- Continuous var.s *x* “uninteresting” → project them away!
- **Iterative solution procedure**:

  1. solve the **master problem relaxation** by using a black-box MILP solver

$$\min \eta \quad Ay \le b \quad y \ \text{integer}$$

$$\ldots \text{ linear cuts in the } (y,\eta) \text{ space } \ldots$$

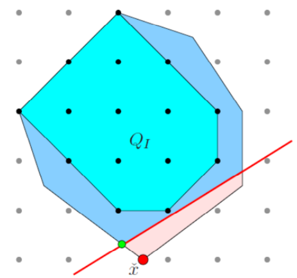
  2. possibly generate new linear cuts in the *(y,η)* space, and repeat

Two distinct ideas

- The original Benders decomposition from the 1960s uses **two** distinct ingredients for solving a Mixed-Integer Linear Program (MILP):

1) A **search strategy** where a relaxed **(NP-hard)** MILP on a variable **subspace** is solved exactly (i.e., to **integrality**) by a black-box solver, and then is iteratively tightened by means of additional **“Benders” linear cuts**

2) The **technicality** of how to actually compute those cuts (Farkas’ projection)



Papers proposing “**a new Benders-like scheme**” typically refer to 1)

Students scared by **“Benders implementations”** typically refer to 2)

Later developments

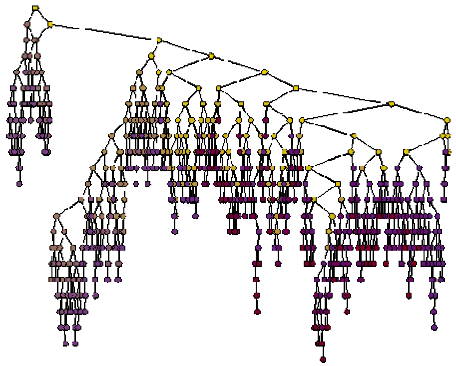
The idea of generating Benders cuts to cut the optimal solution of a MILP was considered not effective (in the 1970's) because "*one wastes a lot of time in solving by enumeration a hard MILP to produce a solution that is immediately cut off*"

- Folklore (Miliotios for TSP?): generate Benders cuts within a **single B&B tree** to cut any infeasible integer solution that is going to update the incumbent

- McDaniel & Devine (1977): use Benders cuts to cut **fractional sol.s** as well (root node only)

- Everything fits very naturally within a modern **Branch-and-Cut** (B&C) framework where Benders cuts are just another source of cutting planes

- **Note**: The original Benders' idea of solving a sequence of MILPs by a **black-box solver** is become more and more appealing due to the dramatic improvement of the MILP technology!



Benders in a nutshell

- Consider again the convex MINLP in the *(x,y)* space

$$\min f(x,y)$$
$$g(x,y) \leq 0$$
$$Ay \leq b$$
$$y \text{ integer}$$

and assume for the sake of simplicity that $S := \{y : Ay \leq b\}$ is nonempty and bounded, and that

$$X(y) := \{x : g(x,y) \leq 0\}$$

is **nonempty**, closed and bounded for all y ∈ S

→ the **convex function** $\Phi(y) := \min_{x \in X(y)} f(x,y)$ is well defined for all y ∈ S

→ no “feasibility cuts” needed (this kind of cuts will be discussed later on)

Working on the y-space (projection)

(1)

$$\min_y \min_x f(x,y)$$
$$g(x,y) \leq 0$$
$$Ay \leq b$$
$$y \text{ integer}$$

(2)

"isolate the inner minimization over *x*"

$$\Phi(y) := \min_x f(x,y)$$
$$g(x,y) \leq 0$$

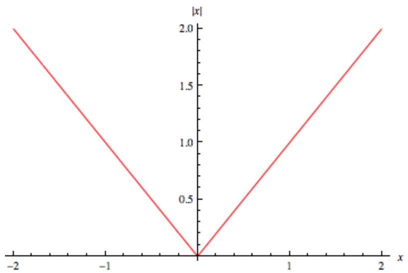
(3)

$$\min \Phi(y)$$
$$Ay \leq b$$
$$y \text{ integer}$$

Original MINLP in the (x,y) space → Benders' **master** problem in the *y* space

Warning: projection changes the objective function (e.g., linear → **piecewise** linear)

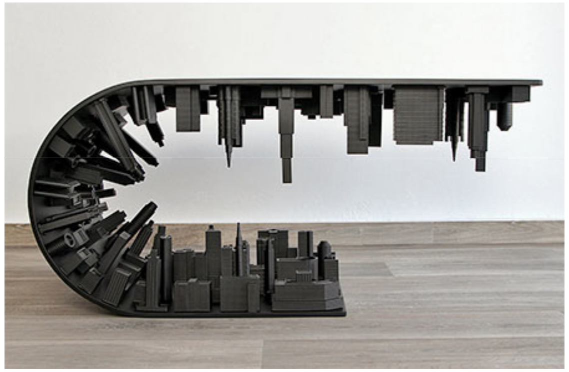
$$\min x$$
$$x \geq y$$
$$x \geq -y$$
$$y \in [-1, 1]$$



$$\min \Phi(y) = |y|$$
$$y \in [-1, 1]$$

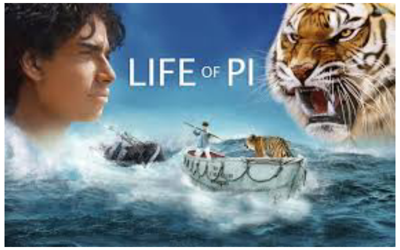
Projection alters the geometry!

The previous example shows that
• even if we start with **linear problem with no integer var.s**
• projection leads to a (convex) **piecewise linear function** with a possibly exponential number of pieces

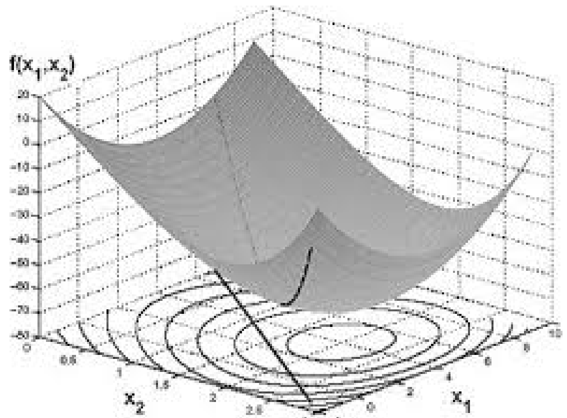


"the inception effect"

Note: A similar effect is obtained by a **Deep Neural Network (DNN)** with ReLU activations that partitions the input space *y* into an exponential number of polyhedra, each corresponding to a linear piece → it relies on "**binary** activation variables" (combinatorial nature of the DNN)



Life of P(H)I

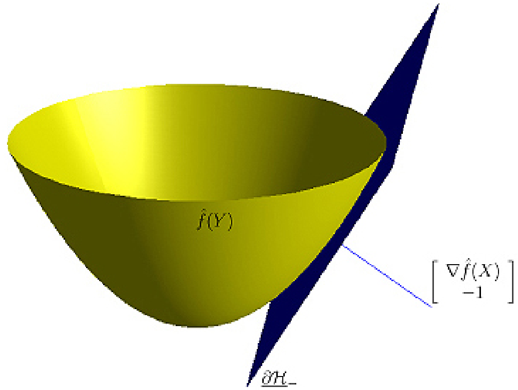


- Solving Benders' master problem calls for the minimization of a **nonlinear** convex function **(even if you start from a linear problem!)**

- Branch-and-cut MINLP solvers generate a sequence of **linear cuts** to approximate this function from below (**outer-approximation**)

$$\begin{aligned} &\min w \\ \text{s.t. } & w \geq \Phi(y) \\ & Ay \leq b \\ & y \text{ integer} \end{aligned}$$

subgradient
(aka Benders) cut →



$$w \geq \Phi(y) \geq \Phi(y^*) + \xi(y^*)^T (y - y^*)$$

Benders cut computation

- **Benders** (for linear) and **Geoffrion** (general convex) told us how to compute a **subgradient** to be used in the cut derivation, by using the optimal primal-dual solution *(x*,u*)* available after computing $\Phi(y^*)$

$$\xi(y^*) = \nabla_y f(x^*, y^*) + u^* \nabla_y g(x^*, y^*)$$

- The above formula is **problem-specific** and perhaps **#scaring**
- Introduce an **artificial variable vector q** (acting as a copy of *y*) to get

$$\Phi(y^*) = \min\{f(x, \mathbf{q}) \mid g(x, \mathbf{q}) \leq 0,\ y^* \leq \mathbf{q} \leq y^*\}$$

and to obtain the following **simpler** and **completely general** cut-recipe:

1) solve the original convex problem with new var. bounds $y^* \leq y \leq y^*$

2) take opt_val and reduced costs r_j's

3) write $w \geq opt\_val + \sum_j r_j (y_j - y_j^*)$

Benders feasibility cuts

- For some important applications, the set

$$X(y) := \{x : g(x,y) \le 0\}$$

can be empty for some “**infeasible**” $y \in S$

→ $\Phi(y) := \min_{x \in X(y)} f(x,y)$ undefined

- This situation can be handled by considering the “phase-1” feasibility condition

$$0 \ge \Psi(y) := \min\{1^T s \,|\, g(x,y) \le s,\ s \ge 0\}$$

where the function $\Psi(y)$ is **convex**
→ it can be approximated by the usual subgradient **“Benders feasibility cut”**

$$0 \ge \Psi(y) \ge \Psi(y^*) + \xi(y^*)^T (y - y^*)$$

to be computed as in the previous **“Benders optimality cut”**

$$w \ge \Phi(y) \ge \Phi(y^*) + \xi(y^*)^T (y - y^*)$$

Successful Benders applications

- Benders decomposition works well when fixing *y* = *y** for computing $\Phi(y^*)$ makes the problem **much simpler to solve**.

- This usually happens when
  - The problem for *y* = *y** decomposes into a number of **independent subproblems**
    - Stochastic Programming
    - Uncapacitated Facility Location
    - etc.

$$
\begin{aligned}
\min \sum_{i\in I} f_i y_i + \sum_{i\in I}\sum_{j\in J} c_{ij} x_{ij} & \\
\text{s.t. } \sum_{i\in I} x_{ij} = 1 & \quad \forall j \in J \\
x_{ij} \le y_i & \quad \forall i \in I, j \in J \\
x_{ij} \ge 0 & \quad \forall i \in I, j \in J \\
y_i \in \{0,1\} & \quad \forall i \in I
\end{aligned}
$$

  - Fixing *y* = *y** **changes the nature** of some constraints:
    - in Capacitated Facility Location, tons of constr.s of the form $x_{ij} \le y_j$ become just variable bounds
    - Second Order Constraints $x_{ij}^2 \le z_{ij}\, y_i$ become quadratic constr.s
    - etc.

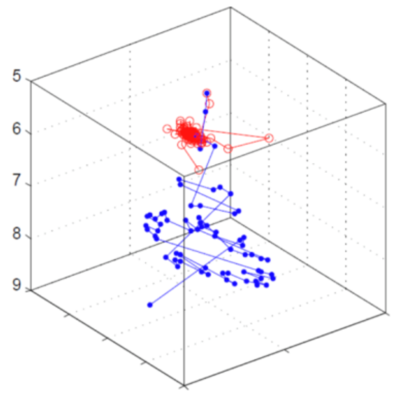
Stabilization

- In practice, Benders decomposition can work quite well, but sometimes it is **desperately slow**

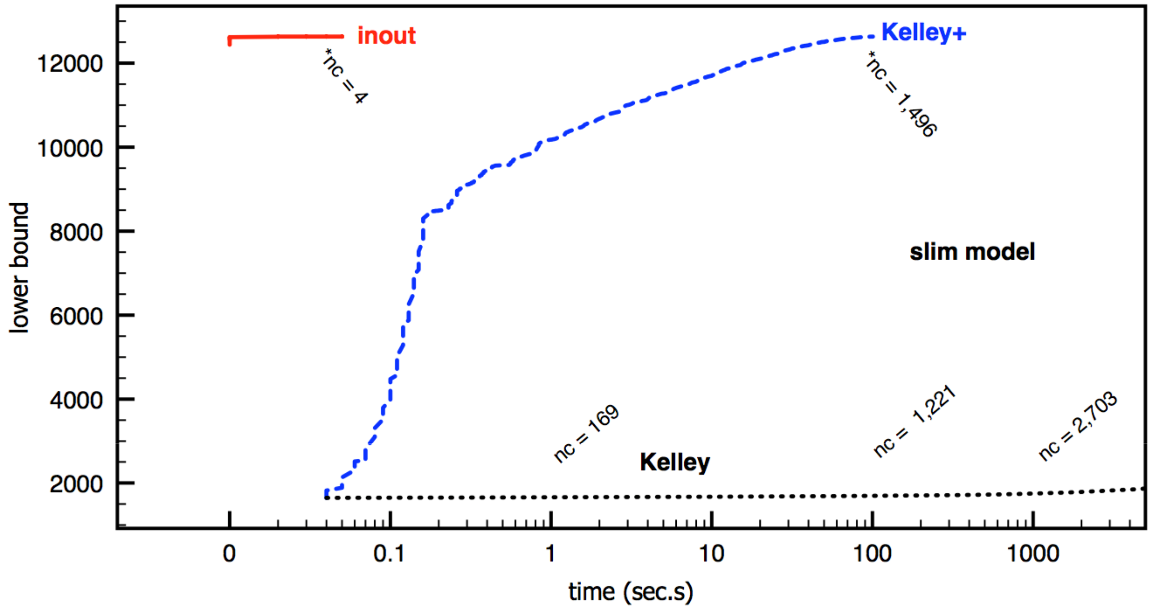
  … as the root node bound does not improve even after the addition of tons of Benders cuts

- Slow convergence is generally attributed to the **poor quality** of Benders cuts, to be cured by a more clever **selection policy** (Pareto optimality of Magnanti and Wong, 1981, etc.) but …

- … the main culprit is often a zig-zagging effect to be cured by **stabilization** methods such as **bundle** (Lemaréchal) or **in-out** (Ben-Ameur and Neto) or **local branching** (Rei, Cordeau, Gendreau, Soriano)



Effect of stabilization



- Comparing **Kelley** cut loop at the root node with **Kelley+** (add epsilon to y*) and with a stabilized method (**inout**)
- Koerkel-Ghosh **qUFL** instance gs250a-1 (250x250, quadratic costs)
- ***nc** = n. of Benders cuts generated at the end of the root node
- times in **logarithmic scale**

Conclusions

To summarize:

- Benders cuts are **easy** to implement within modern B&C (just use a callback where you solve the problem for $y = y^*$ and compute reduced costs)

- It can be **desperately slow** hence stabilization is a **must**

- Implementations in **general** MIP solver available in Cplex 12.7

- The **“old-Benders”** approach (using a black-box MILP solver) can strike again

Slides: http://www.dei.unipd.it/~fisch/papers/slides/

Reference papers:

M. Fischetti, I. Ljubic, M. Sinnl, "Benders decomposition without separability: a computational study for capacitated facility location problems", European Journal of Operational Research, 253, 557-569, 2016.

M. Fischetti, I. Ljubic, M. Sinnl, "Redesigning Benders Decomposition for Large Scale Facility Location", Management Science 63(7), 2146-2162, 2016.