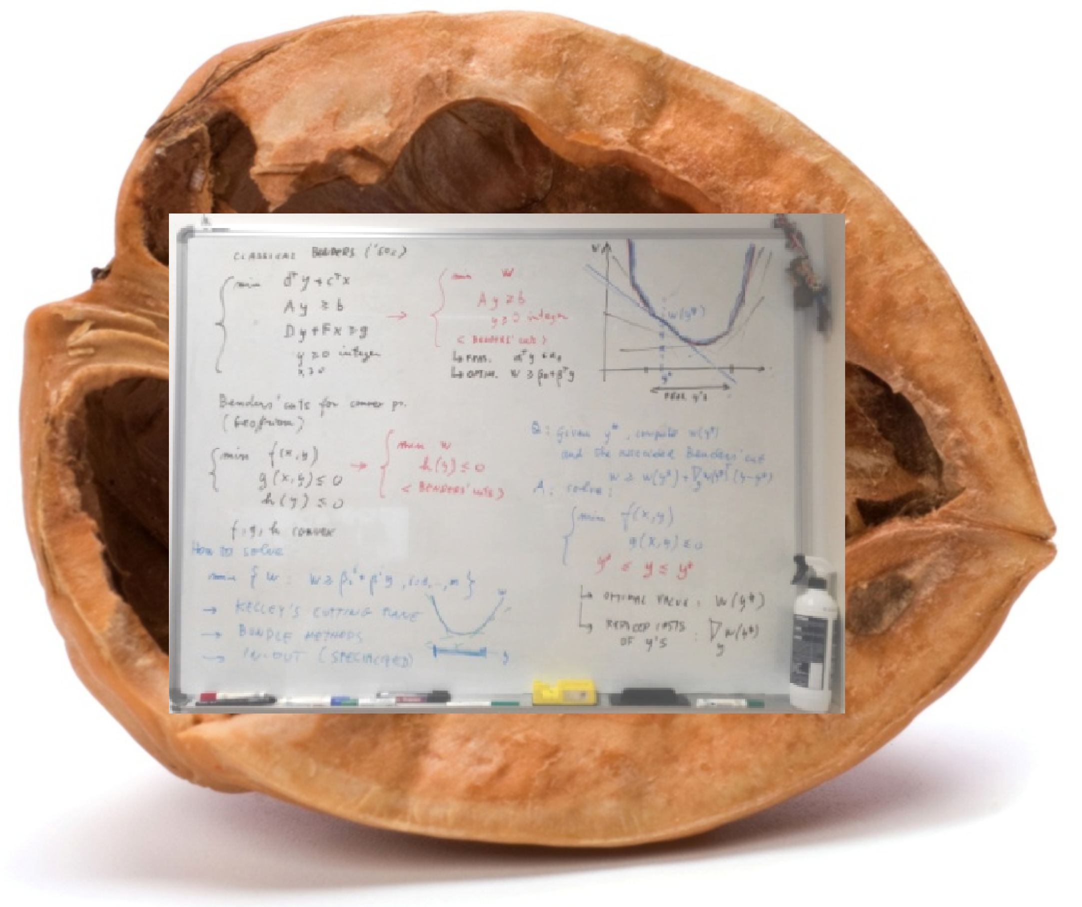
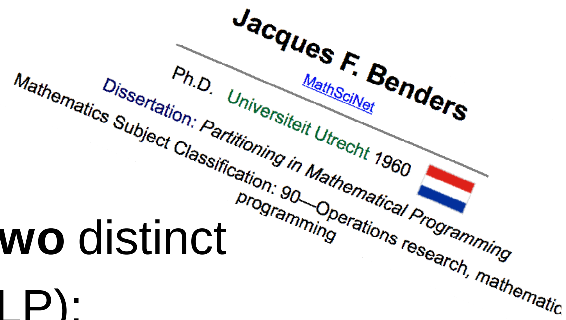
Benders in a nutshell

Matteo Fischetti, University of Padova



Benders decomposition



- 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 in the 1970s:
  - 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.

Branch-and-Cut (B&C)

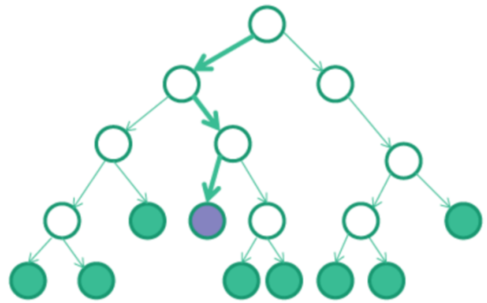
- B&C was proposed by Padberg and Rinaldi in the 1990s and is nowadays the **method of choice** for solving MIPs

- B&C is a clever **mixture** of cutting-plane and branch-and-bound methods

- **Since the beginning, an highly-effective implementation was part of the B&C trademark (use of cut pool, global vs local cuts, variable pricing, etc.)**

- Modern commercial B&C solvers such as IBM ILOG Cplex, Gurobi, XPRESS etc. can be fully **customized** by using ***callback functions***

- Callback functions are just **entry points** in the B&C code where an advanced user **(you!)** can add his/her customized **cuts**, heuristics, etc.



Benders cuts

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

$$\min f(x,y)$$

$$g(x,y) \leq 0$$

$$Ay \leq b$$

$$y \text{ integer}$$

and let that the convex function

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

$$g(x,y) \leq 0$$

is well defined for every *y* in $S := \{y : Ay \leq b\}$

→ no "**Benders feasibility cut**" needed (otherwise, see the full paper)

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*"

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

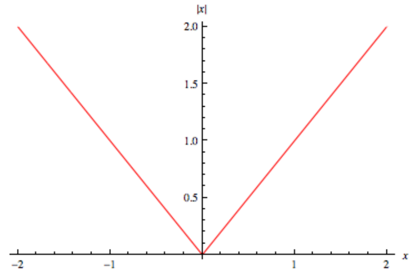
(3)

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

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

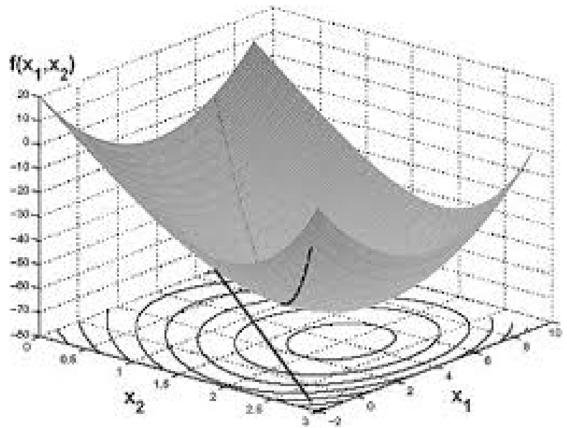
Warning: projection changes the objective function (e.g., linear → convex nonlinear)

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



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

Outer-approximation of the Φ function

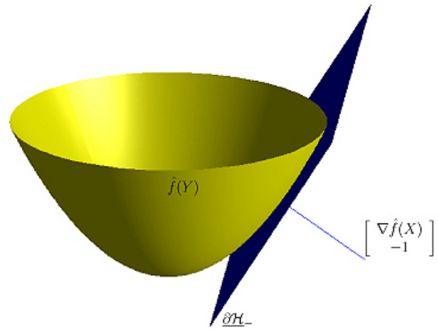


- Solving Benders' master problem calls for the minimization of a **nonlinear** convex function **(even if you start from a linear problem!)**

- 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 at a given *y**
→ Taylor's first order approx.



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

Benders cut separation

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

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

- We propose the use of the following **simpler** and **completely general** recipe to generate a (most violated) Benders cut for a given *y**.

1) solve the original convex problem with new var. bounds $y^* \le y \le y^*$
2) take opt_val and reduced costs r_j's
3) write $w \ge opt\_val + \sum_j r_j (y_j - y_j^*)$

Benders feasibility cuts

- For some important applications, the set

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

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

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

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

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

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

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

to be computed as in the previous **"Benders optimality cut"**

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

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.

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

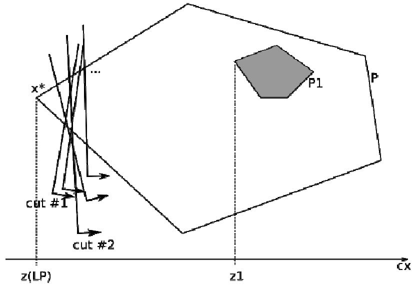
  - Fixing $y = y^*$ **changes the nature** of some constraints:
    - in Capacitated Facility Location, tons of constr.s of the form $x_{ij} \le y_j$ become just variable bounds
    - Second Order Cone const.s $x_{ij}^2 \le z_{ij}\, y_i$ become quadratic
    - etc.

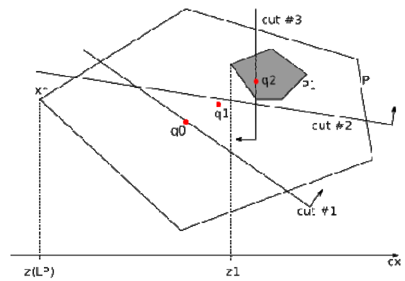
Cut-loop stabilization

- In practice, Benders decomposition can work quite well, but sometimes it is **desperately slow** at the root node (the lower bound does not improve even after the addition of tons of Benders cuts)

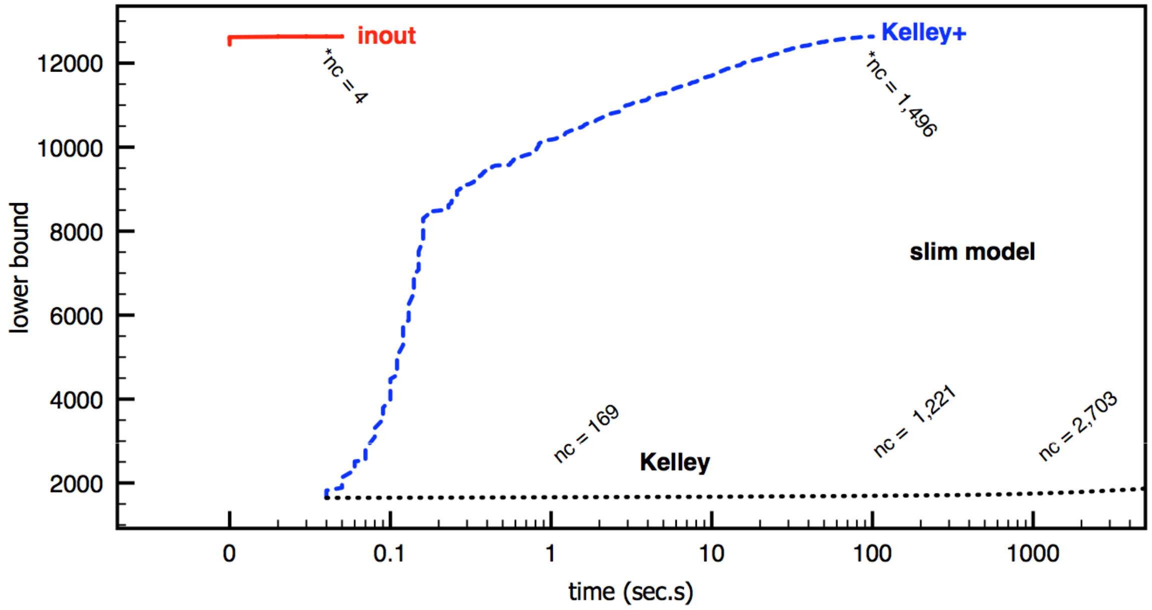
- Slow convergence is generally attributed to the **poor quality** of Benders cuts, to be cured by a more clever **selection policy** (Pareto optimality of Magnanti and Wong, 1981, etc.)

- An important role is also played by the **Kelley's cut loop** (always cut an optimal vertex of the current master problem) → try alternative schemes that cut an **internal point** of the master relaxation (analytic center, bundle, in-out, etc.)





Effect of the improved cut loop



- Comparing **Kelley** cut loop at the root node with **Kelley+** (add epsilon to y*) and with our chase-the-carrot 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)

- Kelley's cut loop can be **desperately slow** hence stabilization is a **must**

- Suitable for **general** MIP solvers (already in Cplex 12.7)

Slides available at 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", to appear in Management Science, 2016.