Local Branching
– *A brand new method*

Thomas Stidsen

tks@imm.dtu.dk

Informatics and Mathematical Modeling
Technical University of Denmark

Outline

- Heuristics for Mixed Integer Programs
  - Rounding
  - Local branching
  - Relaxation induced neighbourhood search
  - Guided dives
  - Experimental results

The OR Work Process

Typically OR people would:

- Analyze problem
- Create MIP model
- Solver fails, now what ?
  - Change model
  - Use decomposition
  - Create meta-heuristic

Problem (according to David Ryan): People far to often just switch to meta-heuristics.

Meta-heuristics and MIP

There are two reasons why switching to meta-heuristics are problematic:

- Some work (the MIP model) is discarded ...
- Handling constraints in meta-heuristics is **hard**: Set Partitioning, Metsa Öy ...

Alternatively we can use MIP-heuristics: Rounding, RINS or Local Branching. These are very new (2003), but not very complicated, techniques ...

Mixed integer programming (MIP)

$$
\begin{aligned}
\text{minimize} \quad & \sum_{j=1}^{n} c_j x_j \\
\text{subject to} \quad & \sum_{j=1}^{n} a_{ij} x_j \geq b_i, && i = 1, \ldots, m \\
& x_j \in \{0, 1\}, && j \in \mathcal{B} \\
& x_j \geq 0, \text{integer} && j \in \mathcal{G} \\
& x_j \geq 0, && j \in \mathcal{C}
\end{aligned}
$$

Local branching

Given a MIP model:

- Construct a feasible solution (may be hard ...)
- Add a "locality" constraint, which limits the search area around the current feasible solution $\bar{x}$ ...
- Enter the main loop
- Find the new best around $\bar{x}$
- Switch to the new solution
- Continue until no better solutions can be found

What's a locality constraint ???

Let $\overline{\mathcal{B}} := \{j \in \mathcal{B} : \bar{x}_j = 1\}$ be the binary support of $\bar{\mathbf{x}}$. Hamming distance between $\bar{\mathbf{x}}$ and $\mathbf{x}$:

$$\Delta(\mathbf{x}, \bar{\mathbf{x}}) := \sum_{j \in \mathcal{B}} |x_j - \bar{x}_j| = \sum_{j \in \overline{\mathcal{B}}} (1 - x_j) + \sum_{j \in \mathcal{B} \setminus \overline{\mathcal{B}}} x_j$$

k-OPT neighbourhood of $\bar{\mathbf{x}}$: Set of solutions $\mathbf{x}$ for which

$$\Delta(\mathbf{x}, \bar{\mathbf{x}}) \leq k$$

Local branching: Algorithm

How do we search a k-OPT neighboorhood? Just add constraint $\Delta(\mathbf{x}, \bar{\mathbf{x}}) \leq k$ to the MIP, and solve the reduced MIP to optimality!

Local branching: Algorithm II

- When k is small, the new MIP is usually easy to solve.
- May use a truncation scheme, i.e., a time- or node-limit for the solution of the reduced MIP.
- Tactical branching: Having explored the neighbourhood $\Delta(\mathbf{x}, \bar{\mathbf{x}}) \leq k$, we may add the constraint $\Delta(\mathbf{x}, \bar{\mathbf{x}}) \geq k + 1$ to the MIP.

Experimental results

Local branching final comments:

- Again: An algorithm which can solve problems **NON** of the metaheuristics can solve ...
- Relatively easy to implement in e.g. GAMS ...