Strong Branching

1 Arbeitsauftrag

1. Lesen Sie den Artikelauszug unten.
2. Diskutieren Sie in der Gruppe: Welche Vor- und Nachteile hat die vorgestellte Methode? Würden Sie das Verfahren in der Praxis verwenden? Welche Einschränkungen würden Sie machen? Welche Veränderungen würden Sie durchführen?
3. Bereiten Sie eine kurze Präsentation vor, in der Sie den anderen Teilnehmern das Verfahren vorstellen, Vor- und Nachteile diskutieren und auf Ihre Verbesserungsvorschläge eingehen.

2 Auszug aus „Branching rules revisited“ ([AKM05])

When choosing a branching strategy in a branch-and-bound algorithm, we have two decisions to make:

1. How do we split a problem? (branching rule)
2. Which (sub)problem do we select next? (node selection rule)

In the following, we focus on the first question and present a branching rule called *strong branching*. Consider the integer linear program (ILP):

$$c^* = \min\ c^T x, Ax \leq b, x \in \mathbb{Z}^n, \text{ with } A \in \mathbb{R}^{m\times n},\ b \in \mathbb{R}^m,\ c \in \mathbb{R}^n.$$

We use the following notation. X_{ILP} denotes the set of feasible solutions for the given ILP problem. Given an LP problem Q, we define $\bar{c}_Q$ to be the objective value of the optimal solution $\bar{x}_Q$ in Q. If it is clear from the context, we omit Q from all parameters and write $\bar{c}$ and $\bar{x}$ instead of $\bar{c}_Q$ and $\bar{x}_Q$, respectively.

The only way to split a problem Q within an LP-based B&B algorithm is to branch on linear inequalities in order to keep the property of having an LP relaxation at hand. The easiest and most common inequalities are trivial inequalities, i. e., inequalities that split the feasible interval of a singleton variable. To be more precise, if i is some variable with a fractional value $\bar{x}_i$ in the current optimal LP solution, we set $f_i^+ = \lceil \bar{x}_i \rceil - \bar{x}_i$ and $f_i^- = \bar{x}_i - \lfloor \bar{x}_i \rfloor$. We obtain two subproblems, one by adding the trivial inequality $x_i \leq \lfloor \bar{x}_i \rfloor$ (called the *left subproblem*, denoted by Q_i^-) and one by adding the trivial inequality $x_i \geq \lceil \bar{x}_i \rceil$ (called the *right subproblem*, denoted by Q_i^+). This rule of branching on trivial inequalities is also called *branching on variables*, because it only requires to change the bounds of variable i.

The basic variable selection process is described by algorithm 1 given below. Different variable selection rules differ in how the score in Step 2 is computed.

The ultimate goal is to find a fast branching strategy that minimizes the number of B&B nodes that need to be evaluated. Since a global approach is unlikely, one tries to find a branching variable that is at least a good choice for the current branching. The quality of a branching is measured by the change in the objective function of the LP relaxations of the two children Q_i^- and Q_i^+ compared to the relaxation of the parent node Q. In order to compare branching candidates, for each candidate

Input: Current subproblem Q with an optimal LP solution $\bar{x} \notin X_{ILP}$.
Output: An index i of a fractional variable $\bar{x}_i \notin \mathbb{Z}$

Let $C = \{i \mid \bar{x}_i \notin \mathbb{Z}\}$ be the set of branching candidates. ;
For all candidates $i \in C$, calculate a score value $s_i \in \mathbb{R}$.;
Return an index $i \in C$ with $s_i = \max_{j \in C}\{s_j\}$.;

Algorithm 1: generic variable selection

the two objective function changes $\Delta_i^- := \bar{c}_{Q_i^-} - \bar{c}_Q$ and $\Delta_i^+ := \bar{c}_{Q_i^+} - \bar{c}_Q$ are mapped on a single score value. This is typically done by using a function of the form

$$\operatorname{score}(q^-, q^+) = (1 - \mu) \cdot \min\{q^-, q^+\} + \mu \cdot \max\{q^-, q^+\}.$$

The *score factor* μ is some number between 0 and 1 that is usually an empirically determined constant (we use $\mu = 1/6$). Note that special treatment is necessary, if one of the subproblems Q_i^- or Q_i^+ is infeasible.

In the forthcoming explanations all cases are symmetric for the left and right subproblem. Therefore we will only consider one direction, the other will be analogous.

Strong Branching

The idea of *strong branching* is to test which of the fractional candidates gives the best progress before actually branching on any of them. This test is done by temporarily introducing a lower bound $\lfloor \bar{x}_i \rfloor$ and subsequently an upper bound $\lceil \bar{x}_i \rceil$ for variable i with fractional LP value $\bar{x}_i$, and solving the linear relaxations. Using $s_i = \operatorname{score}(\Delta_i^-, \Delta_i^+)$ in Algorithm 1 yields what is called *strong branching*. Strong branching can be viewed as finding the locally (with respect to the given score function) best variable to branch on.

Literatur

[AKM05] Tobias Achterberg, Thorsten Koch und Alexander Martin. „Branching rules revisited". In: *Operations Research Letters* 33.1 (2005), S. 42–54. URL: `http://www.sciencedirect.com/science/article/B6V8M-4CPVPMV-1/2/9dc00fc6305ab19587623ea7b982d23f`.