# Branching with Less Repetition

Carnegie Mellon Tepper School of Business

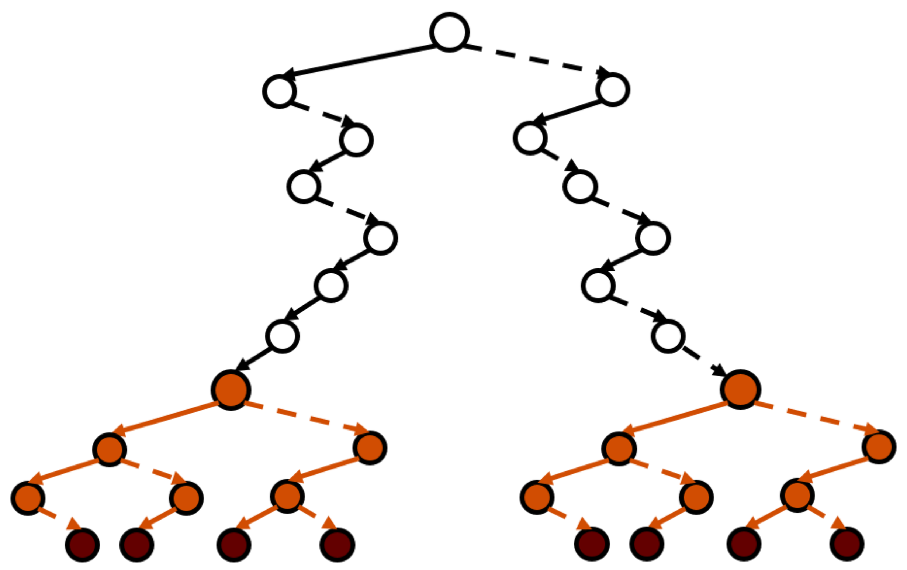
Thiago Serra

tserra@cmu.edu @thserra

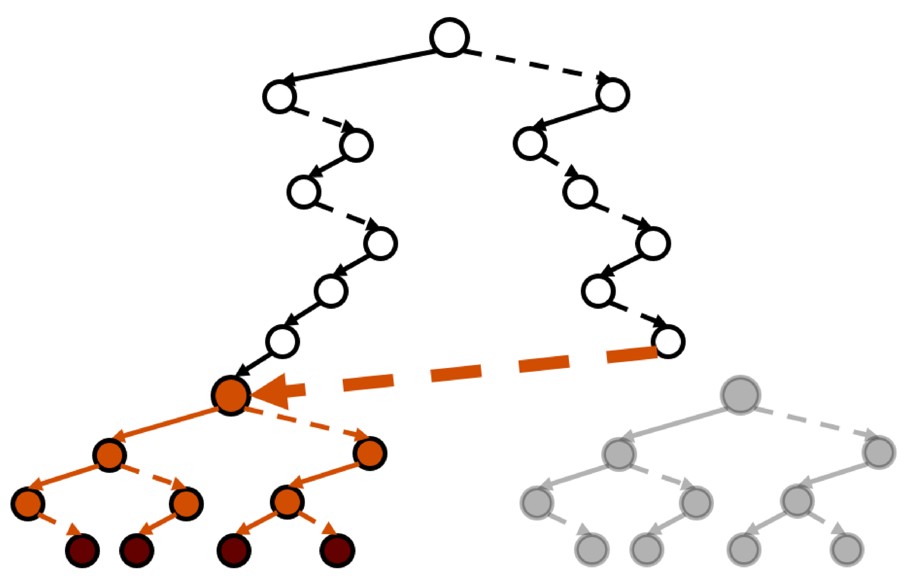
## 1. Collecting IP Solutions

We can collect near-optimal solutions of an integer program by branching to exhaustion, like in the one-tree approach [Danna et al., 2007] used by CPLEX solver [IBM Corp., 2013].

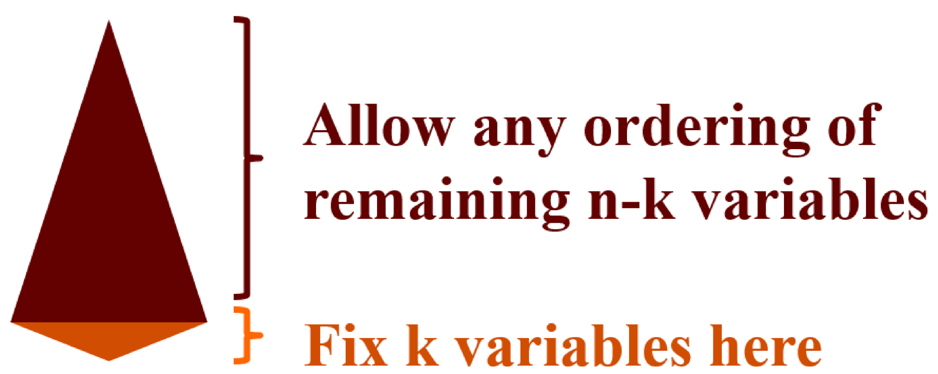
That entails **finding solutions one-by-one**, which **takes too long if there are too many**. Our goal is to remedy that by **identifying subproblems with the same completions**.



We might **add multiple solutions in a single step**, branch less, and possibly finish sooner.



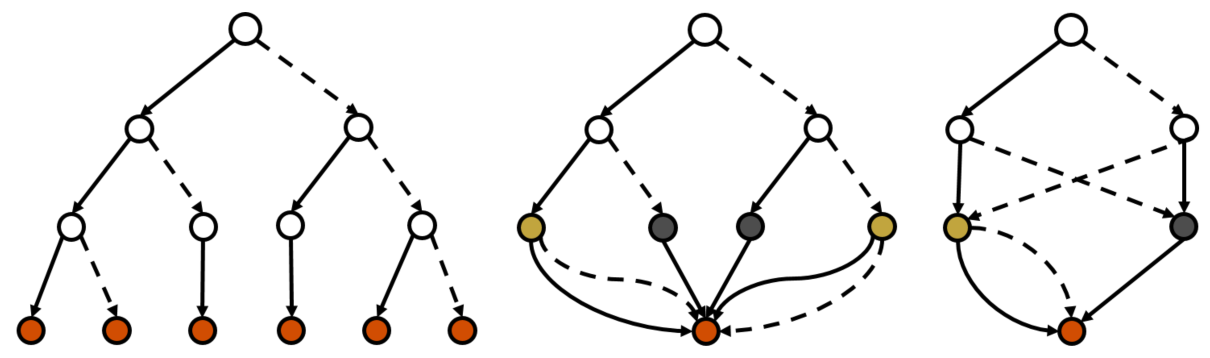
We assume subproblems on the same variables. Branching is not usually done that way, but we **save a set of $k$ variables for branching last**.




In fact, **most equivalences are at the bottom**. For $n-k$ fixed binary variables, there are $O(2^{n-k})$ partial solutions. For $k$ binary variables left, there are $O(2^{2^k})$ completion sets.

## 2. Turning a Branching Tree into a Decision Diagram

A bottom-up pass suffices to merge all nodes with isomorphic completions [Bryant, 1986]:



We can check if nodes root isomorphic subtrees without branching again. After exploring

$$\sum_{i=1}^{n} a_i x_i \leq a_0, \qquad a_0, \ldots, a_n \in \mathbb{Z}, \tag{1}$$

we find **all values $[\beta, \gamma]$ equivalent to $a_0$** [Behle, 2007, Abío et al., 2012, Abío and Stuckey, 2014].

## 3. Equivalence for Single Inequalities

We generalize that to **additively separable inequalities with fractional coefficients**:

$$\sum_{i=1}^{n} f_i(x_i) \leq \rho \tag{2}$$

For a partial solution $\bar{x}_1, \ldots, \bar{x}_{n-k}$, we have

$$\sum_{i=n-k}^{n} f_i(x_i) \leq \rho - \sum_{i=1}^{n-k} f_i(\bar{x}_i) \tag{3}$$

We can compute the **equivalent Right-Hand Side (RHS) values** $[\beta, \gamma]$ for $x_1$ to $x_{n-k}$ fixed, and later compare it with other subproblems:

$$\beta = \max \left\{ \sum_{i=n-k}^{n} f_i(x_i) \;\middle|\; x \text{ is feasible for (3)} \right\} \tag{4}$$

$$\gamma = \min \left\{ \sum_{i=n-k}^{n} f_i(x_i) \;\middle|\; x \text{ is infeasible for (3)} \right\} \tag{5}$$

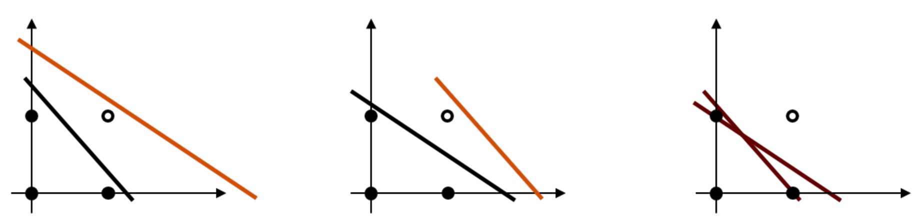
## 4. Equivalence for Multiple Inequalities

A partial solution on $m$ inequalities as those above defines a subproblem of the form

$$\sum_{i=n-k}^{n} f_i^j(x_i) \leq \rho_j - \sum_{i=1}^{n-k} f_i^j(\bar{x}_i), \quad \forall j = 1, \ldots, m \tag{6}$$

**Equivalent RHS values are interdependent**, but there is a **unique tighest RHS vector**:

| | | |
|---|---|---|
| $5x_1 + 4x_2 \leq 6$ | $\leq$ **10** | $\leq$ **5** |
| $6x_1 + 10x_2 \leq$ **17** | $\leq 11$ | $\leq$ **10** |
| $x_1, x_2 \in \{0,1\}$ | | |



For the $j$-th inequality, the tightest RHS is

$$\beta_j = \max \left\{ \sum_{i=n-k}^{n} f_i^j(x_i) \;\middle|\; x \text{ is feasible for (6)} \right\} \tag{7}$$

Given an unexplored node $u$ with RHS $\rho^u$ and an explored node $v$ with tightest RHS $\beta^v$:

**❶ Does $v$ define a subset of $u$?**

Iff $\beta^v \leq \rho^u$

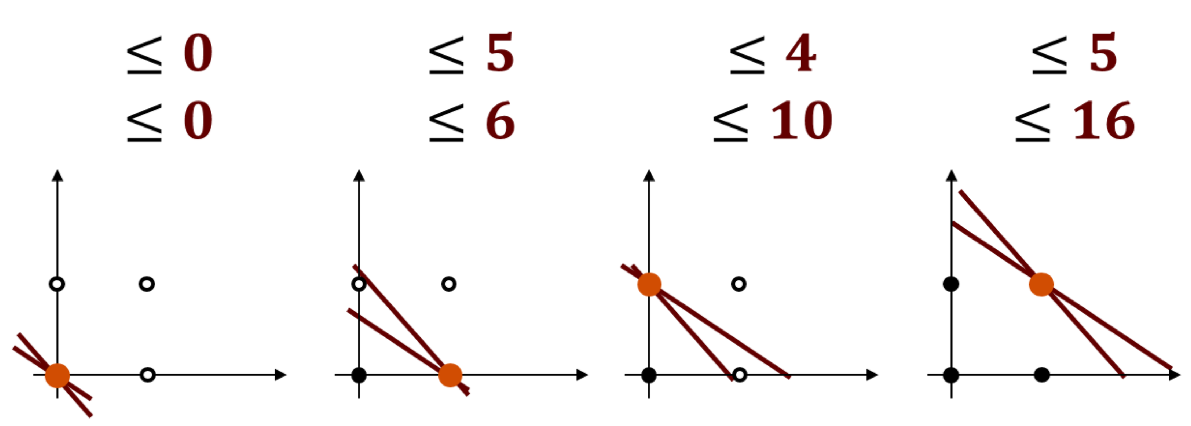
**❷ Does $v$ define the dominating subset of $u$?**

Iff ❶ & $\beta^v \geq \beta^{v'}$ for any node $v'$ satisfying ❶

**❸ Does $v$ define a subset that is not proper?**

Iff ❷ & explored nodes span all completion sets

We generate $O(2^k)$ nodes to span all those sets: one for **the tightest RHS** of **each completion**.




## 5. Results

Our approach is implemented on CPLEX through callbacks. For now, we compare it with one-tree upon fixed variable ordering.

Ours **branched less** and sometimes **ran faster**, with geometric means down by **40%** and **20%**.

| Problem | Pool Gap | Sols | Branches One-tree | Branches $k=9$ | Runtime (s) One-tree | Runtime (s) $k=9$ |
|---|---|---|---|---|---|---|
| bm23 | 59 | 2,168 | **18,066** | **16,203** | 2.4 | 2.7 |
| lseu | 250 | 95,335 | **2,188,870** | **2,162,601** | 220.6 | 237.1 |
| mod008 | 15 | 1,228 | **2,324,558** | **2,317,052** | **188.9** | **183.4** |
| p0033 | 2112 | 10,746 | **20,475** | **7,861** | **1.7** | **1.0** |
| p0040 | 7102 | 519,216 | **851,354** | **144,066** | **74.9** | **13.3** |
| pipex | 120 | 12,266 | **48,978** | **48,842** | 5.2 | 5.9 |
| sentoy | 160 | 3,513 | **550,555** | **473,010** | **41.5** | **38.7** |
| stein15 | 6 | 2,809 | **2,011** | **830** | **0.21** | **0.19** |
| stein27 | 9 | 367,525 | **325,275** | **43,909** | 37.8 | 39.0 |
| stein45 | 1 | 19,603 | **874,245** | **870,796** | 89.3 | 96.8 |
| **Geometric mean** | | | $1.7 \times 10^5$ | $1.0 \times 10^5$ | **16.9** | **13.8** |

## 6. Next Steps

Find good choices for the $k$ bottom variables; explore heuristics to choose where to branch next when changing CPLEX default choices.

Extend this approach to the mixed-integer case.

Mix it with branching dominance, which pays off at the top of the branching tree [Fischetti and Toth, 1988, Fischetti and Salvagnin, 2010].

## 7. References

I. Abío and Peter J. Stuckey. Encoding linear constraints into SAT. In B. O'Sullivan, editor, *Proceedings of CP*, pages 75–91. Springer, 2014.

I. Abío, R. Nieuwenhuis, A. Oliveras, E. Rodríguez-Carbonell, and V. Mayer-Eichberger. A new look at BDDs for pseudo-boolean constraints. *Journal of Artificial Intelligence Research*, 45:443–480, 2012.

M. Behle. *Binary Decision Diagrams and Integer Programming*. PhD thesis, Universität des Saarlandes, 2007.

R.E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Trans. Comput.*, C-35(8):677–691, 1986.

E. Danna, M. Fenelon, Z. Gu, and R. Wunderling. Generating multiple solutions for mixed integer programming problems. In M. Fischetti and D. P. Williamson, editors, *Proceedings of IPCO*, pages 280–294. Springer, 2007.

M. Fischetti and D. Salvagnin. Pruning moves. *INFORMS Journal on Computing*, 22(1):108–119, 2010.

M. Fischetti and P. Toth. A new dominance procedure for combinatorial optimization problems. *OR Lett.*, 7(4):181–187, 1988.

IBM Corp. *IBM ILOG CPLEX Optimization Studio CPLEX Users Manual Version 12 Release 6*, 2013.