Branching Rules Revisited

Alexander Martin

Darmstadt University of Technology

MIP 2006
"Workshop on Mixed Integer Programming"

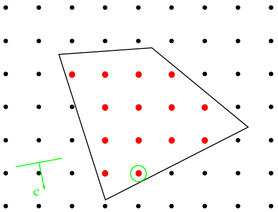
Miami
June 5 – 8, 2006

joint work with Tobias Achterberg and Thorsten Koch

Mixed Integer Program (MIP)

$$\begin{aligned} \text{min} \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b \\ & l \leq x \leq u \\ & x \in \mathbb{Z}^{n-p} \times \mathbb{R}^p \end{aligned} \qquad \text{(MIP)}$$

with $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$, $c, l, u \in \mathbb{R}^n$ and $p \in \{0, \ldots, n\}$.



LP based branch-and-bound

Input: A (MIP)
Output: An opt. solution $x^\star$ or the message "(MIP) is infeasible".

1. Initialize $S := \{P_{\mathsf{LP}}\}$, where P_{LP} is the relaxation of (MIP). Set $c^\star := \infty$.
2. If $S = \emptyset$, exit (return $x^\star$ or "(MIP) is infeasible").
3. Choose a problem $Q \in S$ and delete it from S.
4. Solve the LP $c_Q = \min\{c^T x \mid x \in Q\}$ with opt. solution $\bar{x}$ (Q is possibly strengthened by cuts).
5. If $c_Q \geq c^\star$, goto 2.
6. If $\bar{x}$ integer, set $c^\star := c_Q$ and $x^\star := \bar{x}$, and goto 2.
7. Branching: Split Q into subproblems, add them to S and goto 3.

Branching

Branching = Branching on linear inequalities

(a) Branching on trivial inequalities (= Branching on variables)

- Land & Powel (1979)
- Linderoth & Savelsbergh (1999)
- ...

(b) Branching on non-trivial inequalities

- Clochard & Naddef (1993)
- Borndörfer, Ferreira, Martin (1998)
- Naddef (2002)
- Fischetti, Lodi (2003)
- ...

Variable Selection

Input: Subproblem Q with fractional LP solution $\bar{x}$.
Output: $i \in I$ with $\bar{x}_i \notin \mathbb{Z}$.

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

Straight Away Strategies

Most Infeasible

Choose variable closest to 0.5, i. e.,

$$s_i = 0.5 - |\bar{x}_i - \lfloor \bar{x}_i \rfloor - 0.5|$$

+ seems to have the most impact on the new LPs.
+ fast to compute

Straight Away Strategies

Random

Choose variable randomly, i. e.,

$$s_i = rand()$$

The whole topic is anyway only about “reading tea leaves”!

Strategy	B&B nodes		time (sec)		fails
	total	geom.	total	geom.	
random	21 246 277	275 789.4	46 202.2	923.6	11
most infeasible	19 421 589	262 368.9	48 037.5	938.0	11

Straight Away Strategies

Random

Choose variable randomly, i. e.,

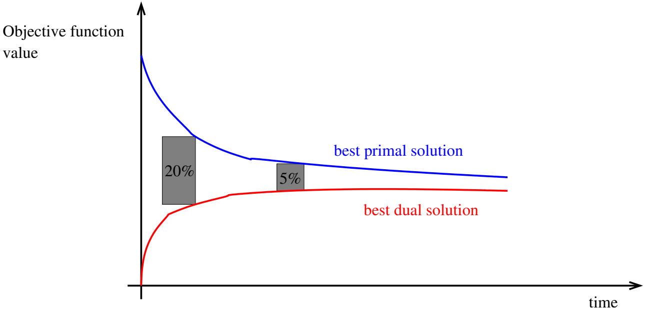
$$s_i = rand()$$

The whole topic is anyway only about "reading tea leaves"!

Strategy	B&B nodes		time (sec)		fails
	total	geom.	total	geom.	
random	21 246 277	275 789.4	46 202.2	923.6	11
most infeasible	19 421 589	262 368.9	48 037.5	938.0	11

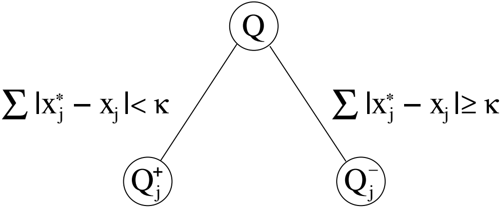
Goals of Branching

1. Improve primal bound
2. Improve dual bound



Primal Branching Rules

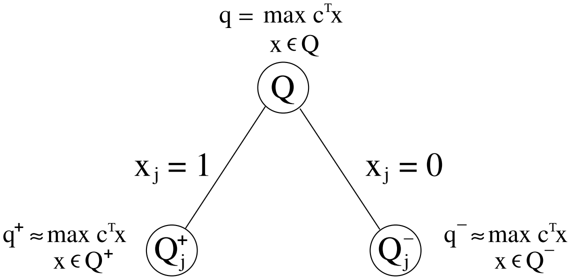
- Local Branching
  Fischetti & Lodi (2003)



- Relaxation Induced Neighborhood Search (RINS)
  Danna, Rothberg & Le Pape (2005)
- Guided Dives
  Danna, Rothberg & Le Pape (2005)

Dual Branching Rules

Measure the success in the increase of the objective function



$$s_i = \text{score}(q^-, q^+) := (1-\mu)\cdot\min\{q^-, q^+\} + \mu\cdot\max\{q^-, q^+\},$$

where μ is some scaling factor, e. g. $\mu = \frac{1}{6}$.

Strong Branching

1. Select $C' \subseteq \{i \mid \bar{x}_i \notin \mathbb{Z}\}$
2. For each $i \in C'$
   (a) Temporally set $u_i = \lfloor \bar{x}_i \rfloor$
   (b) Perform γ simplex iterations yielding obj. fct. value c_i^-
   (c) Temporally set $l_i = \lceil \bar{x}_i \rceil$
   (d) Perform γ simplex iterations yielding obj. fct. value c_i^+
   (e) Set $s_i = \text{score}(c_i^-, c_i^+)$
3. Return an index $i \in C$ with $s_i = \max_{j \in C}\{s_j\}$.

see CPLEX 7.5 and Applegate, Bixby, Chvátal, Cook (2003)

Full Strong Branching

(i) $\gamma = \infty$
(ii) $C' = \{i \mid \bar{x}_i \notin \mathbb{Z}\}$

Strong Branching

1. Select $C' \subseteq \{i \mid \bar{x}_i \notin \mathbb{Z}\}$
2. For each $i \in C'$
   (a) Temporally set $u_i = \lfloor \bar{x}_i \rfloor$
   (b) Perform γ simplex iterations yielding obj. fct. value c_i^-
   (c) Temporally set $l_i = \lceil \bar{x}_i \rceil$
   (d) Perform γ simplex iterations yielding obj. fct. value c_i^+
   (e) Set $s_i = \text{score}(c_i^-, c_i^+)$
3. Return an index $i \in C$ with $s_i = \max_{j \in C}\{s_j\}$.

see CPLEX 7.5 and Applegate, Bixby, Chvátal, Cook (2003)

Full Strong Branching

(i) $\gamma = \infty$

(ii) $C' = \{i \mid \bar{x}_i \notin \mathbb{Z}\}$

Pseudocost Branching

Keep a history of the success of the variables which has already been branched on, see Benichou et al (1971).

$$\begin{aligned}
\varsigma_i^+ &= (\bar{c}_{Q_i^+} - \bar{c}_Q) \,/\, (\lceil \bar{x}_i \rceil - \bar{x}_i) \\
\sigma_i^+ &= \sum_i \varsigma_i^+ \\
\eta_i^+ &= \text{number of these problems solved} \\
\Psi_i^+ &= \sigma_i^+ / \eta_i^+.
\end{aligned}$$

Pseudocost branching

1. Let $C = \{i \in I \mid \bar{x}_i \notin \mathbb{Z}\}$ be the set of candidates.
2. For all candidates $i \in C$, use
   $s_i = \text{score}(\,(\bar{x}_i - \lfloor \bar{x}_i \rfloor) \cdot \Psi_i^-,\ (\lceil \bar{x}_i \rceil - \bar{x}_i) \cdot \Psi_i^+)$
3. Return an index $i \in C$ with $s_i = \max_{j \in C}\{s_j\}$.

Hybrid Strong/Pseudocost Branching

Problem

Uninitialized pseudocosts $\sigma_i^+ = \eta_i^+ = 0$ at the beginning.

Solutions

(1) Initialize pseudocost values with strong branching values
Linderoth & Savelsbergh (1999)

(2) Use strong branching up to level d in the B & B tree,
use pseudocost branching from level $d+1$ on.
see, for instance, LINDO.

Hybrid Strong/Pseudocost Branching

Problem

Uninitialized pseudocosts $\sigma_i^+ = \eta_i^+ = 0$ at the beginning.

Solutions

(1) Initialize pseudocost values with strong branching values
Linderoth & Savelsbergh (1999)

(2) Use strong branching up to level d in the B & B tree,
use pseudocost branching from level $d + 1$ on.
see, for instance, LINDO.

Two simple new ideas

- Use strong branching not only on variables with uninitialized pseudocosts, but also on variables with unreliable pseudocosts.

  The pseudocosts of variable i are called *unreliable*, if

  $$\min\{\eta_i^-, \eta_i^+\} < \eta_{\text{rel}},$$

  with $\eta_{\text{rel}} \in \mathbb{N}$ being the reliability parameter.

- Select the set C of candidates dynamically.

  Introduce a so-called look ahead parameter λ.

  If the best score does not change for λ variables, then stop calling strong branching.

Reliability Branching

1. Let $C = \{i \in I \mid \bar{x}_i \notin \mathbb{Z}\}$ be the set of candidates.
2. Sort C according to non-increasing pseudocosts.

   For all $i \in C$ with $\min\{\eta_i^-, \eta_i^+\} < \eta_{\text{rel}}$, do:

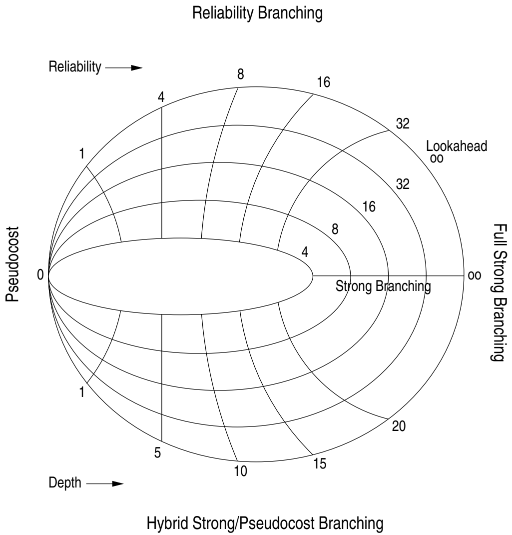
   (a) Perform γ simplex iterations on Q_i^- and Q_i^+.
   Let $\tilde{\Delta}_i^-$ and $\tilde{\Delta}_i^+$ be the objective gains.

   (b) Update the pseudocosts Ψ_i^- and Ψ_i^+ with $\tilde{\Delta}_i^-$ and $\tilde{\Delta}_i^+$.

   (c) Update the score $s_i = \text{score}(\tilde{\Delta}_i^-, \tilde{\Delta}_i^+)$.

   (d) If the maximum score $s^* = \max_{j \in C}\{s_j\}$ has not changed for λ consecutive score updates, goto 3.
3. Return an index $i \in C$ with $s_i = \max_{j \in C}\{s_j\}$.

Branching Rule Classification



Test Set

Instances are taken from

- Miplib 2003, see `http://miplib.zib.de`
- Mittelmann 2003, see `http://plato.asu.edu/bench.html`

where CPLEX 9.0 needs

- at least 5000 B& B nodes
- at most 1 hour CPU time
  (on a 833 MHz Alpha with 4 MB Cache and 2 GB RAM)

These are 24 instances:

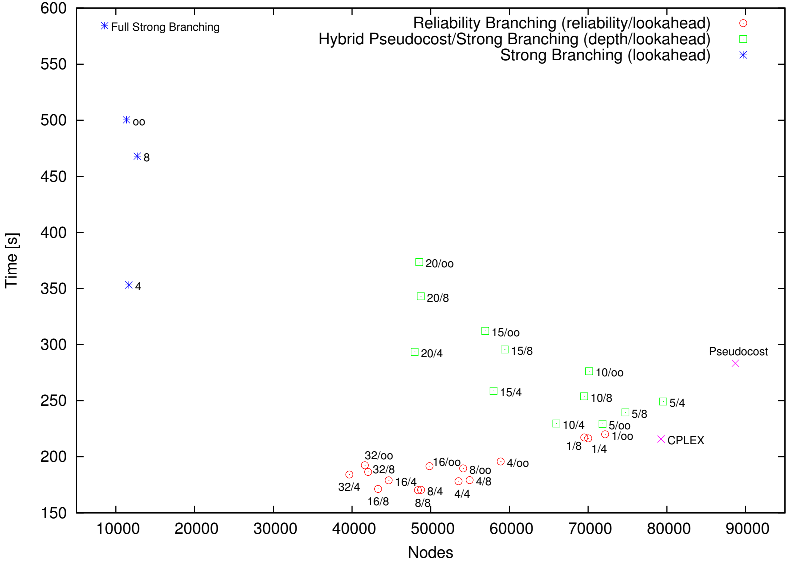
aflow30a	cap6000	gesa2-o	mas74	mas76	misc07
pp08aCUTS	qiu	rout	vpm2	ran8x32	ran10x26
ran13x13	mas284	prod1	bc1	bienst1	neos2
swath1	swath2	neos7	pk1	neos3	ran12x21

Strategy	B&B nodes		time (sec)		strong branchings		fails
	total	geom.	total	geom.	total	geom.	
random	21 246 277	275 789.4	46 202.2	923.6	0	0.0	11
most infeasible	19 421 589	262 368.9	48 037.5	938.0	0	0.0	11
pseudocost	9 381 001	88 706.8	19 945.8	283.4	0	0.0	2
full strong	929 347	7 241.7	26 397.2	504.4	15 569 652	146 784.2	2
lookahead = 4							
strong/pscost (5)	9 698 397	79 535.5	19 487.8	249.3	5 792	216.2	2
strong/pscost (10)	8 251 942	65 966.3	16 499.4	229.6	74 812	2 284.2	2
strong/pscost (15)	7 982 847	57 976.8	17 855.3	258.8	523 377	8 137.8	2
strong/pscost (20)	7 890 374	47 958.5	19 175.6	293.5	2 825 100	17 780.2	2
reliability (1)	9 000 334	70 013.6	17 199.6	216.4	39 126	374.8	1
reliability (4)	6 906 698	53 522.9	13 402.9	178.2	110 628	1 176.7	0
reliability (8)	7 937 968	48 772.8	11 132.7	170.5	117 643	1 850.3	0
reliability (16)	6 022 024	44 649.9	10 782.6	179.0	187 578	3 640.6	0
reliability (32)	7 940 797	39 655.2	11 103.0	184.2	253 014	5 837.8	0
strong branching	1 325 589	11 639.5	20 427.2	353.2	9 965 454	86 188.6	3

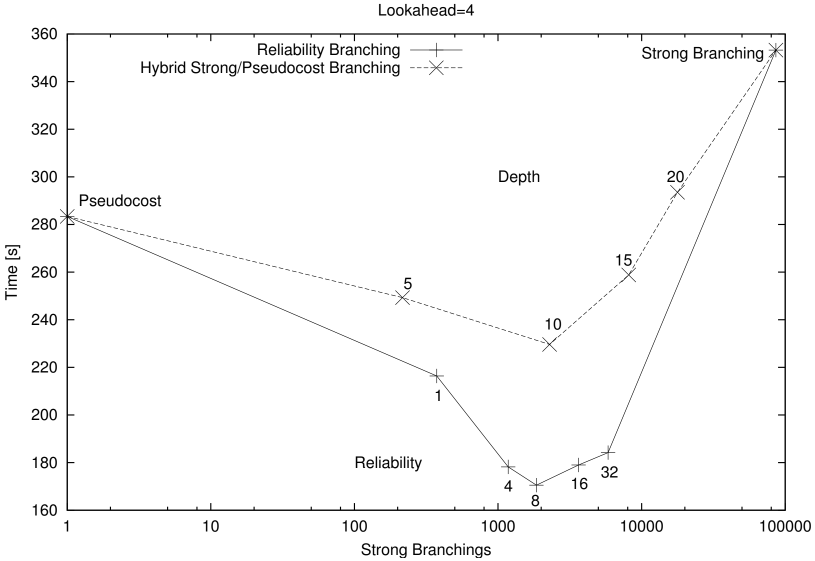
Strategy	B&B nodes		time (sec)		strong branchings		fails
	total	geom.	total	geom.	total	geom.	
lookahead = 8							
strong/pscost (5)	8 653 318	74 730.8	17 389.5	239.6	7 397	268.6	1
strong/pscost (10)	8 332 543	69 489.0	18 362.5	253.9	102 082	2 523.8	2
strong/pscost (15)	7 456 685	59 398.8	20 479.8	295.7	750 577	9 983.1	2
strong/pscost (20)	7 551 419	48 736.5	22 388.8	343.1	3 695 577	20 837.9	3
reliability (1)	8 663 537	69 501.0	16 753.8	217.2	53 557	429.0	1
reliability (4)	8 338 386	54 937.7	12 497.2	179.2	74 906	1 104.6	0
reliability (8)	7 813 409	48 377.3	12 380.2	170.2	133 545	1 998.0	0
reliability (16)	7 579 400	43 311.9	11 946.7	171.3	185 136	3 589.7	0
reliability (32)	7 207 836	42 047.5	11 835.7	186.5	259 482	5 913.2	0
strong branching	1 294 569	12 714.7	25 619.4	468.0	12 651 504	119 799.1	4

Strategy	B&B nodes total	B&B nodes geom.	time (sec) total	time (sec) geom.	strong branchings total	strong branchings geom.	fails
lookahead = ∞							
strong/pscost (5)	8 498 292	71 817.4	18 116.9	229.3	14 675	489.9	2
strong/pscost (10)	9 247 636	70 125.8	20 472.1	276.2	154 458	3 870.1	2
strong/pscost (15)	6 670 440	56 926.6	19 907.4	312.2	890 187	13 127.2	3
strong/pscost (20)	7 627 640	48 547.0	23 538.5	373.6	3 842 516	26 557.4	3
reliability (1)	7 747 290	72 159.1	15 825.8	220.0	48 162	408.4	1
reliability (4)	9 068 723	58 886.4	14 258.1	195.8	78 625	1 096.6	2
reliability (8)	8 551 045	54 118.3	13 563.0	189.6	135 541	2 042.9	1
reliability (16)	6 567 432	49 839.9	12 766.4	191.6	196 220	3 601.0	0
reliability (32)	7 502 942	41 636.1	12 393.6	192.5	281 822	6 000.7	0
strong branching	1 163 822	11 355.7	26 176.4	500.3	12 793 364	127 737.1	4
CPLEX/SIP cuts	10 467 429	79 269.0	18 617.4	215.8	—	—	1

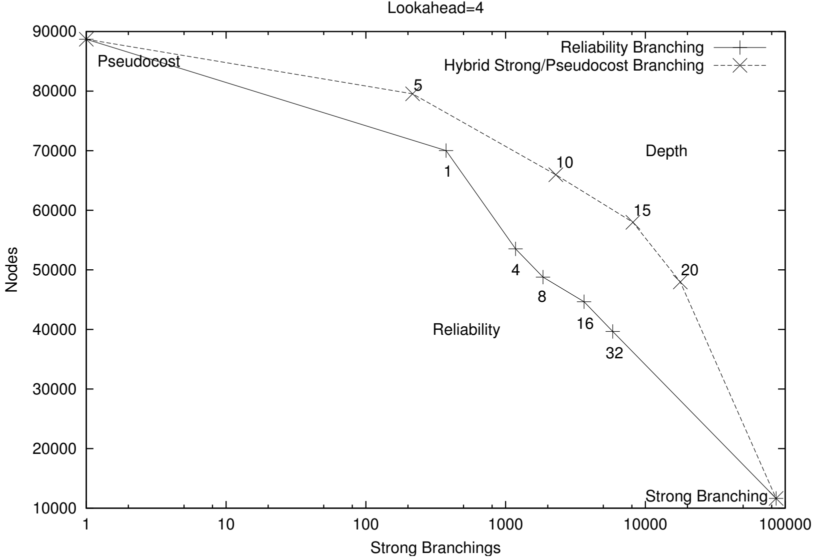
Nodes versus Time



Time versus Strong Branchings



Nodes versus Strong Branchings



Conclusions

Summary

- most infeasible as good as random
- *strong branching* is best with respect to number of nodes, but not with respect to time
- *reliability branching* outperforms *hybrid strong/pseudocost branching*
- Increasing η_{rel} (or the depth d) decreases the number of nodes
- Currently best choice $\eta_{\text{rel}} = 8$ and $\lambda = 4$.

Open

- Bridge the gap to *full strong branching* without increasing the running time.
- Missing Theory !?
- Branching versus modeling with binary variables.