Choosing the best cuts

William Cook   Ricardo Fukasawa   Marcos Goycoolea

MIP2006 - June 5th, 2006

Introduction:

- The problem:
  Given:
  - $P_I = \min\{cx : Ax = b; Cx \leq d; l \leq x \leq u; x_i \in \mathbb{Z} \text{ for } i \in I\}$
  - x^*: an optimal solution to the LP relaxation of P_I
  - A set $\Pi = \{\pi_1, \ldots, \pi_K\}$ cuts valid for P_I and violated by x^*.

  Choose a small "good" subset of cuts to add to the LP relaxation.

- Objective of this work:
  - Formalize what a "good" subset of cuts means and what are the issues involved.
  - Extend the rules used in practice.
  - Quantify how good/bad are the rules.

Literature review:

- Padberg and Rinaldi (1991) give a motivation for the problem:
  `"In our estimation, finding a reasonable quality measure (for a cut) is one of the central issues in the area of polyhedral cutting-plane algorithms that is - as of today - not yet investigated satisfactorily."`
- Padberg and Rinaldi (1991): Cuts for the TSP should be evaluated in the affine subspace of degree constraints.
- Juenger, Reinelt and Thienel (1994) mention the problem but do not attempt to study it.
- Balas, Ceria and Cornuejols (1996):
  - The objective function is not necessarily the “ideal” that we should aim for when evaluating a cut:
    Zero gap problems or infeasible problems.
  - Steepness is good for full dimensional polytopes, but are also a reliable guide even when not full-dimensional.
  - Scaling problem with violation can be resolved by normalizing.
- Andreello, Caprara and Fischetti (2003):
  - Variations of steepness (unspecified) were unsuccesful.
  - Consider angles to make cuts more diverse

Issues

- Need a criteria for evaluating cut selection rules
  - Which set of cuts improves most the objective function?
  - Which set of cuts reduces most the solution time?

  We choose the first since it is less tied to a specific solver and parameter setting.
  We evaluate cuts only in a first round of cut addition to avoid the "curse of the tableau".
- Need to estimate how the set of cuts will perform after being added.
  Choices:
  - Compute an estimate $V_1(\pi_i)$ for each i. Choose k cuts with best $V_1(\pi_i)$
  - Choose the set $\{\pi_1, \ldots, \pi_k\}$ that maximizes an estimate $V_k(\pi_1, \ldots, \pi_k)$

  We focus on the first for simplicity.

How can we estimate performance?

Let:

$$P_i := \text{ current LP relaxation}$$

$$P_{i+1} = P_i \cap H_\pi = P_i \cap \{x : \pi x \leq \pi_o\}$$

Let x_i^* be the optimal solution to $min\{cx : x \in P_i\}$.

- **Idea:** Compute or approximate:
  - $vol(P_i \setminus P_{i+1})$
  - $|cx_{i+1}^* - cx_i^*|$
  - $dist(x_i^*, P_{i+1})$
- **Common rules**:
  - Violation: $V_1(\pi) = \pi x^* - \pi_o$
    NOT INVARIANT UNDER SCALING
  - Steepness (distance from x^* to $H_\pi = \{x : \pi x \leq \pi_o\}$):

$$V_1(\pi) = \frac{\pi x^* - \pi_o}{||\pi||}$$

NOT INVARIANT UNDER ADDITION OF EQUALITIES

Rotated steepness

If we wish to take into account a system of equalities (Padberg and Rinaldi (1991)), we would need to calculate:

$$dist(x^*, H_\pi \cap \{x : Ax = b\})$$

For this, "rotate" π, obtaining $\hat{\pi}$ such that:

$$A\hat{\pi} = 0 \text{ and } \hat{\pi} = (\pi + A^T\lambda)$$

Note that:

$$A(\pi + A^T\lambda) = 0 \Rightarrow \lambda = -(AA^T)^{-1}A\pi$$

Measuring the steepess of $\hat{\pi}$ we obtain:

$$\frac{\pi x^* - \pi_o}{||\pi - A^T(AA^T)^{-1}A\pi||}$$

TSP case:

- Matrix A defined by degree constraints

$$(AA^t)^{-1} = \begin{pmatrix} \alpha(n) & & & \beta(n) \\ & \alpha(n) & & \\ & & \ddots & \\ & \beta(n) & & \alpha(n) \end{pmatrix}$$

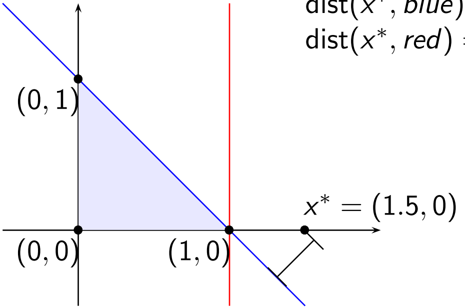
- Then, for $e, f \in E$, let $k = |e \cap f|$. We have:

$$(A^t(AA^t)^{-1}A)_{ef} = k\alpha(n) + (4-k)\beta(n)$$

- The main point: Fast and easy to compute rotated steepness
- Probably also happens in other problems with known structure

Other alternative to steepness

Steepness $= dist(x^*, H_\pi)$
Alternative $= dist(x^*, H_\pi \cap \{x : l \leq x \leq u\})$



$\text{dist}(x^*, \textit{blue}) = 1/(2\sqrt{2}) < 0.5$
$\text{dist}(x^*, \textit{red}) = 0.5$

Steepness with bounds: A simple Algorithm

Define $P = \{x \in \mathbb{R}^n : \pi x \leq \pi_o\}$:

1. Let $\hat{x}$ be the minimizer of $||x - x^*||$ over P.
2. If $l \leq \hat{x} \leq u$, then **STOP**.
3. If $\hat{x}_i > u_i$, then $P \leftarrow P \cap \{x_i = u_i\}$.
4. If $\hat{x}_i < l_i$, then $P \leftarrow P \cap \{x_i = l_i\}$.
5. **GOTO** step 1.

- Provably correct
- Worst case n iterations (there exists an example that achieves it)
- In practice we didn't observe more than 1 iteration

What is the best we can hope for?

- If we are looking at the distance from x_i^* to P_{i+1}, the benchmark we should test against is:

$$V_1(\pi) = dist(x_i^*, P_{i+1} = P_i \cap H_\pi)$$

which we computed using CPLEX.

- If we are looking at costs, the best we can hope for when looking at costs for single rows is:

$$V_1(\pi) = \min\{cx : x \in P_{i+1}\}$$

- We can approximate this by only allowing few pivots (for our tests, we used 10 pivots).

Results - Benchmarked against adding 100% of the cuts

Average over TSPLIB problems. Cuts used were MOD-2 cuts.

Cuts (%)	Cut selection rule (results in % of total possible improvement)						
	viol	steep	steepwb	rsteep	distpoly	primal	primalpivot
10	7.29	37.28	37.02	37.77	36.03	67.39	67.22
20	11.61	60.77	60.94	61.75	60.18	80.4	80.28
30	18.75	75.89	75.49	76.42	75.86	87.07	86.72
40	24.87	82.97	83.1	83.62	83.81	90.73	90.51
50	29.95	88.36	88.37	88.83	93.19	92.98	92.83
60	39.41	92.62	92.76	92.82	95.85	95.69	95.57
70	53.05	95.32	95.41	95.21	97.76	97.44	97.5
80	67.25	97.77	97.83	97.65	98.63	98.96	98.95
90	84.55	99.01	98.98	99.01	99.28	99.4	99.39

Average over MIPLIB problems. Cuts used were t-MIR cuts. (*) is incomplete

Cuts (%)	Cut selection rule (results in % of total possible improvement)						
	viol	steep	steepwb	rsteep(*)	distpoly	primal	primalpivot
10	23.3	58.44	59.3	51.5	64.67	80.78	80.53
20	37.26	77.2	76.08	77.73	78.45	89.2	88.1
30	50.64	88.52	88.69	88.49	85.87	93.72	91.14
40	58.04	91.97	91.76	91.85	90.07	96.43	94.15
50	73.09	93.87	93.17	94.75	92.2	97.18	95.65
60	82.52	95.01	95.49	96.91	94.43	97.85	97.73
70	88.47	97.54	96.0	98.91	97.35	98.82	98.72
80	93.19	99.36	99.52	99.4	99.78	99.44	99.5
90	99.56	99.73	99.95	100.0	99.95	99.84	99.79

Note: On average 8.4% of cuts are "good" (i.e. have nonzero dual variables when adding all cuts). Any rule above needs at least 87% of the cuts to cover all the "good" cuts.

Final Remarks:

What we did:

- Review of the problem through different perspectives
- What important points should we be aware of?
- Extensions of common rules

Conclusions:

- The rules based on distance seem to have a similar performance.
- Rules based on cost do better.

Next questions:

- Study rules for evaluating sets of cuts
- Do tests for other classes of cuts besides MIR and Mod-2?
- Effectiveness of rules to evaluate cuts from one class against cuts from a different one (e.g.: Combs vs. MIR's)?
- Try to efficiently approximate cost improvement.
- Consider volume as a performance estimate