Experiments On General Disjunctions

Some Dumb Ideas We Tried That Didn't Work* and Others We Haven't Tried Yet

*But that may provide some insight

Ted Ralphs, Serdar Yildiz

COR@L Lab, Department of Industrial and Systems Engineering, Lehigh University

INFORMS, October 15 2012

COR@L
COMPUTATIONAL OPTIMIZATION RESEARCH AT LEHIGH

Outline

Quick Review

Mixed Integer Linear Program (MILP)

$$\begin{aligned} z_{IP} &= \min cx \\ s.t.\ \ Ax &\geq b \qquad \text{(MILP)} \\ x &\in \mathbb{Z}^d \times \mathbb{R}^{n-d}, \end{aligned}$$

where $A \in \mathbb{Q}^{m \times n}$, $b \in \mathbb{Q}^m$, $c \in \mathbb{Q}^n$, $m, n, d \in \mathbb{N}$ are given.

Linear Programming (LP) Relaxation

$$\begin{aligned} z_{LP} &= \min cx \\ s.t.\ \ Ax &\geq b \qquad \text{(LP)} \\ x &\in \mathbb{R}^n, \end{aligned}$$

where $A \in \mathbb{Q}^{m \times n}$, $b \in \mathbb{Q}^m$, $c \in \mathbb{Q}^n$, $m, n \in \mathbb{N}$ are given.

1. $z_{LP} \leq z_{IP}$ provides a lower bound (z_l) on z_{IP}.
2. Any $\hat{x} \in Z^d \times \mathbb{R}^{n-d}$ s.t. $A\hat{x} \geq b$ provides an upper bound (z_u) on z_{IP}.
3. "Tighten" the feasible region of the (LP) relaxation iteratively.
4. Repeat until $z_l = z_u$.

Disjunctions

- A *binary disjunction* is defined by (π, π_0, π_1) for $\pi \in \mathbb{R}^n, \pi_0, \pi_1 \in \mathbb{R}$.
- Such a disjunction is said to be *valid* if

$$\pi x \leq \pi_0 \vee \pi x \geq \pi_1 \quad \forall x \in \mathcal{P}_I, \tag{1}$$

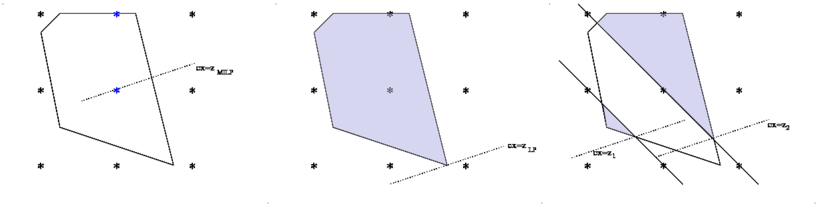
- If $\pi \in \mathbb{Z}^n$, $\pi_0 \in \mathbb{Z}$, $\pi_1 = \pi_o + 1, \pi_i = 0, i > d$, then (π, π_0, π_1) is valid for (MILP) and is called a *general dijunction*.
- *Disjunctive subsets* associated with (π, π_0, π_1) are

$$\mathcal{P}_0 = \{x \in \mathbb{R}^n \mid Ax \geq b, \pi x \leq \pi_0\} \tag{2}$$

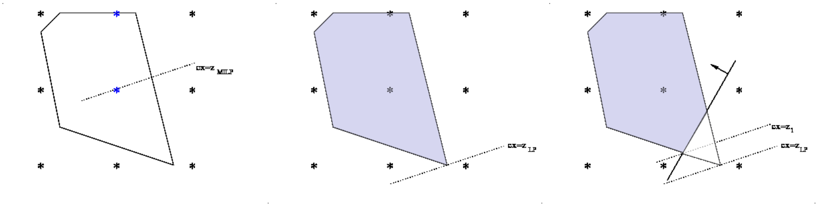
$$\mathcal{P}_1 = \{x \in \mathbb{R}^n \mid Ax \geq b, \pi x \geq \pi_1\} \tag{3}$$

- A *valid inequality* is a valid disjunction for which $\mathcal{P}_1 = \emptyset$.
- A special class that will be of interest is the *variable disjunctions* for which π is a unit vector.

Two Algorithms



Branch and Bound Algorithm



Cutting Plane Algorithm

Selecting a Disjunction

- Our goal is to iteratively obtain upper and lower bounds that are equal.
- It would seem natural that we want relaxations that yield good lower bounds.
- Thus, it would seem that disjunction selection should be driven by bound improvement.
- Suppose, however, that we have an optimal solution x^*.
- The goal is then to prove that $\mathcal{P}_I \cap \{x \in \mathbb{R}^n \mid cx < cx^*\} = \emptyset$.
- What does the bound have to do with it?

Exploiting a Disjunction: Branching

Branching on (π, π_0, π_1) creates two *subproblems* with feasible regions.

$$\mathcal{P}_0 = \min\{x \in \mathbb{R}^n, Ax \geq b, \pi x \leq \pi_0\}, \tag{4}$$

$$\mathcal{P}_1 = \min\{x \in \mathbb{R}^n, Ax \geq b, \pi x \geq \pi_1\} \tag{5}$$

Solving the LP relaxations of each yields

$$z_0 = \min\{cx \mid x \in \mathbb{R}^n, Ax \geq b, \pi x \leq \pi_0\}, \tag{6}$$

$$z_1 = \min\{cx \mid x \in \mathbb{R}^n, Ax \geq b, \pi x \geq \pi_1\} \tag{7}$$

and we obtain the lower bound

$$z_B = min(z_0, z_1) \tag{8}$$

Exploiting a Disjunction: Cutting

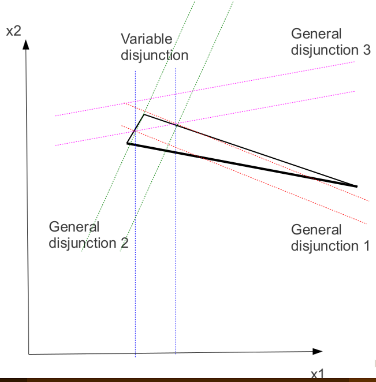
- A *disjunctive cut* associated with (π, π_0, π_1) is an inequality (α, β) valid for $\mathcal{P}_0 \cup \mathcal{P}_1$.
- Note that a disjunctive cut is an inequality valid for (MILP).
- If we solve the augmented LP relaxation

$$z_C = \min\{cx \mid x \in \mathbb{R}^n, Ax \geq b, \alpha x \leq \beta\}, \tag{9}$$

  how does the bound compare?
- We have $z_c \leq z_B$ with equality if $\alpha = c$ and $\beta = z_B$.

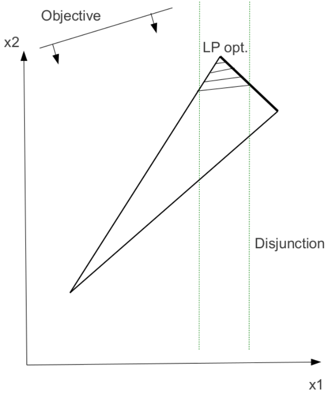
Measuring Effectiveness: Branching

- In selecting branching disjunctions, we usually consider bound improvement as our primary measure.
- In general, there are an infinite number of disjunctions that achieve the same bound.
- What is the difference between these disjunctions?



Measuring Effectiveness: Cutting

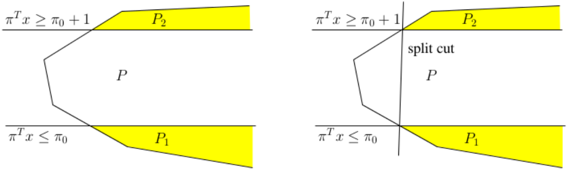
- In selecting disjunctions for cutting, we usually consider a measure based on *volume*.
- This is seen as a proxy for bound improvement, but might it be the other way around?



Branching Versus Cutting

What is the tradeoff of branching versus cutting?

1. Cutting does not create any additional subproblems.
2. Cutting degrades numerics.
3. The bound from cutting can be weaker.



Nannicini et al (2011)

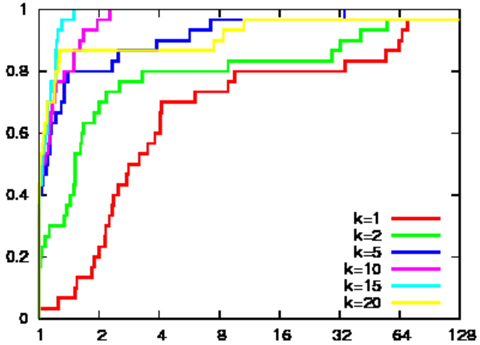
Outline

Previous Literature On Selection Criteria

- Cutting
  - Balas and Saxena (2007) for split inequalities,
  - Fischetti and Lodi (2005) for C-G inequalities,
  - Balas et al. (1996a) for lift-and-project inequalities,
  - Gu et al. (1998) for lifted-cover inequalities.
  - Mahajan (2009) (maximum bound improvement)
- Branching
  - Bound improvement
    - Mahajan (2009).
    - Nannicini et al (2011).
    - Karamanov and Cornuéjols (2011).
    - Cornuéjols et al. (2008).
    - Nannicini et al (2011).
  - Thin directions, Gao and Zhang (2002).

Bound Improvement

- See (Mahajan and R., 2009a, 2009b) for details.
- Formulated a MIP to determine if the bound can be improved by K.
- Solved a sequence of MIPs with varying K.
- Add additional constraints (like # non-zeros in disjunction $\leq k$).



Performance Profile of number of nodes in branch-and-bound tree.

Violation

- Cuts that remove more “volume” result in a “tighter” formulation.
- Rather than directly measuring volume, we often measure a cut’s degree of violation.
- Violation is also a proxy for bound improvement.
- Can we use degree of violation to select general disjunctions?
- Spoiler: yes, but you shouldn’t!

Selecting Disjunctions by Violation

- We considered two different violation measures for a set of MIPLIB 3.0 problems.
  - $min(\hat{\pi}\hat{x} - \hat{\pi}_0, \hat{\pi}_0 + 1 - \hat{\pi}\hat{x})$
  - $max(\hat{\pi}x - \hat{\pi}_0, \hat{\pi}_0 + 1 - \hat{\pi}x)$
- We compare our results with strong branching and the method of Karamanov and Cornuéjols (2011).

Selecting Disjunctions by Cut Violation

- Our experiments are compared with the method of Karamanov and Cornuéjols (2011).
- A split disjunction that generates a deep cut may also be useful for branching.
- Procedure
  - Obtain MIG cuts from LP opt. tableau.
  - For each MIG, obtain split disjunctions that MIG can be generated from.
  - For each split disjunction, calculate the coefficients of intersection cut.
  - Calculate the depth of the intersection cut.
  - Select disjunctions corresponding to the deepest k intersection cuts.

Relative gap closed at root node

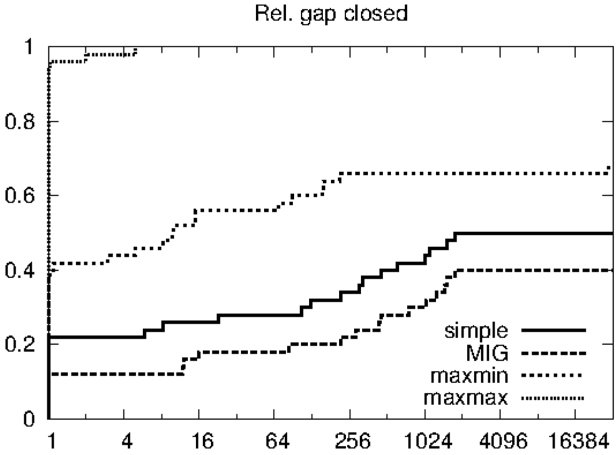


Figure: Relative gap closed at root node

Relative gap closed After 2000 nodes

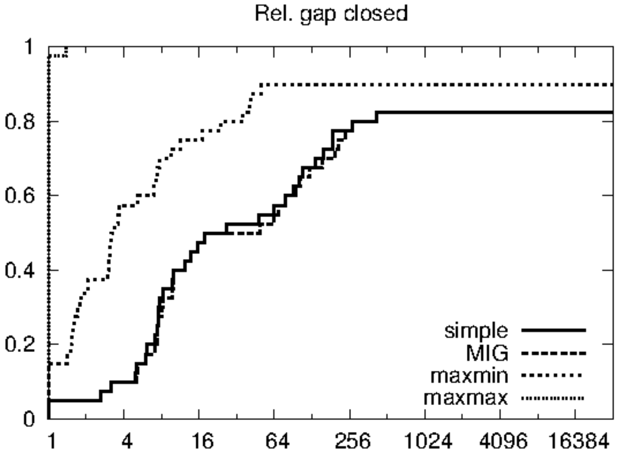


Figure: Rel gap closed with 2000 node limit

Efficient general disjunction

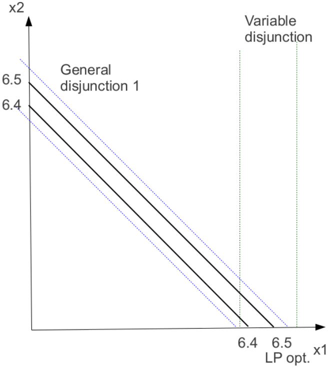


Figure: Efficient general disjunction

- $x_1 \leq 6, x_1 \geq 7$, variable disjunction
- $x_1 + x_2 \leq 6, x_1 + x_2 \geq 7$, general disjunction.

Cut off polyhedron

- Our experience with maximum violating disjunction motivated us to experiment with maximumizing cut off volume.
- *Cut off polyhedron* is the polyhedron cut off by disjunction which is

$$\{x \in \mathbb{R}^n | Ax \leq b, \hat{\pi}x \geq \hat{\pi}_0, \hat{\pi}x \leq \hat{\pi}_0 + 1\}$$

- Calculating volume of polyhedra is $\sharp P$-hard!
- There is no open source or commercial code to calculate volume of polyhedron except special polyhedra such as
  - hyper-rectange
  - hyper-cube etc.
- How to approximate the volume?

Cut off polyhedron

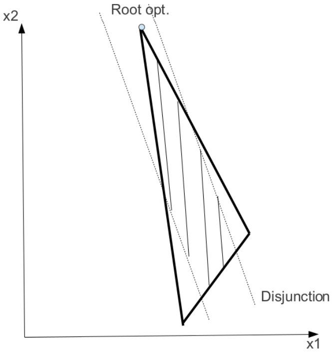


Figure: Cut off polyhedron

Outline

Maximum inscribed ellipsoid

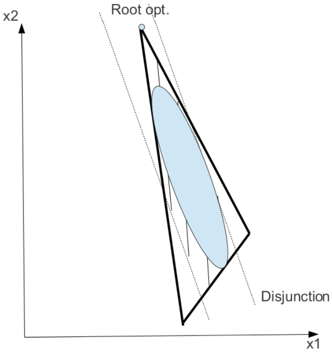


Figure: Ellipsoid

- Lower bound for the volume of cut off polyhedron.

Minimum outscribed hyper-rectangle

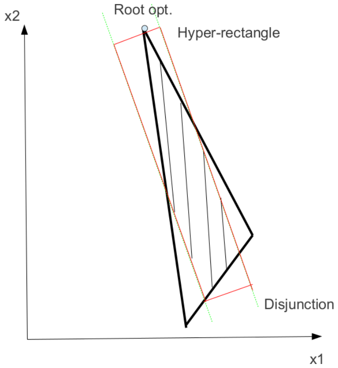


Figure: Minimum outscribed hyper-rectangle

Maximum distance in cut off polyhedron

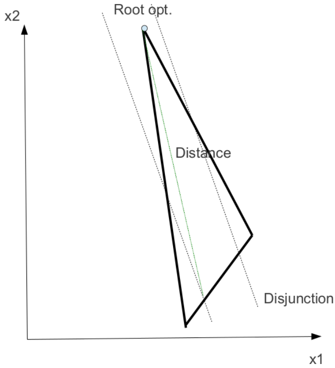


Figure: Maximum distance in cut off polyhedron

Maximum width direction

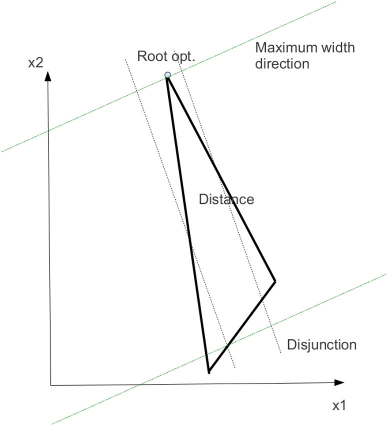


Figure: Maximum width direction

Maximum distance to LP optimum solution

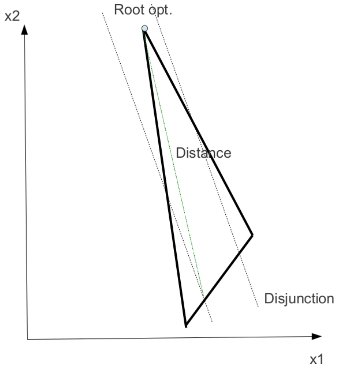


Figure: Maximum distance to LP optimum solution

Outline

Conclusion

- Bound improvement efficient but expensive.
- There are cases bound improvement may fail.
- Violation does not work well in practice.
- Next criteria to be considered is cut-off polyhedron volume.

A. Mahajan. On selecting disjunctions for solving mixed integer programming problems, 2009.

M. Karamanov and G. Cornuéjols. Branching on general disjunctions. *Math. Program. Ser. A.*, 128: 403–436, 2011.

G. Nannicini, G. Cornuéjols, M. Karamanov, L. Liberti. Branching on split disjunctions. *Combinatorial Optimization: Methods and Applications*, V. Chvatal ed., IOS Press:164–182, 2011.

E. Balas and A. Saxena. Optimizing over the split closure. *Mathematical Programming, Series A*, 2007. To appear.

M. Fischetti and A. Lodi. *Integer Programming and Combinatorial Optimization*, chapter Optimizing over the First Chvátal Closure, pages 12–22. Springer Berlin / Heidelberg, 2005.

E. Balas, S. Ceria, and G. Cornuéjols. Mixed 0-1 programming by lift-and-project in a branch-and-cut framework. *Management Science*, 42 (9):1229–1246, 1996.

Z. Gu, G. Nemhauser, and M. Savelsbergh. Lifted cover inequalities for 0-1 integer programs: Computation. *INFORMS Journal on Computing*, 10: 427–437, 1998.

L. Gao and Y.Zhang. Computational Experience With Lenstra's Algorithm. Tecnical Report, Available at http://www.caam.rice.edu/caam/trs/2002_abstracts.html#TR02-12, 2002.

G. Cornuéjols, L. Liberti, and G. Nannicini. Improved strategies for branching on general disjunctions. Mathematical Programming manuscript, Working Paper, 2008.

Questions?