Separation, Inverse Optimization, and Decomposition: Some Observations

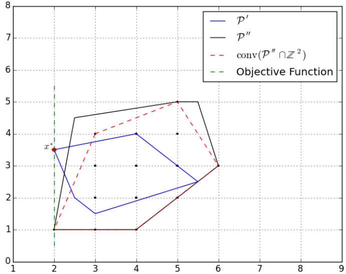
Ted Ralphs[1]
Joint work with:
Aykut Bulut[1]

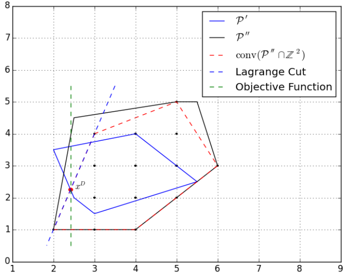
[1]COR@L Lab, Department of Industrial and Systems Engineering, Lehigh University

COLGEN 2016, Buzios, Brazil, 25 May 2016

What Is This Talk About?

- Duality in integer programming.
- Connecting some concepts.
    - Decomposition methods
    - Inverse optimization
    - Separation problem
    - Primal cutting plane algorithms for MILP
- A review of some "well-known"(?) classic results.
- Googledipity!





Setting

- We focus on the case of the mixed integer linear optimization problem (MILP), but many of the concepts are more general.

$$z_{IP} = \min_{x \in \mathcal{S}} c^\top x, \tag{MILP}$$

where, $c \in \mathbb{R}^n$, $\mathcal{S} = \{x \in \mathbb{Z}^r \times \mathbb{R}^{n-r} \mid Ax \leq b\}$ with $A \in \mathbb{Q}^{m \times n}$, $b \in \mathbb{Q}^m$.

- For most of the talk, we consider the case $r = n$ and $\mathcal{P}$ bounded for simplicity.

Googledipity*

Googling for something and finding something else that turns out to be incredibly useful.

*I thought I had invented this term until I Googled it!

Duality in Mathematical Optimization

- It is difficult to define precisely what is meant by "duality" in general mathematics, though the literature is replete with various "dualities."
  - Set Theory and Logic (De Morgan Laws)
  - Geometry (Pascal's Theorem & Brianchon's Theorem)
  - Combinatorics (Graph Coloring)
- In optimization, duality is *the* central concept from which much theory and computational practice emerges.

Forms of Duality in Optimization

- NP versus co-NP (computational complexity)
- Separation versus optimization (polarity)
- Inverse optimization versus forward optimization
- Weyl-Minkowski duality (representation theorem)
- Economic duality (pricing and sensitivity)
- Primal/dual functions/problems

What is Duality Used For?

- One way of viewing duality is as a tool for *transformation*.
  - Primal $\Rightarrow$ Dual
  - H-representation $\Rightarrow$ V-representation
  - Membership $\Rightarrow$ Separation
  - Upper bound $\Rightarrow$ Lower bound
  - Primal solutions $\Rightarrow$ Valid inequalities
- Optimization methodologies exploit these dualities in various ways.
  - Solution methods based on primal/dual bounding
  - Generation of valid inequalities
  - Inverse optimization
  - Sensitivity analysis, pricing, warm-starting

Duality in Integer Programming

- The following generalized *dual* can be associated with the base instance (MILP) (see [Güzelsoy and R(2007)])

$$\max \ \{F(b) \mid F(\beta) \leq \phi_D(\beta), \ \beta \in \mathbb{R}^m, F \in \Upsilon^m\} \qquad \text{(D)}$$

where $\Upsilon^m \subseteq \{f \mid f : \mathbb{R}^m \rightarrow \mathbb{R}\}$ and ϕ_D is the *(dual) value function* associated with the base instance (MILP), defined as

$$\phi_D(\beta) = \min_{x \in \mathcal{S}(\beta)} c^\top x \qquad \text{(DVF)}$$

for $\beta \in \mathbb{R}^m$, where $\mathcal{S}(\beta) = \{x \in \mathbb{Z}^r \times \mathbb{R}^{n-r} \mid Ax \leq \beta\}$.

- We call F^* *strong* for this instance if F^* is a *feasible* dual function and $F^*(b) = \phi_D(b)$.

The Membership Problem

Membership Problem

Given $x^* \in \mathbb{R}^n$ and polyhedron $\mathcal{P}$, determine whether $x^* \in \mathcal{P}$.

For $\mathcal{P} = \text{conv}(\mathcal{S})$, the membership problem can be formulated as the following LP.

$$\min_{\lambda \in \mathbb{R}^{\mathcal{E}}_{+}} \left\{ 0^\top \lambda \;\middle|\; E\lambda = x^*, 1^\top \lambda = 1 \right\} \tag{MEM}$$

where $\mathcal{E}$ is the set of extreme points of $\mathcal{P}$ and E is a matrix whose columns are in correspondence with the members of $\mathcal{E}$.

- When (MEM) is feasible, then we have a proof that $x^* \in \mathcal{P}$.
- When (MEM) is infeasible, we obtain a separating hyperplane.
- In fact, the dual of (MEM) is a variant of the separation problem.

The Separation Problem

Separation Problem

Given a polyhedron $\mathcal{P}$ and $x^* \in \mathbb{R}^n$, either certify $x^* \in \mathcal{P}$ or determine (π, π_0), a valid inequality for $\mathcal{P}$, such that $\pi x^* > \pi_0$.

For $\mathcal{P}$, the separation problem can be formulated as the dual of (MEM).

$$\max \left\{ \pi x^* - \pi_0 \;\middle|\; \pi^\top x \leq \pi_0 \; \forall x \in \mathcal{E}, (\pi, \pi_0) \in \mathbb{R}^{n+1} \right\} \qquad \text{(SEP)}$$

where $\mathcal{E}$ is the set of extreme points of $\mathcal{P}$.

- Note that we need some appropriate normalization.
- Assuming 0 is in the interior of $\mathcal{P}$, we can take $\pi_0 = 1$.
- In this case, we are optimizing over the *1-polar* of $\mathcal{P}$.
- This is equivalent to changing the objective of (MEM) to $\min 1^\top \lambda$.

The 1-Polar

Assuming 0 is in the interior of $\mathcal{P}$, the set of all inequalities valid for $\mathcal{P}$ is

$$\mathcal{P}^* = \left\{ \pi \in \mathbb{R}^n \;\middle|\; \pi^\top x \leq 1 \;\forall x \in \mathcal{P} \right\} \tag{1}$$

and is called its *1-polar*.

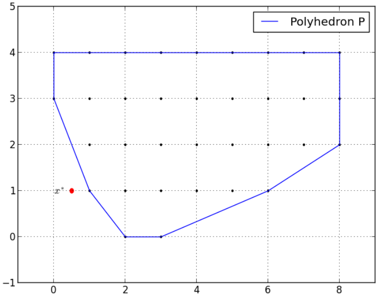
Properties of the 1-Polar

- $\mathcal{P}^*$ is a polyhedron;
- $\mathcal{P}^{**} = \mathcal{P}$;
- $x \in \mathcal{P}$ if and only if $\pi^\top x \leq 1 \;\forall \pi \in \mathcal{P}^*$;
- If $\mathcal{E}$ and $\mathcal{R}$ are the extreme points and extreme rays of $\mathcal{P}$, respectively, then
$$\mathcal{P}^* = \left\{ \pi \in \mathbb{R}^n \;\middle|\; \pi^\top x \leq 1 \;\forall x \in \mathcal{E}, \pi^\top r \leq 0 \;\forall r \in \mathcal{R} \right\}.$$

- A converse of the last result also holds.
- Separation can be interpreted as optimization over the polar.

Separation Using an Optimization Oracle

- We can solve (SEP) using a cutting plane algorithm that separates intermediate solutions from the 1-polar.
- The separation problem for the 1-polar of $\mathcal{P}$ is precisely a linear optimization problem over $\mathcal{P}$.
- We can visualize this in the dual space as column generation wrt (MEM).
- Example



Separation Example: Iteration 1

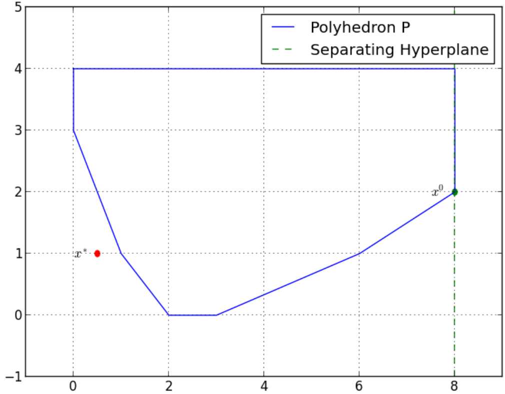


Figure: Separating x^* from $\mathcal{P}$ (Iteration 1)

Separation Example: Iteration 2

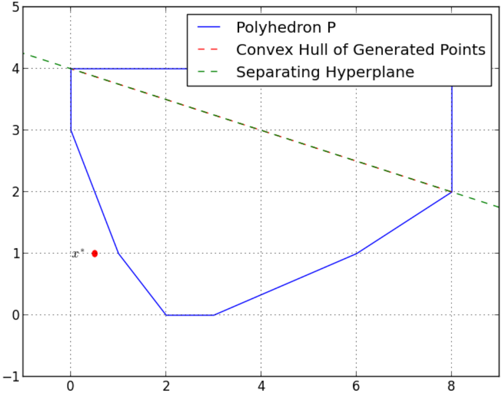


Figure: Separating x^* from $\mathcal{P}$ (Iteration 2)

Separation Example: Iteration 3

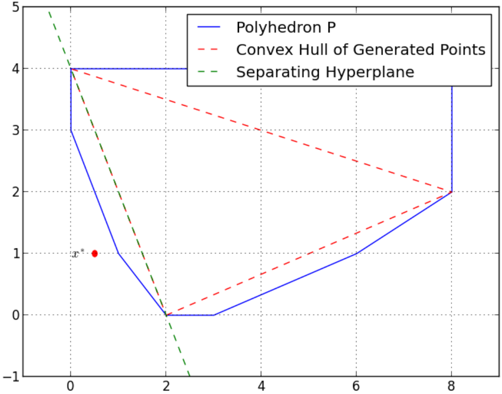


Figure: Separating x^* from $\mathcal{P}$ (Iteration 3)

Separation Example: Iteration 4

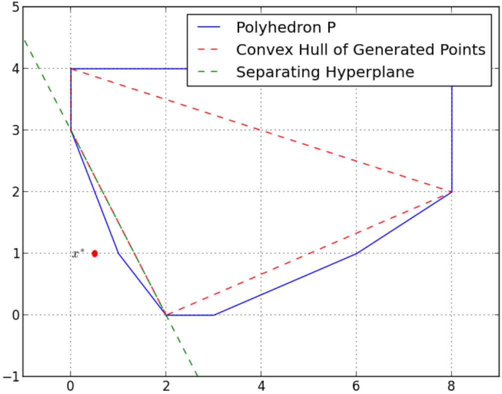


Figure: Separating x^* from $\mathcal{P}$ (Iteration 4)

Separation Example: Iteration 5

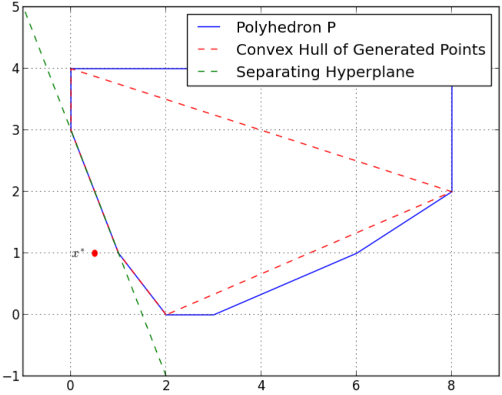


Figure: Separating x^* from $\mathcal{P}$ (Iteration 5)

Inverse Problems

What is an inverse problem?

Given a function, an inverse problem is that of determining *input* that would produce a given *output*.

- The input may be partially specified.
- We may want an answer as close as possible to a given *target*.

- This is precisely the mathematical notion of the inverse of a function.
- A *value function* is a function whose value is the optimal solution of an optimization problem defined by the given input.
- The inverse problem with respect to an optimization problem is to evaluate the inverse of a given *value function*.

Why is Inverse Optimization Useful?

Inverse optimization is useful when we can observe the result of solving an optimization problem and we want to know what the input was.

Example: Consumer preferences

- Let's assume consumers are rational and are making decisions by solving an underlying optimization problem.
- By observing their choices, we try ascertain their utility function.

Example: Analyzing seismic waves

- We know that the path of seismic waves travels along paths that are optimal with respect to some physical model of the earth.
- By observing how these waves travel during an earthquake, we can infer things about the composition of the earth.

Formal Setting

We consider the inverse of the *(primal) value function* ϕ_P, defined as

$$\phi_P(d) = \min_{x \in \mathcal{S}} d^\top x = \min_{x \in \text{conv}(\mathcal{S})} d^\top x \ \forall d \in \mathbb{R}^n. \tag{PVF}$$

With respect to a given $x^0 \in \mathcal{S}$, the inverse problem is defined as

$$\min \left\{ f(d) \;\middle|\; d^\top x^0 = \phi_P(d) \right\}, \tag{INV}$$

- The classical objective function is taken to be $f(d) = \|c - d\|$, where $c \in \mathbb{R}^n$ is a given target.
- If we take $f(d) = d^\top x^0 - d^\top x^*$ for given $x^* \in \mathbb{R}^n$, then this is equivalent to what [Padberg and Grötschel(1985)] called the *primal separation problem* (see also [Lodi and Letchford(2003)]).

A Small Example

- The feasible set of the inverse problem is the set of objective vectors that make x^0 optimal.
- This is precisely the dual of cone$(\mathcal{S} - \{x^0\})$, which is, roughly, a translation of the polyhedron described by the inequalities binding at x^0.

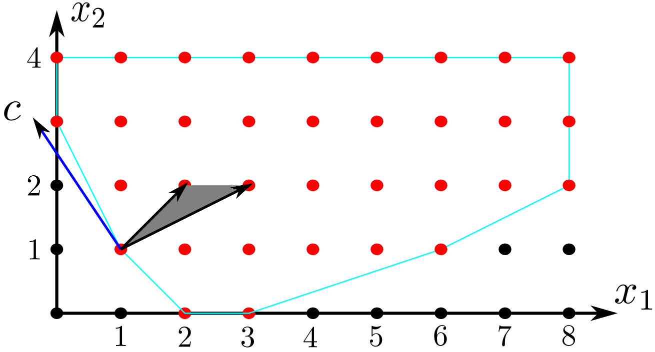


Figure: conv$(\mathcal{S})$ and cone $\mathcal{D}$ of feasible objectives

Inverse Optimization as a Mathematical Program

- To formulate as a mathematical program, we need to represent the implicit constraints of (INV) explicitly.
- The cone of feasible objective vectors can be described as

$$\mathcal{D} = \left\{ d \in \mathbb{R}^n \;\middle|\; d^\top x^0 \leq d^\top x \;\forall x \in \mathcal{S} \right\} \tag{IFS}$$

- Since $\mathcal{P}$ is bounded, we need only the inequalities corresponding to extreme points of $\text{conv}(\mathcal{S})$.
- This set of constraints is exponential in size, but we can generate them dynamically, as we will see.

Formulating the Inverse Problem

General Formulation

$$
\begin{aligned}
\min \quad & f(d) & & \\
s.t. \quad & d^\top x^0 \leq d^\top x & \forall x \in \mathcal{E} & \qquad \text{(INVMP)}
\end{aligned}
$$

- With $f(d) = \|c - d\|$, this can be linearized for ℓ_1 and ℓ_∞ norms.
- Note that this is the separation problem for the conic hull of $\mathcal{S} - \{x^0\}$.
- If x^0 is an extreme point of $\mathcal{P}$, then this is nothing more than the corner polyhedron associated with x^0 wrt $\text{conv}(\mathcal{S})$.
- Note that if we take $f(d) = d^\top x^0 - d^\top x^*$, as mentioned earlier, then we need a normalization to ensure $d \neq 0$.
- A straightforward option is to take $1^\top d = 1$.

Separation and Inverse Optimization

- It should be clear that inverse optimization and separation are very closely related.
- First, note that the inequality

$$\pi^\top x \geq \pi_0 \qquad \text{(PI)}$$

is valid for $\mathcal{P}$ if and only if $\pi_0 \leq \phi_P(\pi)$.

- We refer to inequalities of the form (PI) for which $\pi_0 = \phi_P(\pi)$ as *primal inequalities*.
- This is as opposed to *dual inequalities* for which $\pi_0 = \phi_D^\pi(b)$, where ϕ_D^π is the *dual value function* of (MILP) with objective function π.
- The feasible set of (INV) can be seen as the set of all valid primal inequalities that are tight at x^0.

Dual of the Inverse Problem

- Roughly speaking, the dual of (INVMP) is the membership problem for $\mathrm{cone}(\mathcal{S} - \{x^0\})$.

$$\min_{\lambda \in \mathbb{R}^{\mathcal{E}}_{+}} \left\{ 0^\top \lambda \;\middle|\; \bar{E}\lambda = x^* - x^0 \right\} \qquad \text{(CMEM)}$$

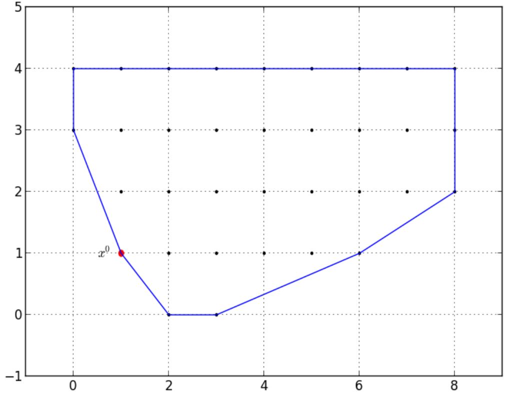
- With the normalization, this becomes

$$\min_{\lambda \in \mathbb{R}^{\mathcal{E}}_{+}} \left\{ \alpha \;\middle|\; \bar{E}\lambda + \alpha 1 = x^* - x^0 \right\}, \qquad \text{(CMEMN)}$$

where $\bar{E}$ is the set of extreme rays of the conic hull of $\mathrm{conv}(\mathcal{S}) - \{x^0\}$

Inverse Optimization with Forward Optimization Oracle

- We can use an algorithm almost identical to the one from earlier.
- We now generate inequalities valid for the corner polyhedron associated with x^0.



Inverse Example: Iteration 1

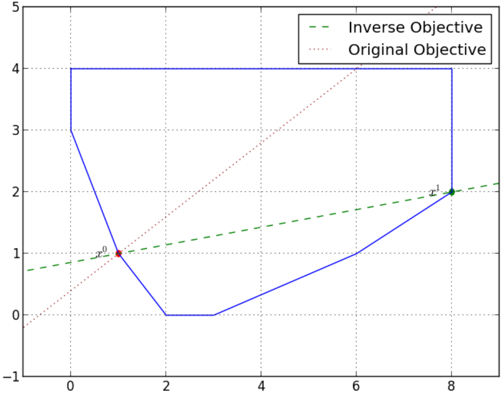


Figure: Solving the inverse problem for $\mathcal{P}$ (Iteration 1)

Inverse Example: Iteration 2

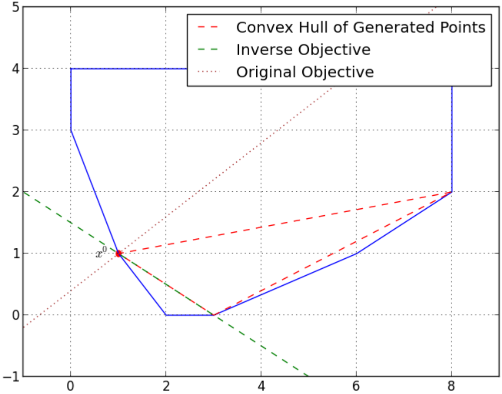


Figure: Solving the inverse problem for $\mathcal{P}$ (Iteration 3)

Inverse Example: Iteration 3

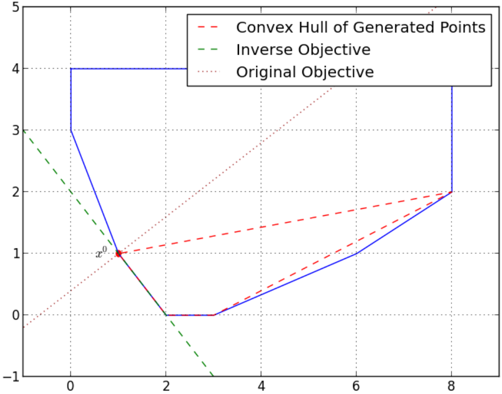


Figure: Solving the inverse problem for $\mathcal{P}$ (Iteration 3)

Solvability of Inverse MILP

Theorem 1 *[Bulut and R(2015)] Inverse MILP optimization problem under ℓ_∞/ℓ_1 norm is solvable in time polynomial in the size of the problem input, given an oracle for the MILP decision problem.*

- This is a direct result of the well-known result of [Grötschel et al.(1993)Grötschel, Lovász, and Schrijver].
- GLS does not, however, tell us the formal complexity.

Formal Complexity of Inverse MILP

Sets

$$\mathcal{K}(\gamma) = \{d \in \mathbb{R}^n \mid \|c - d\| \leq \gamma\}$$

$$\mathcal{X}(\gamma) = \{x \in \mathcal{S} \mid \exists d \in \mathcal{K}(\gamma) \ s.t. \ d^\top (x - x^0) > 0\},$$

$$\mathcal{K}^*(\gamma) = \{x \in \mathbb{R}^n \mid d^\top (x - x^0) \geq 0 \ \forall d \in \mathcal{K}(\gamma)\}.$$

Inverse MILP Decision Problem (INVD)

Inputs: γ, c, $x^0 \in \mathcal{S}$ and MILP feasible set $\mathcal{S}$.
Problem: Decide whether $\mathcal{K}(\gamma) \cap \mathcal{D}$ is non-empty.

Theorem 2 *[Bulut and R(2015)] INVD is* coNP*–complete.*

Theorem 3 *[Bulut and R(2015)] Both* (MILP) *and* (INV) *optimal value problems are* D^p*–complete.*

Connections to Constraint Decomposition

As usual, we divide the constraints into two sets.

$$
\begin{aligned}
\min\ & c^\top x \\
\text{s.t. } A'x &\leq b' \text{ (the "nice" constraints)} \\
A''x &\leq b'' \text{ (the "complicating" constraints)} \\
x &\in \mathbb{Z}^n
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{P}' &= \{x \in \mathbb{R}^n \mid A'x \leq b'\}, \\
\mathcal{P}'' &= \{x \in \mathbb{R}^n \mid A''x \leq b''\}, \\
\mathcal{P} &= \mathcal{P}' \cap \mathcal{P}'', \\
\mathcal{S} &= \mathcal{P} \cap \mathbb{Z}^n, \text{ and} \\
\mathcal{S}_R &= \mathcal{P}' \cap \mathbb{Z}^n.
\end{aligned}
$$

Reformulation

- Using an approach similar to that used in the linear programming case, we can obtain the following reformulation.

$$\min \quad c^\top x \tag{2}$$

$$\text{s.t.} \quad \sum_{s\in\mathcal{E}} \lambda_s s = x \tag{3}$$

$$A'' x \leq b'' \tag{4}$$

$$\sum_{s\in\mathcal{E}} \lambda_s = 1 \tag{5}$$

$$\lambda \in \mathbb{R}_+^{\mathcal{E}} \tag{6}$$

$$x \in \mathbb{Z}^n \tag{7}$$

where $\mathcal{E}$ is the set of extreme points of $\text{conv}(\mathcal{S}_R)$.

- If we relax the integrality consraints (7), then we can also drop (3) and we obtain a relaxation which is tractable.
- This relaxation may yield a bound better than that of the LP relaxation.

The Decomposition Bound

Using the aformentioned relaxation, we obtain a formulation for the so-called *decomposition bound*.

$$
\begin{aligned}
z_{\text{IP}} &= \min_{x \in \mathbb{Z}^n} \left\{ c^\top x \mid A'x \leq b', A''x \leq b'' \right\} \\
z_{\text{LP}} &= \min_{x \in \mathbb{R}^n} \left\{ c^\top x \mid A'x \leq b', A''x \leq b'' \right\} \\
z_{\text{D}} &= \min_{x \in \text{conv}(\mathcal{S}_R)} \left\{ c^\top x \mid A''x \leq b'' \right\}
\end{aligned}
$$

$$
z_{\text{IP}} \geq z_{\text{D}} \geq z_{\text{LP}}
$$

It is well-known that this bound can be computed using various decomposition-based algorithms:

- Lagrangian relaxation
- Dantzig-Wolfe decomposition
- Cutting plane method

Shameless plug: Try out DIP/DipPy!

A framework for switching between various decomp-based algorithms.

Lagrange Cuts

- [Boyd(1990)] observed that for $u \in \mathbb{R}^m_-$, a *Lagrange cut* of the form

$$(c - uA'')^\top x \geq LR(u) - ub'' \tag{LC}$$

  is valid for $\mathcal{P}$.

- If we take u^* to be the optimal solution to the Lagrangian dual, then this inequality reduces to

$$(c - u^*A'')^\top x \geq z_D - ub'' \tag{OLC}$$

- If we now take

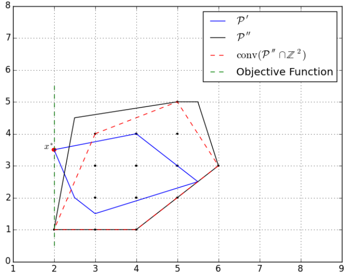
$$x^D \in \operatorname{argmin}\left\{c^\top x \mid A''x \leq b'', (c - u^*A'')^\top x \geq z_D - ub''\right\},$$

  then we have $c^\top x^D = z_D$.

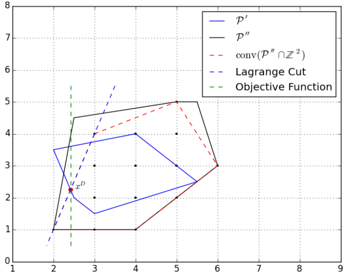
Connecting the Dots

Results

- The inequality (OLC) is a primal inequality for $\text{conv}(\mathcal{S}_R)$ wrt x^D.
- $c - uA''$ is a solution to the inverse problem wrt $\text{conv}(\mathcal{S}_R)$ and x^D.
- These properties also hold for $e \in \mathcal{E}$ such that $\lambda_e^* > 0$ in the RMP.



(a) Original LP relaxation



(b) After adding Langrange cut

Conclusions and Future Work

- We gave a brief overview of commnections between a number of different problems and methodologies.
- Exploring these connections may be useful to impoving intuition and understanding.
- The connection to primal cutting plane algorithms is still largely unexplored, but this should lead to new algorithms for the inverse problem.
- We did not touch much on complexity, but it should be possible to generalize complexity results to the separation/optimization context.
- We believe GLS can be extended to show that inverse optimization forward optimization, and and separation are all complexity-wise equivalent.
- Much of that is discussed here can be further generalized to general computation via Turing machines (useful?).

Shameless Promotion: Free Stuff!

- CHiPPS: Parallel tree search framework
- DIP/DipPy: Decomposition-based modeling language and MILP solver
- DiSCO, OsiConic, CglConic: Mixed integer conic solver
- MibS: Mixed integer bilevel solver
- SYMPHONY: MILP solver framework with bicriteria, MILP dual construction. warm starting, continuous bilevel (soon to come), etc.
- GiMPy, GrUMPy: Visualizations and illustrative implementations for graph and optimization algorithms.
- CuPPy: Cutting planes in Python
- Value Function: Algorithm for constructing value functions
- And more...

`http://github.com/tkralphs`

References I

Boyd, A. 1990.
The lagrangian and other primal cutting planes for linear integer programming.
Technical Report TR90-03, Rice University, Department of Computational and Applied Mathematics.

Bulut, A. and T. R 2015.
On the complexity of inverse milp.
Technical report, COR@L Laboratory, Lehigh University.

Grötschel, M., L. Lovász, and A. Schrijver 1993.
Geometric Algorithms and Combinatorial Optimization, volume 2 of *Algorithms and Combinatorics*.
Springer, second corrected edition edition.
ISBN 3-540-56740-2, 0-387-56740-2 (U.S.).

References II

Güzelsoy, M. and T. K. R 2007.
Duality for mixed-integer linear programs.
International Journal of Operations Research **4**, 118–137.

Lodi, A. and A. Letchford 2003.
Primal separation algorithms.
4OR **1**, 209–224.

Padberg, M. and M. Grötschel 1985.
Polyhedral computations.
In E. L. Lawler, J. K. Lenstra, A. H. G. R. Kan, and D. B. Shmoys, editors, *The Traveling Salesman Problem. A Guided Tour of Combinatorial Optimization*, chapter 9, pages 307–360. Wiley.