Optimization, Separation, and Inverse Optimization

Ted Ralphs
Aykut Bulut

COR@L
COMPUTATIONAL OPTIMIZATION RESEARCH AT LEHIGH

LEHIGH UNIVERSITY

INFORMS Annual Meeting
San Francisco, CA, 12 November 2014

What is an Inverse Problem?

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 value function

$$z_{IP}(d) = \min_{x \in \mathcal{S}} d^T x = \min_{x \in \text{conv}(\mathcal{S})} d^T x \tag{1}$$

for $d \in \mathbb{R}^n$, where

$$\mathcal{S} = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\} \cap (\mathbb{Z}^r \times \mathbb{R}^{n-r})\}. \tag{2}$$

for $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$. With respect to a given target $c \in \mathbb{R}^n$ and a given $x^0 \in \mathcal{S}$, the inverse problem is defined as

$$\min\{\|c - d\| \mid d^\top x^0 = z_{IP}(d)\} \tag{INV}$$

Assumption: $\mathcal{S}$ is bounded (for simplicity of presentation).

Formulating as a Mathematical Program

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

$$\mathcal{D} = \{d \in \mathbb{R}^n \mid d^T x^0 \leq d^T x \; \forall x \in \mathcal{S}\}. \quad (3)$$

- In the pure integer case, this is a finite number of inequalities and the above is thus an linear program (LP).
- Even in the mixed case, 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.

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 polar of the cone of facet-defining inequalities of $\text{conv}(\mathcal{S})$ that are binding at x^0.

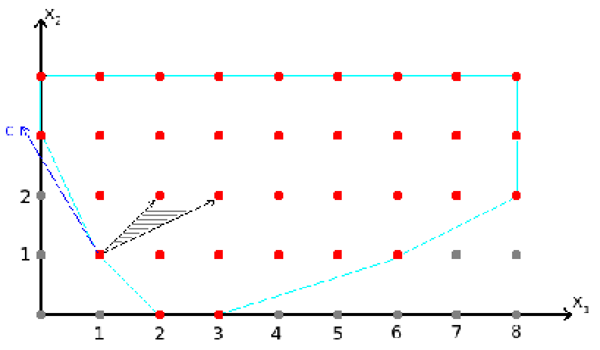


Figure: $\text{conv}(\mathcal{S})$ and cone $\mathcal{D}$ of feasible objectives

The Separation Problem

- For reasons that will become clear, we now consider the *separation problem* for a polyhedron.

Separation Problem

Given a polyhedron $\mathcal{P} \subseteq \mathbb{R}^n$ and $x^* \in \mathbb{R}^n$, determine whether $x^* \in \mathcal{P}$ and if not, determine (π, π_0), a valid inequality for $\mathcal{P}$ such that $\pi x^* > \pi_0$.

- The separation problem can be formulated as

$$\max\{\pi x^* - \pi_0 \mid \pi^\top x \leq \pi_0 \ \forall x \in \mathcal{P}, (\pi, \pi_0) \in \mathbb{R}^{n+1}\} \tag{4}$$

along with some appropriate normalization.

- When $\mathcal{P}$ is a polytope, we can reformulate this problem as the LP

$$\max\{\pi x^* - \pi_0 \mid \pi^\top x \leq \pi_0 \ \forall x \in \mathcal{E}, (\pi, \pi_0) \in \mathbb{R}^{n+1}\} \tag{5}$$

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

The Polar

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

$$\mathcal{P}^* = \{\pi \in \mathbb{R}^n \mid \pi^\top x \leq 1 \ \forall x \in \mathcal{P}\} \tag{6}$$

and is called its *polar*.

Properties of the 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

$$\mathcal{P}^* = \{\pi \in \mathbb{R}^n \mid \pi^\top x \leq 1 \ \forall x \in \mathcal{E}, \pi^\top r \leq 0 \ \forall r \in \mathcal{R}\}.$$

- 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 the separation problem (5) using a cutting plane algorithm.
- The separation problem for the polar of a polyhedron $\mathcal{P}$ is precisely a linear optimization problem over $\mathcal{P}$.
- Example

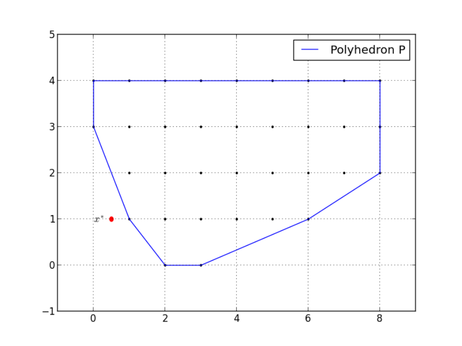


Figure: x^* (to be separated) and $\mathcal{P}$

Separation Example: Iteration 1

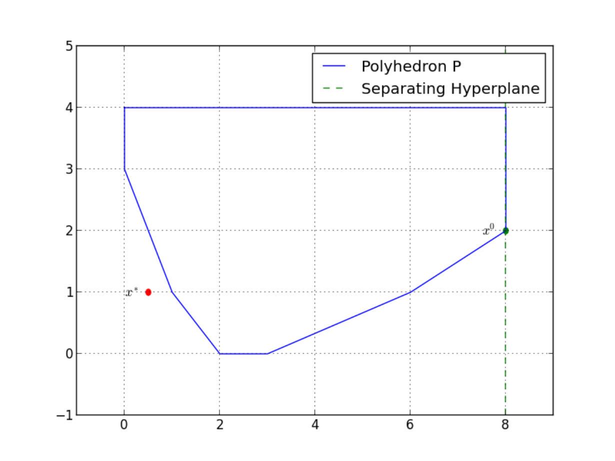


Figure: Separating x^* from $\mathcal{P}$ (Iteration 1)

Separation Example: Iteration 2

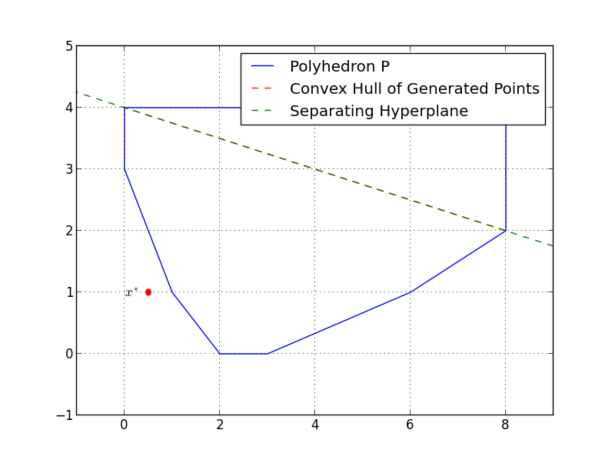


Figure: Separating x^* from $\mathcal{P}$ (Iteration 2)

Separation Example: Iteration 3

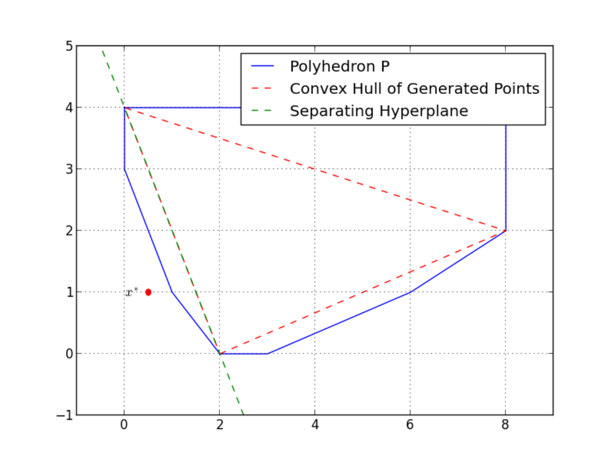


Figure: Separating x^* from $\mathcal{P}$ (Iteration 3)

Separation Example: Iteration 4

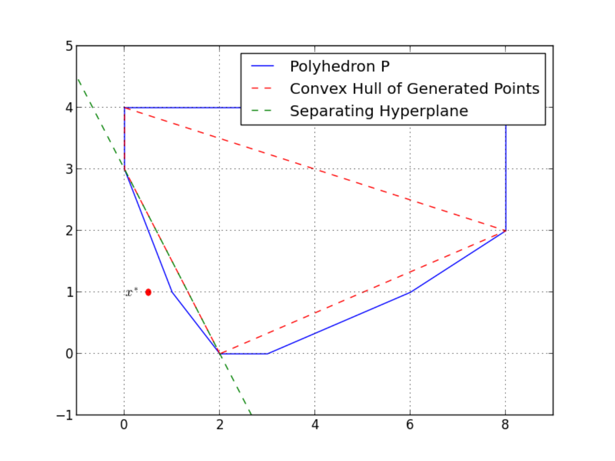


Figure: Separating x^* from $\mathcal{P}$ (Iteration 4)

Separation Example: Iteration 5

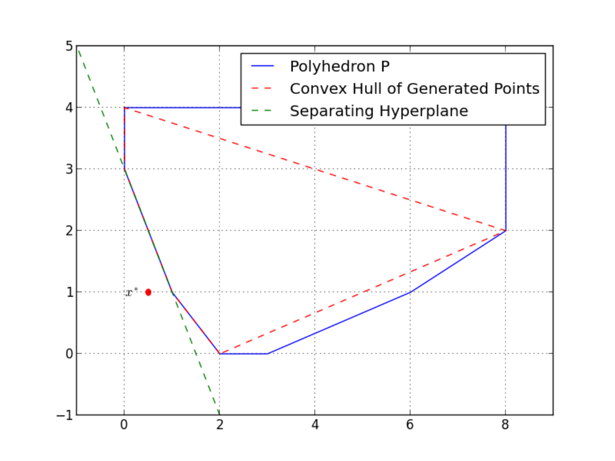


Figure: Separating x^* from $\mathcal{P}$ (Iteration 5)

Solving the Inverse Problem Using Optimization Oracle

- The feasible solution to the inverse problem are essentially valid inequalities that are satisfied at equality by x^0.
- What happens if we use the same algorithm to "separate" x^0?

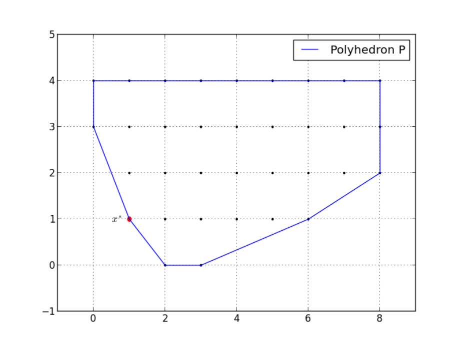


Figure: x^0 and $\mathcal{P}$

Inverse Example: Iteration 1

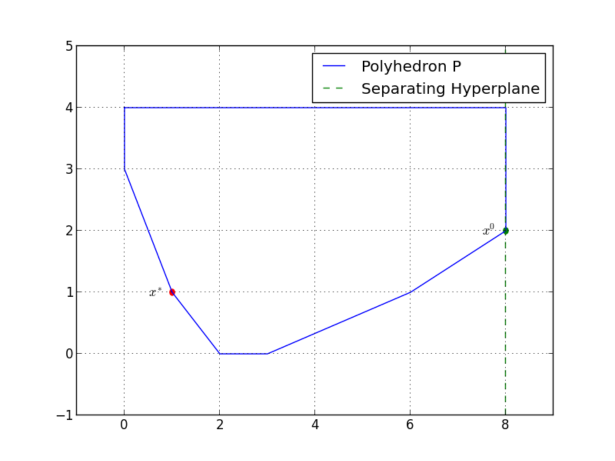


Figure: Solving the inverse problem for $\mathcal{P}$ (Iteration 1)

Inverse Example: Iteration 2

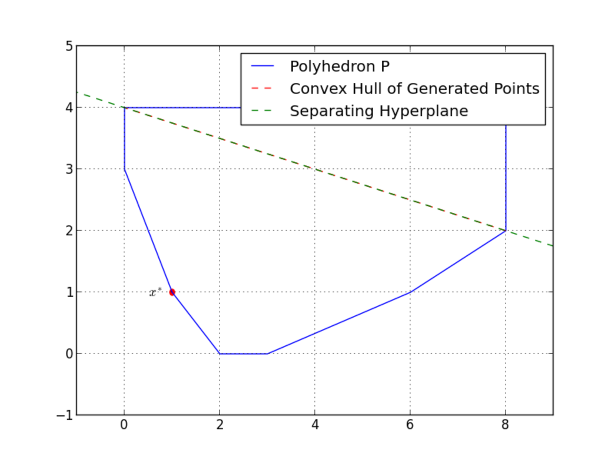


Figure: Solving the inverse problem for $\mathcal{P}$ (Iteration 2)

Inverse Example: Iteration 3

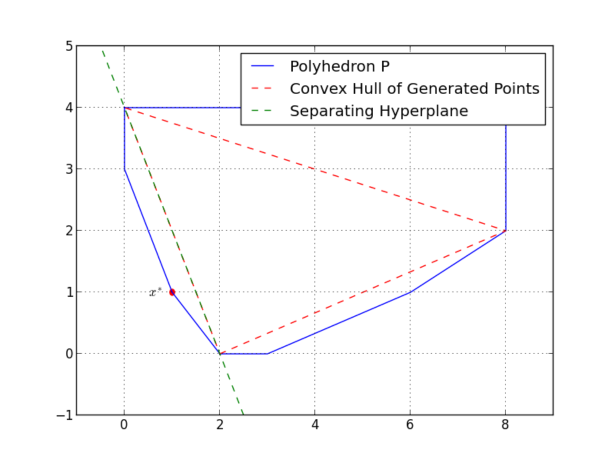


Figure: Solving the inverse problem for $\mathcal{P}$ (Iteration 3)

Inverse Example: Iteration 4

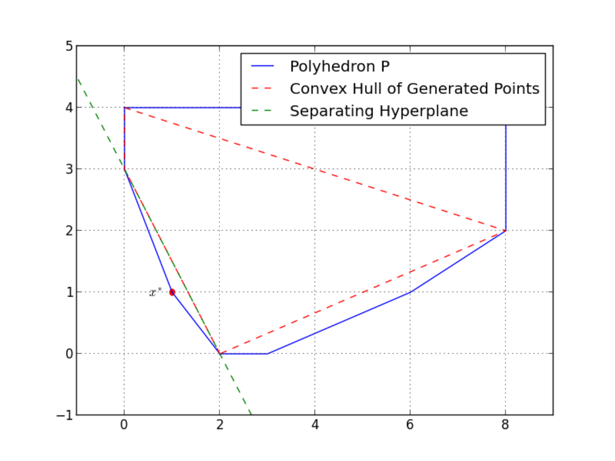


Figure: Solving the inverse problem for $\mathcal{P}$ (Iteration 4)

Inverse Example: Iteration 5

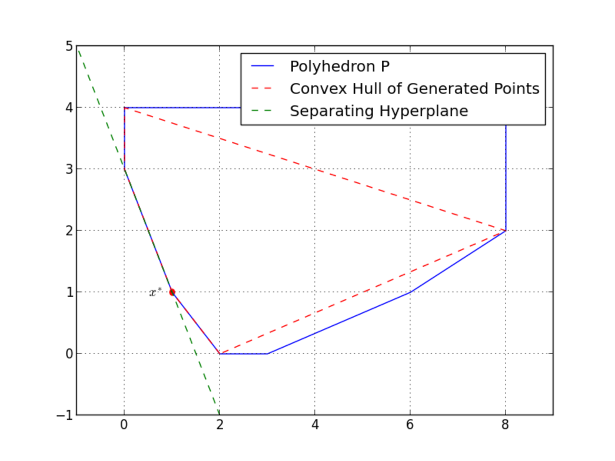


Figure: Solving the inverse problem for $\mathcal{P}$ (Iteration 5)

Separation Problems and Inverse Problems

- The separation problem produces a feasible solution to the inverse problem, but it may not be the one we want.
- Since we know that the optimal solution value of the separation problem will be zero, we don't actually need to optimize.
- We can exchange the objective function for one that produces the inequality we want.

Back to the Inverse Problem

General Formulation

$$\begin{aligned} \min \quad & \|c - d\| & \\ s.t. \quad & d^T x^0 \leq d^T x & \forall x \in \mathcal{S} \end{aligned} \tag{7}$$

- Model can be linearized for ℓ_1 and ℓ_∞ norms.
- Convex hull of $\mathcal{S}$ is a polytope.
- Last constraint set can be represented with the set of extreme points of convex hull of $\mathcal{S}$.
- Let $\mathcal{E}$ be the set of extreme points of convex hull of $\mathcal{P}$, $\mathcal{E}$ is finite.

Inverse MILP with ℓ_1 norm

Linearized ℓ_1 Formulation

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{n} \theta_i & & \\
s.t. \quad & c_i - d_i \le \theta_i & \forall i \in \{1, 2, \ldots, n\} \quad & (8) \\
& d_i - c_i \le \theta_i & \forall i \in \{1, 2, \ldots, n\} \quad & (9) \\
& d^T x^0 \le d^T x & \forall x \in \mathcal{E}. &
\end{aligned}
$$

Inverse MILP with ℓ_∞ norm

Linearized ℓ_∞ Formulation

$$
\begin{aligned}
\min \quad & y & & \\
s.t. \quad & c_i - d_i \le y & & \forall i \in \{1, 2, \ldots, n\} \\
& d_i - c_i \le y & & \forall i \in \{1, 2, \ldots, n\} \qquad (10) \\
& d^T x^0 \le d^T x & & \forall x \in \mathcal{E}.
\end{aligned}
$$

For the remainder of the presentation, we deal with the case of ℓ_∞ norm.

Inverse Example

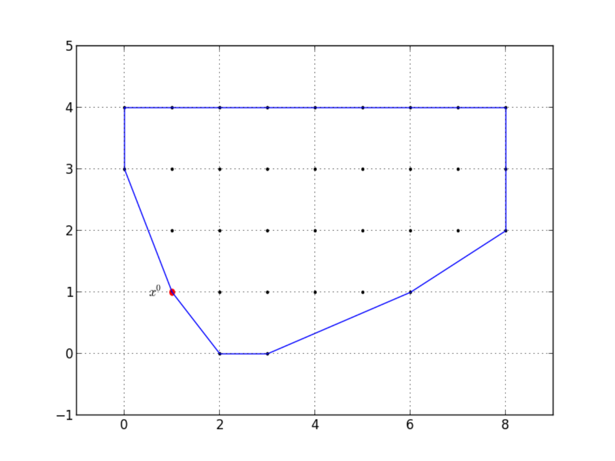


Figure: x^0 and $\mathcal{P}$

Inverse Example: Iteration 1

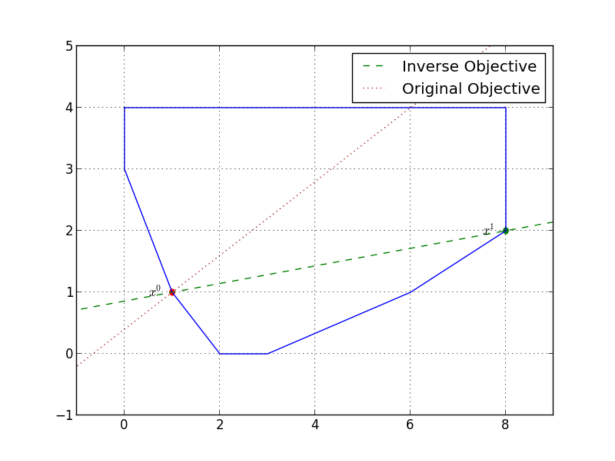


Figure: Solving the inverse problem for $\mathcal{P}$ (Iteration 1)

Inverse Example: Iteration 2

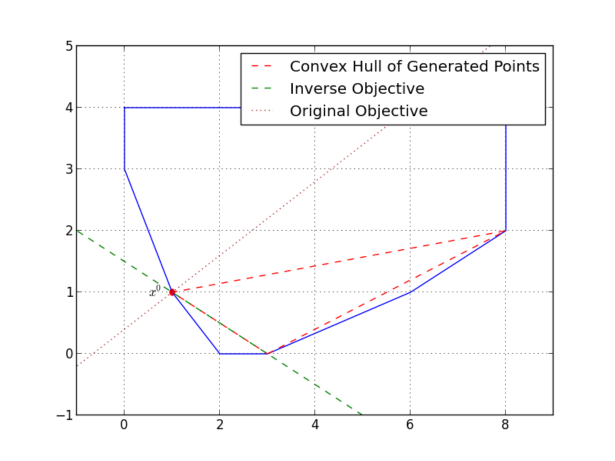


Figure: Solving the inverse problem for $\mathcal{P}$ (Iteration 3)

Inverse Example: Iteration 3

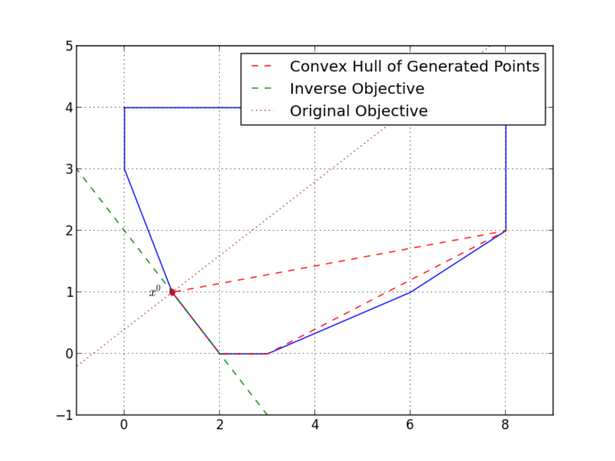


Figure: Solving the inverse problem for $\mathcal{P}$ (Iteration 3)

Observations

- Note the similarity to the separation problem.
- One difference is that x^0 is implicitly include among the extreme points found so far.
- In each iteration, we generate a valid inequality for the convex hull of all points found and x^0.
- We also require x^0 to be on the associated face.
- The optimization guides the process to produce the optimal objective vector.

Solvability of Inverse MILP

Theorem 1 (Bulut, R. '14) *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 GLS.
- Separation over the polar is optimization over the original polyhedron.

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 (INV)

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

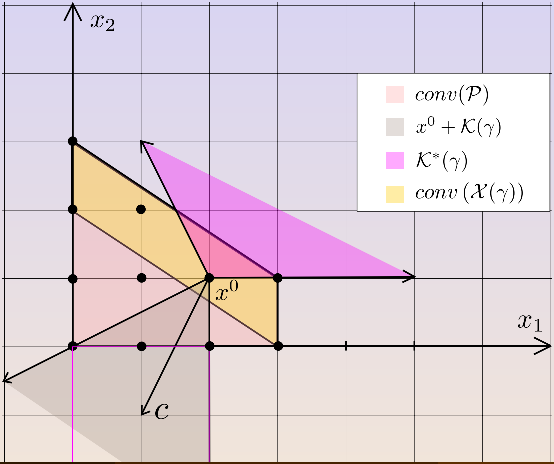
Theorem 2 (Bulut, R. '14) *INV is* coNP*–complete.*

Proof of Formal Complexity

- We prove this result by showing the existence of a short certificate for the negative answer.
- When answer is NO, then $\text{conv}(\mathcal{X}(\gamma)) \cap \textit{int}(\mathcal{K}^*(\gamma)) \neq \emptyset$.
- We thus need short certificates for membership in $\text{conv}(\mathcal{X}(\gamma))$ and $\text{int}(\mathcal{K}^*(\gamma))$.
- These are both polyhedral sets for which we have explicit descriptions.

Proof of Formal Complexity

- The figure shows the relevant sets under for a maximization problem under the ℓ_∞ norm for the pure integer case.
- In this case, the answer is negative.



Conclusions and Generalization

- This entire framework can be generalized to algorithms on Turing machines.
- Inverse problems have a natural interpretation in this setting.
- A “separation problem” can be defined in terms of the distance from a given target instance to a solution to the inverse problem.
- Is this a useful generalization? It seems to be an interesting extension of the usual complexity framework, but it remains to be seen...
- We have developed a framework that interprets solution of inverse problems in terms of problems we already know.
- We expect to be able to improve the solution of inverse problems using this insight.