On the Complexity of Inverse Mixed Integer Linear Optimization

Aykut Bulut and Ted K. Ralphs

Department of Industrial and Systems Engineering, Lehigh University, USA

COR@L Technical Report 15T-001-R4

On the Complexity of Inverse Mixed Integer Linear Optimization

Aykut Bulut*[1] and Ted K. Ralphs†[1]

[1]Department of Industrial and Systems Engineering, Lehigh University, USA

Original Publication: March 13, 2015
Last Revised: October 30, 2020

Abstract

Inverse optimization is the problem of determining the values of missing input parameters that are closest to given estimates and that will make a given solution optimal. This study is concerned with the relationship of a particular inverse mixed integer linear optimization problem (MILPs) to both the original problem and the separation problem associated with its feasible region. We show that the decision version of the inverse MILP is coNP–complete (extending the result ofAhuja and Orlin [2001] to the discrete case) and that the optimal value verification problem for both the inverse problem *and* the associated *forward* problem are both complete for the complexity class D^{P}. We also describe a cutting plane algorithm for solving inverse MILPs that illustrates the close relationship between the inverse problem and the separation problem for the convex hull of solutions to the forward problem. The inverse problem is shown to be equivalent to the separation problem for the radial cone defined by all inequalities valid for the convex hull of solutions to the MILP that are binding at the solution serving as input to the inverse problem. Thus, the inverse, forward, and separation problems can be said to be equivalent.

Keywords: Inverse optimization, mixed integer linear optimization, computational complexity, polynomial hierarchy

1 Introduction

In this paper, we study the relationship of the inverse integer linear optimization problem to both the optimization problem from which it arose and the associated separation problem. We show that these three problems have a strong relationship from an algorithmic standpoint by describing a cutting-plane algorithm for the inverse problem that uses the forward problem as an oracle. A modified version of this algorithm also solves the separation problem. From a complexity standpoint, we show that certain decision versions of these three problems are all complete for the complexity class D^{P}, introduced originally by Papadimitriou and Yannakakis [1982]. As such, we propose that D^{P} (and associated classes) may provide a more natural way of classifying difficult optimization problems than the more commonly used NP-hard.

An *optimization problem* is that of determining a member of a feasible set (an *optimal solution*) that minimizes the value of a given objective function. The feasible set is typically described as the points in a vector space satisfying a given set of equations, inequalities, and disjunctions (the latter are usually in the form of a requirement that the value of a certain element of the solution take on an integral value).

*E-mail: aykutblt@gmail.com
†E-mail: ted@lehigh.edu

An *inverse optimization problem*, in contrast, is a related problem in which we assume that the description of the original optimization problem, which we refer to as the *forward problem* in order to distinguish the two, is incomplete (some parameters are missing or cannot be observed), but a full or partial solution *can* be observed. The goal is to determine values for the missing parameters with respect to which the given solution would be optimal for the resulting problem. Estimates for the missing parameters may be given, in which case the goal is to produce a set of parameters that is as close to the given estimates as possible by some given metric.

The forward optimization problem of interest in this paper is the *mixed integer linear optimization problem*

$$\max_{x \in \mathcal{S}} d^\top x \tag{MILP}$$

where $d \in \mathbb{Q}^n$ and

$$\mathcal{S} = \{x \in \mathbb{R}^n \mid Ax \leq b\} \cap (\mathbb{Z}^r \times \mathbb{R}^{n-r}). \tag{1}$$

for $A \in \mathbb{Q}^{m \times n}$, $b \in \mathbb{Q}^m$ and for some non-negative integer r. (MILP) is known simply as a linear optimization problem (LP) in the case when $r = 0$.

One can define a number of different inverse problems associated with (MILP), depending on what parts of the description (A, b, d) are unknown and what form the objective function of the inverse problem takes. Here, we study the case in which the objective function d of the forward problem is the unknown element of the input, but we are given A and b, as well as a *target vector* $x^0 \in \mathbb{Q}^n$. A feasible solution to the inverse problem (which we refer to as a *feasible objective*) is any $d^* \in \mathbb{Q}^n$ for which x^0 would be optimal if d^* were the missing objective function.

It is important in the analysis that follows to be precise about the assumptions on the target vector x^0. The problem as informally introduced above implicitly assumes that $x^0 \in \mathcal{S}$, since otherwise, x^0 cannot technically be an optimal solution, regardless of the objective function chosen. On the other hand, the mathematical formulations we introduce shortly do not require $x^0 \in \mathcal{S}$ and can be interpreted even when $x^0 \notin \mathcal{S}$. As a practical matter when solving inverse problems in practice, this subtle distinction is usually not very important, since membership in $\mathcal{S}$ can be verified in a pre-processing step if necessary. However, in the context of complexity analysis and in considering the relationship of the inverse problem to other well-known problems, this point *is* important and we will return to it. For example, without this assumption, the inverse optimization problem can be seen to be equivalent to the problem of verifying a given lower bound on the optimal solution value of an MILP and also to the separation problem.

For these and other reasons that will become clear, we do not assume $x^0 \in \mathcal{S}$, but this makes the definition of the problem actually being a solved a bit ambiguous. To resolve any ambiguity, we simply take the feasible region of the forward problem to be $\mathcal{S} \cup \{x^0\}$. The problems is then to find an objective function for which x^0 is optimal over this augmented feasible region (of which it is obviously guaranteed to be a member).

1.1 Formulations

We now describe several rigorous mathematical formulations of what we refer to from now on as the *inverse mixed integer linear optimization problem* (IMILP). A strightforward way of formalizing this problem that explains why we refer to this general class of problem as "inverse" problem is as that of computing the mathematical inverse of a certain *value function*. A value function is one that expresses either the optimal value (typically) or the set of all optimal solutions to an optimization problem, as a function of input parameters. In this case, the relevant value function is

$$\phi(d) = \operatorname{argmax}_{x \in \mathcal{S} \cup \{x^0\}} d^\top x.$$

In terms of the value function ϕ, a feasible objective is any element of the preimage $\phi^{-1}(x^0)$. To make the IMILP an optimization problem in itself, we add an objective function to obtain the general formulation

$$\min_{d \in \phi^{-1}(x^0)} f(d) \tag{INV}$$

where $f : \mathbb{Q}^n \to \mathbb{Q}$ is the chosen objective function.

The traditional objective function used for inverse problems in the literature is $f(d) := \|c - d\|$, the minimum norm distance from d to a certain given target $c \in \mathbb{Q}^n$ (the specific norm is not important for defining the problem, but later, we'll assume a p-norm when proving some results). This choice of objective, although standard, is not scale-invariant—scaling a given feasible objective d^* changes the resulting objective function value). This has implications we discuss further below.

The formulation (INV) does not suggest any direct connection to existing methodology for solving mathematical optimization problems, so we next discuss several alternative formulations of the problem as a standard mathematical optimization problem. We first consider the following formulation of the IMILP as the semi-infinite optimization problem

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

In (INVMILP), d is a vector of variables, while $c \in \mathbb{Q}^n$ is the estimate or target value. Note that in (INVMILP), if we instead let x^0 vary, replacing it with a variable x, and interpret d as a fixed objective function, replacing $\|c - d\|$ with the objective $d^\top x$ of the forward problem, we get a reformulation of the forward problem (MILP) itself. This formulation can be made finite, in the case that $\mathcal{S}$ is bounded, by replacing the possibly infinite set of inequalities with only those corresponding to the extreme points of $\text{conv}(\mathcal{S})$. In the unbounded case, we also to include inequalities corresponding to the extreme rays.

Problem (INVMILP) can also be formulated as a conic problem. In terms of the conic sets

$$\begin{aligned} &\mathcal{K} := \{(y, d) \in \mathbb{R}^{n+1} \mid \|c - d\| \leq y\}, \\ &\mathcal{K}(\gamma) := \{d \in \mathbb{R}^n \mid (\gamma, d) \in \mathcal{K}\}, \text{ and} \\ &\mathcal{D}(x^0) := \{d \in \mathbb{R}^n \mid d^\top (x - x^0) \leq 0 \; \forall x \in \mathcal{S}\}, \end{aligned}$$

(INVMILP) can be reformulated as

$$\min_{d \in \mathcal{K}(y) \cap \mathcal{D}(x^0)} y. \tag{INVMILP-C}$$

Here, $\mathcal{K}$ is a general norm cone and $\mathcal{D}(x^0)$ is a linear cone for a given x^0. The set $\mathcal{D}(x^0)$ is the feasible set of the IMILP with target vector x^0 (recall that its members are called *feasible objectives*). Note that $\mathcal{D}(x^0)$ is precisely the polar of $\text{cone}(\mathcal{S} - \{x^0\})$. The notational dependence on x^0 is for convenience later when various target vectors will be constructed in the reductions used in the complexity proofs.

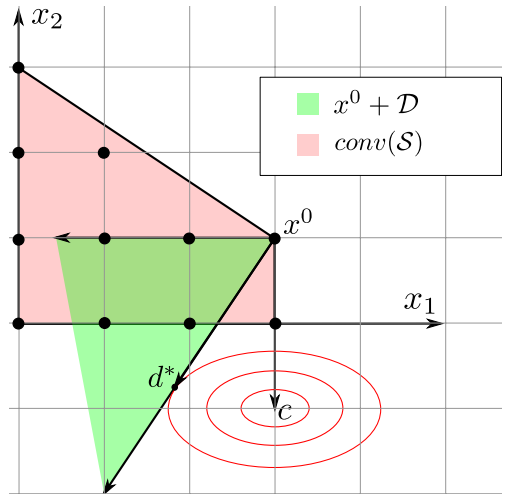
The interpretation of the set $\mathcal{D}(x^0)$ as the polar of $\text{cone}(\mathcal{S} - \{x^0\})$ leads to a third formulation in terms of the so-called 1-polar of $\text{conv}(\mathcal{S})$. Assuming $\text{conv}(\mathcal{S})$ is a polytope, its 1-polar is defined as

$$\mathcal{P}^1 = \{d \in \mathbb{R}^n \mid d^\top x \leq 1 \; \forall x \in \text{conv}(\mathcal{S})\}.$$

When $\text{conv}(\mathcal{S})$ is full-dimensional and $0 \in \text{int}\,(\text{conv}(\mathcal{S}))$, the 1-polar is the set of all normalized inequalities valid for $\text{conv}(\mathcal{S})$ (see [Schrijver, 1986] for definitions). Under these assumptions, (INVMILP) can also be reformulated as

$$\begin{aligned} \min \quad & \|c - \rho d\| \\ \text{s.t.} \quad & d \in \mathcal{P}^1 \\ & d^\top x^0 \geq 1 \\ & \rho \geq 0. \end{aligned} \tag{INVMILP-1P}$$

Figure 1: Two dimensional inverse MILP



In formulation (INVMILP-1P), ρ is a multiplier on d that allows scaling in order to improve the objective function value. We might also require $\|c\| = 1$ or normalize in some other way to avoid this scaling. The constraint $d^\top x^0 \geq 1$ ensures that d is feasible to (INVMILP). Observe also that relaxing the constraint $d^\top x^0 \geq 1$ yields a problem similar to the classical separation problem, but with a different objective function. We revisit this idea in Section 2.

Figure 1 illustrates the geometry of the inverse MILP. $\mathcal{S}$ is a discrete set indicated by the black dots. The vector $c = (0, -2)$ and $x^0 = (3, 1)$. The convex hull of $\mathcal{S}$ and the cone $\mathcal{D}(x^0)$ (translated to x^0) are shaded. The ellipsoids show the sets of points with a fixed distance to $x^0 + c$ for some given norm. The optimal solution for this example is vector d^*, and point $x^0 + d^*$ is also illustrated.

Note that when x^0 is in the relative interior of $\text{conv}(\mathcal{S})$, any objective vector in the subspace orthogonal to the affine space containing $\text{conv}(\mathcal{S})$ is feasible for the inverse problem, i.e., optimizes x^0. If we let $c_\mathcal{S}$ be the projection of c onto the smallest affine space that contains $\mathcal{S}$ and $c_\mathcal{S}^\perp$ be the projection of c onto the orthogonal subspace, then we have that $c = c_\mathcal{S} + c_\mathcal{S}^\perp$ and $c_\mathcal{S} \perp c_\mathcal{S}^\perp$. When $\text{conv}(\mathcal{S})$ is full-dimensional, then $c = c_\mathcal{S}$, $d^* = c_\mathcal{S}^\perp = 0$ and the optimal value to the inverse problem is $\|c_\mathcal{S}\| = \|c\|$. When c is in the orthogonal subspace, then $c = c_\mathcal{S}^\perp$, $d^* = c_\mathcal{S}^\perp$ and the optimal value to the inverse problem is 0.

1.2 The Separation Problem

There is an obvious close relationship between the inverse problem (INV) and the separation problem for $\text{conv}(\mathcal{S})$. Given an $\hat{x} \in \mathbb{Q}^n$, the separation problem for $\mathcal{S}$ is to determine whether $\hat{x} \in \mathcal{S}$ and, if not, to generate a hyperplane separating $\hat{x}$ from $\text{conv}(\mathcal{S})$. Associated with each such hyperplane is a *valid inequality* violated by $\hat{x}$, defined as follows.

Definition 1. *A* valid inequality *for a set $\mathcal{Q}$ is a pair $(a, b) \in \mathbb{Q}^{n+1}$ such that $\mathcal{Q} \subseteq \{x \in \mathbb{R}^n \mid a^\top x \leq b\}$. The inequality is said to be* violated by $\hat{x} \in \mathbb{Q}^n$ *if $a^\top \hat{x} > b$.*

Generating a separating hyperplane is equivalent to determining the existence of $d^* \in \mathbb{Q}^n$ such that

$$d^{*\top} \hat{x} > d^{*\top} x \; \forall x \in \mathcal{S}.$$

In such a case, $(d^{*\top}, \max_{x \in \mathcal{S}} d^{*\top} x)$ is an inequality valid for $\text{conv}(\mathcal{S})$ that is violated by $\hat{x}$ (therefore proving that $\hat{x} \notin \text{conv}(\mathcal{S})$). On the other hand, as we describe in more detail the next section, d^* is feasible for (INV)

if and only if

$$d^{*\top} x \leq d^{*\top} x^{)} \quad \forall x \in \mathcal{S},$$

which similarly means that $(d^{*\top}, d^{*\top} x^0)$ is a valid inequality for $\text{conv}(\mathcal{S})$. In other words, a feasible solution to (INV) can also be viewed as an inequality valid for $\text{conv}(\mathcal{S})$ that is binding at x^0. Assuming that x^0 is an extreme point of $\text{conv}(\mathcal{S})$, the feasible set of (INV) corresponds exactly to inequalities describing the so-called *radial cone of* $\text{conv}(S)$ at x^0, a radial cone comprised of the facet-defining inequalities valid for $\text{conv}(\mathcal{S})$ that are binding at x^0.

1.3 Previous Work

There are a range of different flavors of inverse optimization problem. The inverse problem we investigate is to determine objective function coefficients that make a given solution optimal, but other flavors of inverse optimization include constructing a missing part of either the coefficient matrix or the right-hand side that makes a given solution optimal. Heuberger [2004] provides a detailed survey of inverse combinatorial optimization problems. In this paper, different types of inverse problems, including types for which the inverse problem seeks parameters other than objective function coefficients, are examined. A survey of solution procedures for specific combinatorial problems is provided, as well as a classification of the inverse problems that are common in the literature. According to this classification, the inverse problem we study in this paper is an *unconstrained*, *single feasible object*, and *unit weight norm* inverse problem. Our results can be straightforwardly extended to some related cases, such as multiple given solutions.

Cai et al. [1999] examine an inverse center location problem in which the aim is to construct part of the coefficient matrix, in this case the distances between nodes from a given optimal solution. It is shows that even though the center location problem is polynomially solvable, this particular inverse inverse problem is NP–hard. This is done by way of a polynomial transformation of the satisfiability problem to the decision version of the inverse center location problem. This analysis indicates that the problem of constructing part of the coefficient matrix is harder than the forward version of the problem.

Huang [2005] examines the inverse knapsack problem and inverse integer optimization problems. In this paper, a pseudo–polynomial algorithm for the inverse knapsack problem is presented. It is also shown that inverse integer optimization with a fixed number of constraints is pseudo–polynomial by transforming the inverse problem to a shortest path problem on a directed graph. When the number of constraints are fixed, this results a pseudo–polynomial algorithm for inverse integer optimization.

Schaefer [2009] studies general inverse integer optimization problems. Using super-additive duality, a polyhedral description of the set of all feasible objective functions is derived. This description has only continuous variables but an exponential number of constraints. A solution method using this polyhedral description is proposed. Finally, Wang [2009] suggests a cutting plane algorithm similar to the one suggested herein and presents computational results on several test problem with an implementation of this algorithm.

The case when the feasible set is an explicitly described polyhedron is well–studied by Ahuja and Orlin [2001]. In their study, they analyze the shortest path, assignment, minimum cut, and minimum cost flow problems under the ℓ_1 and ℓ_∞ norms in detail. They also conclude that inverse optimization problem is polynomially solvable when the forward problem is polynomially solvable. The present study aims to generalize the result of Ahuja and Orlin to the case when the forward problem is not necessarily polynomially solvable, as well as to make connections to other well-known problems.

In the remainder of the paper, we first introduce a cutting-plane algorithm for solving (INVMILP) and then address its computational complexity. As written, this is a semi-infinite program, but it is easy to see that we can replace the infinite set of constraints with a finite set corresponding to the extreme points of $\text{conv}(\mathcal{S})$. This still leaves us with what is ostensibly a non-linear objective function. We show in Section 2 that for the ℓ_∞ and ℓ_1 norms, this problem can be expressed as a standard *linear optimization problem* (LP), albeit

one with an exponential number of constraints. The reformulation can be readily solved in practice using a standard cutting plane approach. On the other hand, we show in Section 3 that the complexity does not depend on the norm.

2 A Cutting-plane Algorithm

In this section, we describe a basic cutting-plane algorithm for solving (INVMILP) under the ℓ_1 and ℓ_∞ norms. The algorithm is conceptual in nature and presented in order to illustrate the relationship of the inverse problem to both the forward problem and the separation problem. A practical implementation of this algorithm would require additional sophistication and the development of such an algorithm is the subject of current research.

The first step is to formulate (INVMILP) explicitly for ℓ_1 and ℓ_∞ norms as an LP using standard linearization techniques. The objective function of an inverse MILP under the ℓ_1 norm can be linearized by the introduction of variable vector y, and associated constraints as in (INVMILP-L1).

$$
\begin{aligned}
z_1^{-1} = \min \quad & \theta \\
\text{s.t.} \quad & \theta = \sum_{i=1}^{n} y_i && && \text{(INVMILP-L1a)} \\
& c_i - d_i \leq y_i && \forall i \in \{1, 2, \ldots, n\} && \text{(INVMILP-L1b)} \\
& d_i - c_i \leq y_i && \forall i \in \{1, 2, \ldots, n\} && \text{(INVMILP-L1c)} \\
& d^\top x \leq d^\top x^0 && \forall x \in \mathcal{S}. && \text{(INVMILP-L1d)}
\end{aligned}
$$

For ℓ_∞ norm case, variable θ and two sets of constraints are introduced to linearize the problem.

$$
\begin{aligned}
z_\infty^{-1} = \min \quad & \theta \\
\text{s.t.} \quad & c_i - d_i \leq \theta && \forall i \in \{1, 2, \ldots, n\} && \text{(INVMILP-INFa)} \\
& d_i - c_i \leq \theta && \forall i \in \{1, 2, \ldots, n\} && \text{(INVMILP-INFb)} \\
& d^\top x \leq d^\top x^0 && \forall x \in \mathcal{S}. && \text{(INVMILP-INFc)}
\end{aligned}
$$

Both (INVMILP-L1) and (INVMILP-INF) are continuous, semi-infinite optimization problems. To obtain a finite problem, one can replace the inequalities (INVMILP-L1d) and (INVMILP-INFc) with constraints (2) and (3) involving the finite set $\mathcal{E}$ of extreme points and $\mathcal{R}$ of rays of the convex hull of $\mathcal{S}$.

$$
\begin{aligned}
& d^\top x \leq d^\top x^0 && \forall x \in \mathcal{E} && (2) \\
& d^\top r \geq 0 && \forall r \in \mathcal{R}. && (3)
\end{aligned}
$$

Although constraints (2) and (3) yield a finite formulation, the cardinality of $\mathcal{E}$ and $\mathcal{R}$ may still be very large and generating them explicitly is likely to be very difficult in any case. It is thus not practical to write this formulation down explicitly via a priori enumeration and hand it to a solver. The proposed algorithm avoids explicitly enumerating the inequalities in the formulation by generating them dynamically using a standard cutting-plane approach. Wang [2009] has previously described a similar approach, but our purpose in describing it again here is to illustrate basic principles and to make the connection to a similar existing algorithm for solving the standard separation problem.

We describe the algorithm for the case of (INVMILP-INF), but note that the extension to (INVMILP-L1) is straightforward. We also assume $\mathcal{S}$ is bounded (so that $\text{conv}(\mathcal{S})$ has no extreme rays and $\mathcal{R} = \emptyset$) As previously observed, (INVMILP-INF) is an LP with an exponential class (INVMILP-INFc) of inequalities. Nevertheless, the result of Grötschel et al. [1993] tells us that (INVMILP-INF) can be solved efficiently using

a cutting-plane algorithm, provided we can solve the problem of separating a given point from the feasible region efficiently. The constraints (INVMILP-INFa) and (INVMILP-INFb) can be explicitly enumerated, so we focus on separation with respect to constraints (INVMILP-INFc), which means we are solving the separation problem for set $\mathcal{D}(x^0)$. For an arbitrary $\hat{d} \in \mathbb{R}^n$, this separation problem is to determine a hyperplane separating $\hat{d}$ from $\mathcal{D}(x^0)$ or to verify that $\hat{d} \in \mathcal{D}(x^0)$.

The question of whether $\hat{d} \in \mathcal{D}(x^0)$ is equivalent to asking whether $\hat{d}^\top x \leq \overline{d}^\top x^0$ for all $x \in \mathcal{S}$. This can be answered by determining $x^* \in \max_{x \in \mathcal{S}} \overline{d}^\top x$. When $d^\top x^* > d^\top x^0$, then x^* yields a new inequality valid for $\mathcal{D}(x^0)$ that is violated by $\overline{d}$. Otherwise, we have a proof that $\overline{d} \in \mathcal{D}(x^0)$. Hence, the separation problem for $\mathcal{D}(x^0)$ is equivalent to the forward optimization problem.

The proposed algorithm alternates between solving a master problem and the separation problem just described, as usual. The initial master problem is an LP obtained by relaxing (INVMILP-INFc) in (INVMILP-INF). We then attempt to separate the solution to this master LP from the set $\mathcal{D}(x^0)$ and either add the separating inequality or terminate, as appropriate. More formally, we have

$$\begin{aligned} \min \quad & y \\ \text{s.t.} \quad & c_i - d_i \leq y \qquad \forall i \in \{1, 2, \ldots, n\} \\ & d_i - c_i \leq y \qquad \forall i \in \{1, 2, \ldots, n\} \\ & d^\top x \leq d^\top x^0 \qquad\qquad \forall x \in \mathcal{E}^k \end{aligned} \tag{Inv$\mathcal{P}_k$}$$

$$x^k \in \operatorname{argmax}_{x \in \mathcal{S}} d^{k\top} x \tag{$\mathcal{P}_k$}$$

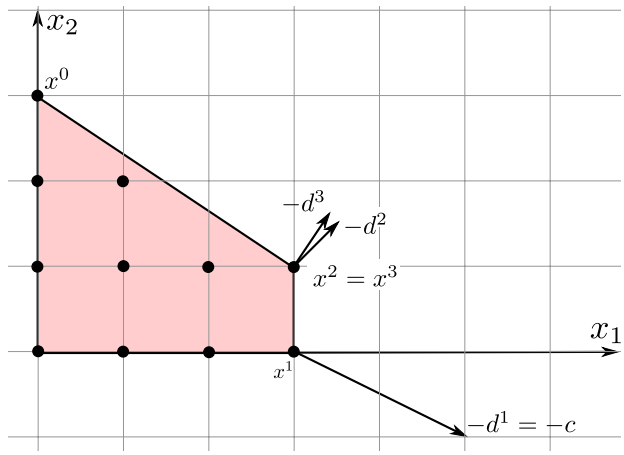
Here, $\mathcal{E}^k = \{x^1, \ldots, x^k\}$ are the points in $\mathcal{S}$ generated so far. (Inv$\mathcal{P}_k$) is the relaxation of (INVMILP-INF) along with valid inequalities corresponding to point in $\mathcal{E}^k$. When ($\mathcal{P}_k$) is unbounded, then x^0 is in the relative interior of $\operatorname{conv}(\mathcal{S})$ and $d = c_{\mathcal{S}}^{\perp}$ is an optimal solution, as mentioned earlier. The overall procedure is given in Algorithm 1.

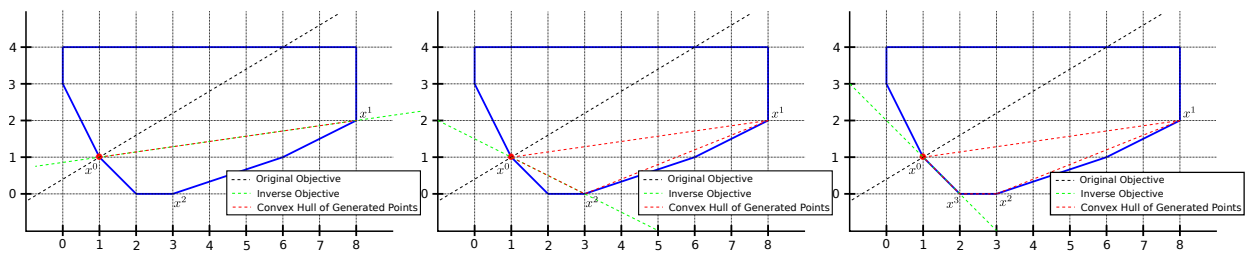
Algorithm 1 Cutting plane algorithm for (INVMILP-INF)

```
k ← 0, E^1 ← ∅.
do
    k ← k + 1.
    Solve (InvP_k), d^k ← d*.
    Solve (P_k).
    if (P_k) unbounded then
        y* ← ||c||_∞, d* ← 0, STOP.
    else
        x^k ← x*.
    end if
    E^{k+1} ← E^k ∪ {x^k}.
while d^k⊤(x^k − x^0) > 0
y* ← ||c − d^k||_∞, d* ← d^k, STOP.
```

To understand the nature of the algorithm, observe that in iteration k, the master problem is equivalent to the inverse problem with respect to the feasible set $\mathcal{E}^k \cup \{x^0\}$. Equivalently, we are replacing $\mathcal{D}(x^0)$ with the restricted set $\mathcal{D}^k(x^0) = \operatorname{cone}(\mathcal{E}^k \cup \{x^0\})$. We illustrate by considering a small example. Let $c = (2, -1)$, $x^0 = (0, 3)$ and $\mathcal{S}$ given as in Figure 2 where both x_1 and x_2 are integer and convex hull of $\mathcal{S}$ is given. k, d^k and x^k values through iterations are given in Table 1.

Figure 2: Feasible region and iterations of example problem





(a) Iteration 1 (b) Iteration 2 (c) Iteration 3

Figure 3: Pictorial illustration of Algorithm 1

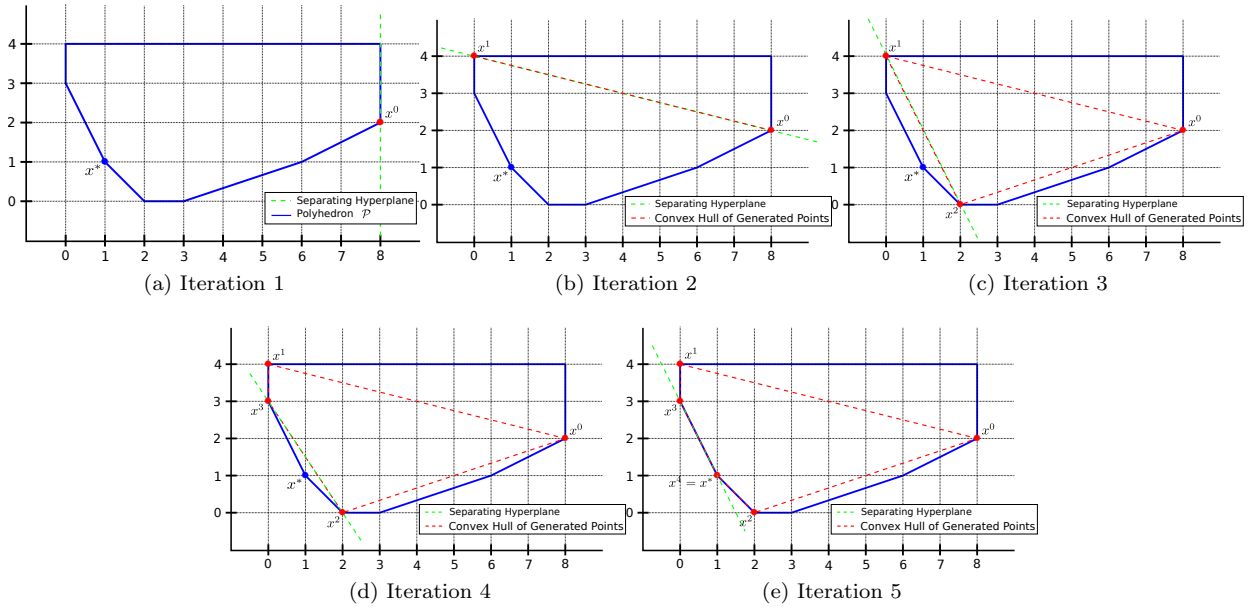
Table 1: k, d^k, x^k and $\mathcal{E}^k$ values through iterations

	k	$\mathcal{E}^k$	d^k	x^k	$\|c - d^k\|_\infty$
initialization	1	$\emptyset$	$(2, -1)$	$(3, 0)$	0
iteration 1	2	$\{(3, 0)\}$	$(0.5, 0.5)$	$(3, 1)$	1.5
iteration 2	3	$\{(3, 0), (3, 1)\}$	$(0.4, 0.6)$	$(3, 1)$	1.6

The (unique) optimal solution of this small example is $d^3 = (0.4, 0.6)$, and the optimal value is $y^* = \|c - d^3\|_\infty = 1.6$. Figure 3 provides a geometric visualization illustrating how the algorithm would proceed for an example where the set $\mathcal{S}$ is the collection of integer points inside the blue polyhedron.

The above interpretations again highlight the close relationship between the inverse and separation problems for $\mathcal{S}$. Another way of interpreting (Inv$\mathcal{P}_k$) is as an optimization problem over the set of inequalities that are both valid for conv $\left(\mathcal{E}^k \cup \{x^0\}\right)$ and binding at x^0. ($\mathcal{P}_k$) can then be interpreted as the problem of determining whether there is a member of set $\mathcal{S}$ that violates this generated inequality. The existence of such a point shows that the inequality is not valid for conv($\mathcal{S}$) and that d^k is hence not feasible for (INVMILP-INF).

Algorithm 1 can be easily modified to solve the generic separation problem for conv($\mathcal{S}$) by interpreting x^0 as the point to be separated and replacing the objective function (and associated auxiliary constraints) of (Inv$\mathcal{P}_k$) with one measuring the degree of violation of x^0. In this case, (Inv$\mathcal{P}_k$) can be interpreted as the problem of separating x^0 from conv($\mathcal{E}^k$). The dual of (Inv$\mathcal{P}_k$) can then be interpreted as the problem of



(a) Iteration 1 (b) Iteration 2 (c) Iteration 3

(d) Iteration 4 (e) Iteration 5

Figure 4: Pictorial illustration of algorithm for generating Fenchel cut

determining whether x^0 can be expressed as a convex combination of the members of $\mathcal{E}^k$ (the membership problem for $\text{conv}(\mathcal{E}^k)$). If not, the proof is a separating hyperplane, which is an inequality valid for $\text{conv}(\mathcal{E}^k)$. As in the inverse case, $(\mathcal{P}_k)$ is interpreted as the problem of determining whether there is an $x^k \in \mathcal{S}$ that is violated by the associated valid inequality. The generated valid inequalities are sometimes called *Fenchel cuts* [Boyd, 1994]. Figure 4 illustrates how the algorithm for generating Fenchel cuts might proceed for the same polyhedron as in Figure 3.

3 Computational Complexity

In this section, we briefly review the major concepts in complexity theory and the classes into which (the decision versions of) optimizations problems are generally placed, as well as give archetypal examples of problems that fall into these classes. The material here will be familiar to most readers with the possible exception of two concepts that are lesser known in the mathematical optimization literature, but are important in this paper. The first of these are the definitions of the complexity classes D^{P} and Δ_2^p, which play a role in our results below. The second of these is the distinction between the polynomial Turing reductions used by Cook [1971] in his seminal work and the polynomial many-to-one reductions used by Karp [1972].

In what follows, we mainly follow the framework laid out in Garey and Johnson [1979], whose notion of NP-completeness is based on Karp reduction. We omit many details and refer the reader to either their book or the sweeping introduction to complexity given by Arora and Barak [2007] for more details. Although these and many other excellent treatments of various concepts in complexity theory are readily accessible, we nevertheless briefly review the concepts here in order to make the presentation coherent and self-contained. We also do so to emphasize our slightly non-traditional point of view and to bring out some of the details that are normally not emphasized, but are important here.

The fundamentals of complexity theory and NP-completeness developed by Cook [1971], Karp [1972], Edmonds [1971], and others provide a rigorous framework within which problems arising in discrete optimization

can be analyzed. The theory derived from the earlier work on the Entscheidungsproblem by Turing [1937] and perhaps for that reason, it was originally developed to analyze decision problems, e.g., problems where the output is YES or NO. Although there exists a theory of complexity that applies directly to optimization problems (see [Krentel, 1988, Vollmer and Wagner, 1995, Krentel, 1987a]), most analyses are done by converting the optimization problem to an equivalent decision problem form.

The decision problem form typically used for most discrete optimization problems is that of determining whether a given upper bound is valid. For most problems of current practical interest, this decision problem is in either the class P or the class NP. Notable exceptions are the bilevel (and other multilevel) optimization problems, whose decision versions are in higher levels of the so-called *polynomial-time hierarchy* Stockmeyer [1976a].

3.1 Complexity Classes

Definitions. In the framework of Garey and Johnson [1979], an algorithm is a procedure implemented using the well-known logic of a Turing machine, a simple model of a computer that implements and runs a single program. The input to the algorithm is a string in a given *alphabet*, which is simply $\{0, 1\}$ on all modern computing devices. Thus, the set of all possible input strings is $\{0, 1\}^*$. A *problem* (or *problem class*) is the subset L of $\{0, 1\}^*$, also called a *language*, for which the output is YES. A problem *instance* is one particular member of L. An algorithm *solves* a problem specified by language $L \subseteq \{0, 1\}^*$ if it correctly outputs YES if and only if the input string is in L. In this case, we say the associated Turing machine *recognizes* the language L.

The *running time* of an algorithm for a given problem is the worst-case number of steps/operations required by the associated Turing machine across all instances of that problem. This worst case is usually expressed as a function of the "size" of the input, since the worst case would otherwise be unbounded for any class with arbitrarily large instances. The size of the input is formally defined to be its *encoding length*, which is the length of the string representing the input in the given alphabet. Since we take the alphabet to be $\{0, 1\}$, the encoding length of an integer n is

$$\langle n \rangle = 1 + \lceil \log_2(|n| + 1) \rceil.$$

Further, the encoding length of a rational number $r = p/q$ is $\langle r \rangle = \langle p \rangle + \langle q \rangle$ (encoding lengths play an important role in the complexity proofs of Section 4 below; see Grötschel et al. [1993] for detailed coverage of definitions and concepts). The *computational complexity* of a given problem is the running time (function) of the "best" known algorithm (though there isn't always a unique ordering, as not all running-time functions are comparable).

Reduction is the means by which an algorithm for one class of problems (specified by, say, language L_1) can be used as a subroutine within an algorithm for another class of problems (specified by, say, language L_2). It is also the means by which problem equivalence and complexity classes are defined.

There are two distinct notions of reduction and the difference between them is important in what follows. The notion that is most relevant in the theory of NP-completeness is the *polynomial many-to-one reduction* introduced by Karp [1972]. There is a Karp reduction from a problem specified by language L_2 to a problem specified by a language L_1 if there exists a mapping $f : \{0, 1\}^* \to \{0, 1\}^*$ such that

- $f(x)$ is computable in time polynomial in $\langle x \rangle$ and
- $x \in L_2$ if and only if $f(x) \in L_1$.

Thus, if we have an algorithm (Turing machine) for recognizing the language L_1 and such a mapping f, we implicitly have an algorithm for recognizing L_2. In this case, we say there is a Karp reduction from L_2 to L_1.

A second notion of reduction is the polynomial Turing reduction introduced by Cook [1971] in his seminal work. This type of reduction is defined in terms of *oracles*. An oracle is a conceptual subroutine that can solve a given problem or class of problems in constant time. Roughly speaking, the *oracle complexity* of a problem is its complexity *given the theoretical existence of a certain oracle*. There is a Cook reduction from a problem specified by language L_2 to a problem specified by language L_1 if there is a polynomial-time algorithm for solving L_2 that utilizes an oracle for L_1. Hence, the only requirement is that the number of calls to the oracle must be bounded by a polynomial. The difference between Karp reduction and Cook reduction is that Karp reduction can be thought of as allowing only a single call to the oracle as the last step of the algorithm, whereas Cook reduction allows a polynomial number of calls to the oracle. There are a range of other notions of reduction that utilize other different bounds on the number of oracle calls [Krentel, 1987b].

Decision problems specified by languages L_1 and L_2 are said to be *polynomially equivalent* if there is a reduction in both directions—L_1 reduces to L_2 and L_2 reduces to L_1. Equivalence can be defined using either the Karp or Cook notions of reduction. It is conjectured (though not known, see Beigel and Fortnow [2003]) that these notions of equivalence are distinct and yield different equivalence classes of problems.

A problem in a complexity class is said to be *complete* for the class if every other problem in the class can be reduced to it. Informally, this means that the complete problems are at least as difficult to solve as any other problem in the class (in a worst-case sense). Completeness of a given problem in a given class can be shown by providing a reduction from an already known complete problem for the given class. Polynomial equivalence, as described above, is an equivalence relation in the mathematical sense and can thus be used to define equivalence classes for problems. The complete problems for a class are exactly those in the largest such equivalence class that is contained in the class. Note that this means that the set of complete problems is different for different notions of equivalence (Karp versus Cook). The theory of NP-completeness is defined in terms of Karp reduction.

Finally, we have the concept of a *certificate*. A certificate is a string that, when concatenated with the original input string, forms a (longer) input string to an associated decision problem (which we informally call the *verification problem*) that yields the same output as the original one (but presumably can be solved more efficiently). A certificate can be viewed as a proof of the result of a computation. When produced by an algorithm for solving the original problem, the certificate serves to certify the result of that computation after the fact. The efficiency with which such proofs can be checked is another property of classes of problems (like the running time) that can be used to partition problems into classes according to difficulty. We discuss more about the use of certificates and their formal definition in particular contexts below.

Class P. The most well-known complexity class is P, the class of decision problems that can be solved in polynomial time on a deterministic Turing machine[Stockmeyer, 1976a]. Alternatively, the class P can be defined as the *smallest* equivalence class of problems according to the polynomial equivalence relation described earlier. Note that for problems in P, there is no distinction between equivalence according to Karp and Cook. The decision version of LP, minimum cost network flow problems and related variants, as well as the problem of determining whether a given system of liner inequalities has a solution, are all in this class.

Class NP. NP is the class of problems that can be solved in polynomial time with a non-deterministic Turing machine. Informally, a non-deterministic Turing machine is a Turing machine with an infinite number of parallel processors. With such a machine, a search algorithm, for example, may be efficiently implemented by following all possible search paths simultaneously, even if there are exponentially many of them and most are dead ends. Alternatively, NP can be defined as the class of decision problems for which there exists a certificate that can be verified in time polynomial in the length of the input when the output is YES. In fact, these two informal definitions can be formalized and shown to be equivalent. Intuitively, the idea is that the certificate can be taken to be an encoding of an execution path that actually leads to a program state that

proves the output is YES (discarding all the irrelevant dead ends).

More formally, if $L \in \mathsf{NP}$, then there exists $L^C \in \mathsf{P}$ such that

$$x \in L \Leftrightarrow \exists y \in \{0,1\}^* \text{ such that } (x,y) \in L^C \text{ and } \langle y \rangle \text{ is polynomial in } \langle x \rangle.$$

In this case, y is the certificate. Because such a certificate has an encoding length polynomial in the encoding length of x and can be verified in time polynomial in the encoding length of x, such certificate are often said to be *short* and NP is said to be the class of decision problems having a short certificate. We discuss classes not having short certificates below.

The lower bound verification[1] problem for (MILP) (usually referred to in the literature as *the* decision version of MILP) is a prototypical problem in this class and is defined as follows.

Definition 2. ***MILP Lower Bound Verification Problem (MLBVP)***

- *INPUT:* $\gamma \in \mathbb{Q}$, $d \in \mathbb{Q}^n$, $A \in \mathbb{Q}^{m \times n}$, $b \in \mathbb{Q}^m$, *and* $r \in \mathbb{N}$, *where* (A, b, r) *is an encoding of the set* $\mathcal{S}$ *in* (MILP) *and* $(d, \mathcal{S})$ *is the input to* (MILP).
- *OUTPUT: YES, if there exists* $x \in \mathcal{S}$ *such that* $d^\top x \geq \gamma$, *NO otherwise.*

The MLBVP is in NP since there always exists a vector x whose existence proves the YES answer specified in Definition 2 that is itself a polynomially verifiable certificate. In particular, when the answer is YES, there is always an extreme point of $\text{conv}(\mathcal{S})$ that can serve as such a certificate and has encoding length polynomially bounded by the encoding length of the problem input. The set of all problems that are complete for NP form NP-complete, the largest equivalence class of problems in NP according to the polynomial equivalence relation of Karp described earlier (using Cook reduction yields a different set of complete problems, assuming that $\mathsf{P} \neq \mathsf{NP}$). The first problem shown to be complete for class NP was the satisfiability (SAT) problem [Cook, 1971]. It was proved to be complete by providing a Karp reduction of any problem that can be solved by a non-deterministic Turing machine to the SAT problem. The MLBVP is complete for NP because SAT can be Karp-reduced to it. It is well-known that the question of whether $\mathsf{P} = \mathsf{NP}$ is currently unresolved, though it is widely believed that they are distinct classes.

Class coNP. Informally, coNP is the class of decision problems for which there exists a certificate that can be verified in time polynomial in the encoding length of the input when the output is *NO*. The formal definition parallels the definition for NP so we do not repeat it here. In the context of optimization, it is convenient to think of NP as problems requiring to show that some element of a given set *has* a certain polynomially verifiable property ($\exists$ an element of the set with the given property), whereas coNP contains problems in which *all* elements of the set has the property (property holds $\forall$ elements of the set).

The MILP Upper Bound Verification Problem is an example of a prototypical problem in coNP.

Definition 3. ***MILP Upper Bound Verification Problem (MUBVP)***

- *INPUT:* $\gamma \in \mathbb{Q}$, $d \in \mathbb{Q}^n$, $A \in \mathbb{Q}^{m \times n}$, $b \in \mathbb{Q}^m$, *and* $r \in \mathbb{N}$, *where* (A, b, r) *is an encoding of the set* $\mathcal{S}$ *in* (MILP) *and* $(d, \mathcal{S})$ *is the input to* (MILP).
- *OUTPUT: YES, if* $\max_{x \in \mathcal{S}} \leq \gamma$, *NO otherwise.*

The input to the MUBVP is (γ, d, A, b, r), as in the case of the MLBVP. The MUBVP is in coNP because when the output is NO, a feasible solution in $\mathcal{S}$ with an objective value strictly greater than γ is a short certificate that is verifiable in polynomial time, as described above.

[1] The term "verification" is used here in a slightly different way than it is used in the context of certificates, although the uses are related and the meaning can be generalized to include both uses.

Class D^{P}. While NP and coNP are both well-known classes, the class D^{P} introduced by Papadimitriou and Yannakakis [1982] is lesser-known, though very useful nevertheless. It is the class of problems associated with languages that are intersections of a language from NP and a language from coNP. A prototypical problem complete for D^{P} is MOVVP, the MILP Optimal Value Verification Problem, defined as follows.

Definition 4. ***MILP Optimal Value Verification Problem (MOVVP)***

- *INPUT:* $\gamma \in \mathbb{Q}$, $d \in \mathbb{Q}^n$, $A \in \mathbb{Q}^{m\times n}$, $b \in \mathbb{Q}^m$, *and* $r \in \mathbb{N}$, *where* (A, b, r) *is an enconding of the set* $\mathcal{S}$ *in* (MILP) *and* $(d, \mathcal{S})$ *is the input to* (MILP).
- *OUTPUT: YES, if* $\max_{x\in\mathcal{S}} d^\top x := \gamma$, *NO otherwise.*

It is easy to see that language of MOVVP is intersection of languages of the MLBVP and the MUBVP. The output of MOVVP is YES if and only if output of both the MLBVP and the MUBVP are YES, i.e., γ is both an upper and a lower bound for (MILP).

An idea that revealed to us while doing this work and that we would like to propose here for more general adoption is that the MOVVP is a more natural decision problem to associate with discrete optimization problems than the more traditional MLBVP. Most algorithms for discrete optimization are based on iterative construction of separate certificates for the validity of the upper and lower bound, which must be equal to certify optimality. Given certificates for the upper and lower bound verification problems, a certificate for the optimal value verification problem can thus be constructed directly. The class D^{P} thus contains the optimal value verification problem associated with (MILP), whereas it is the associated MLBVP that is contained in the class NP-complete. We find this is somewhat unsatisfying, since the original problem (MILP) is only Cook reducible to the MLBVP.

The Polynomial Hierarchy. Further classes in the so-called *polynomial-time hierarchy* (PH), described in the seminal work of Stockmeyer [1976a] are defined recursively using oracle computation. The notation A^{B} is used to denote the class of problems that would be in A, assuming the existence of an oracle for problems in class B.

Using this concept, Δ_2^p is the class of decision problems that can be solved in polynomial time given an NP oracle, i.e., the class P^{NP}. This class is a second-level member of the PH. Further levels are defined according to the following recursion.

$$\begin{aligned}
\Delta_0^p &:= \Sigma_0^p := \Pi_0^p := \mathsf{P},\\
\Delta_{k+1}^p &:= \mathsf{P}^{\Sigma_k^p},\\
\Sigma_{k+1}^p &:= \mathsf{NP}^{\Sigma_k^p}, \text{ and}\\
\Pi_{k+1}^p &:= \mathsf{coNP}^{\Sigma_k^p}.
\end{aligned}$$

PH is the union of all levels of the hierarchy. There is also an equivalent definition that uses the notion of certificates. Roughly speaking, each level of the hierarchy consists of problems with certificates of polynomial size, but whose verification problem is in the class one level lower in the hierarchy. In other words, the problem of verifying a certificate for a problem in Σ_i^p is a problem in the class Σ_{i-1}^p [Stockmeyer, 1976b].

Figure 5 illustrates class Δ_2^p relative to $\mathcal{D}^{\mathsf{P}}$, NP, coNP and P, assuming $\mathsf{P} \neq \mathsf{NP}$. If $\mathsf{P} = \mathsf{NP}$, we conclude that all classes are equivalent, i.e., $\Delta_2^p = \mathsf{D}^{\mathsf{P}} = \mathsf{NP} = \mathsf{coNP} = \mathsf{P}$. This theoretical possibility is known as the collapse of PH to its first level [Papadimitriou, 2003] and is thought to be highly unlikely. A prototypical problem complete for Δ_2^p is the problem of showing that a given solution to an MILP is *unique* [Papadimitriou, 2003].

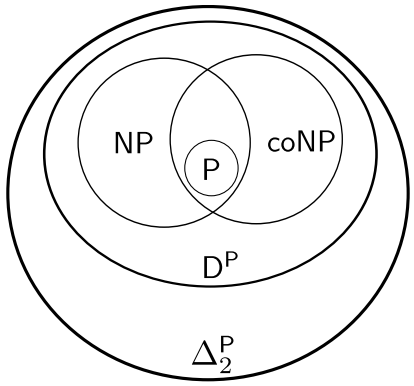


Figure 5: Complexity classes Δ_2^p, $\mathcal{D}^{\mathsf{P}}$, NP, coNP and P, assuming $\mathsf{P} \neq \mathsf{NP}$

3.2 Optimization and Separation

The concepts of reduction and polynomial equivalence can be extended to problems other than decision problems, but this requires some additional care and machinery. Decision problems can be (and often are) reduced to optimization problems, in a fashion similar to that described earlier, in an attempt to classify them. When a problem that is complete for a given class can be Karp reduced to an optimization problem, we refer to the optimization problem as *hard* for the class. When a decision problem that is complete for NP can be Karp-reduced to an optimization problem, for example, the optimization problem is classified as NP-hard. This *does not*, however, require that the decision version of this optimization problem is in NP. In reality, it may be on some other level of the hierarchy. Hardness results can therefore be somewhat misleading in some cases.

In their foundational work, Grötschel et al. [1993] develop a detailed theoretical basis for the claim that the separation problem for an implicitly defined polyhedron is polynomially equivalent to the optimization problem over that same polyhedron. This is done very carefully, beginning from certain decision problems and proceeding to show their equivalence to related optimization problems. The notion of reduction used, however, is Cook reduction. We use Karp reduction to show that that the inverse and forward optimization problems are in the same complexity class.

Using an optimization oracle to solve the separation problem (and vice versa) involves converting between different representations of a given polyhedron. Particularly, we consider (implicit) descriptions in terms of both valid inequalities (so-called H-representations) and in terms of vertices and extreme rays (so-called V-representations). There is a duality relating these two forms of polyhedral representation that is at the heart of the equivalence of optimization and separation. It is this very same duality that is also at the the heart of the equivalence between optimization problems and their inverse version, as we attempt to demonstrate in the next section.

The framework laid out by Grötschel et al. [1993] emphasizes that the efficiency with which the various representations can be manipulated algorithmically depends inherently and crucially on, among other things, the encoding length of the elements of these representations. For the purposes of their analysis, Grötschel et al. [1993] defined the notions of the *vertex complexity* and *facet complexity* of a polyhedron, which we repeat here, due to their relevance in the remainder of the paper.

Definition 5. *(Grötschel et al. [1993]).*

(*i*) *A polyhedron $\mathcal{P} \subseteq \mathbb{R}^n$ has* facet complexity *of at most φ if there exists a rational system of inequalities describing the polyhedron in which the encoding length of each inequality is at most φ (this is an H-representation).*

(*ii*) *Similarly, the* vertex complexity *of* $\mathcal{P}$ *is at most* ν *if there exist finite sets* V, E *such that* $\mathcal{P} = \text{conv}(V) + \text{cone}(E)$ *(this is a V-representation) and such that each of the vectors in* V *and* E *has encoding length at most* ν*.*

It is important to point out that these definitions are not given in terms of the encoding length of a full description of $\mathcal{P}$ because $\mathcal{P}$ may be an implicitly defined polyhedron whose description is never fully constructed. What *is* explicitly constructed are the components of the description (extreme points and facet-defining inequalities). The importance of the facet complexity and vertex complexity in the analysis is primarily that they provide bounds on the norms of *these* vectors. The ability to derive such bounds is a crucial element in the overall framework they present. The following are relevant results from Grötschel et al. [1993]

Proposition 1. *(Grötschel et al. [1993])*

(*i*) *(1.3.3) For any* $r \in \mathbb{Q}$, $2^{-\langle r \rangle + 2} \leq |r| \leq 2^{\langle r \rangle - 1} - 1$.

(*ii*) *(1.3.3) For any* $x \in \mathbb{Q}^n$, $\|x\|_p < 2^{\langle x \rangle - n}$ *for* $p \geq 1$.

(*iii*) *(6.2.9) If* $\mathcal{P}$ *is a polyhedron with vertex complexity at most* ν *and* $(a, b) \in \mathbb{Z}^{n+1}$ *is such that*

$$a^\top x \leq b + 2^{-\nu - 1}$$

for all $x \in \mathcal{P}$*, then* (a, b) *is also valid for* $\mathcal{P}$*.*

In other words, the facet complexity and the encoding lengths of the vectors involved specify a "granularity" that can allow us to, for example, replace a "$<$" with a "$\leq$" if we can bound the encoding length of the numbers involved.

Utilizing the above definitions, the result showing the equivalence of optimization and separation can be formally stated as follows.

Theorem 1. *(Grötschel et al. [1993]) Let* $\mathcal{P} \subseteq \mathbb{R}^n$ *be a polyhedron with facet-complexity* φ*. Given an oracle for any one of*

- *the upper-bound verification problem over* $\mathcal{P}$ *with linear objective* $c \in \mathbb{Q}^n$*,*
- *the separation problem for* $\mathcal{P}$ *with respect to* $\hat{x} \in \mathbb{Q}^n$*, or*
- *the lower-bound verification problem for* $\mathcal{P}$ *with linear objective* $c \in \mathbb{Q}^n$*,*

there exists an oracle polynomial-time algorithm for solving either of the other two problems. Further, all three problems are solvable in time polynomial in n*,* φ*, and either* $\langle c \rangle$ *(in the case of optimization or violation) or* $\langle \hat{x} \rangle$ *(in the case of separation).*

The problem of verifying a given upper bound was called the *violation problem* in Grötschel et al. [1993]. The above result refers only to the facet complexity φ, but we could also replace it with the vertex complexity ν, since it is easy to show that $\nu \leq 4n^2\varphi$. In the remainder of the paper, we refer to the "polyhedral complexity" whenever the facet complexity and vertex complexity can be used interchangeably.

4 Complexity of Inverse MILP

In this section, we apply the framework discussed in Section 3 to analyze the complexity of the inverse MILP. We follow the traditional approach and describe the complexity of the decision versions. In addition to the standard upper bound verification problem, we also consider the lower bound and optimal value verification problems. We show that the upper bound, lower bound and optimal value verification problems for the inverse MILP are in the complexity classes coNP–complete, NP–complete, and D^{P}–complete, respectively.

4.1 Polynomially Solvable Cases

The result of Ahuja and Orlin [2001] can be applied directly to observe that there are cases of the inverse MILP that are polynomially solvable. In particular, Ahuja and Orlin [2001] showed that the inverse problem can be solved in polynomial time whenever the forward problem is polynomially solvable.

Theorem 2. *(Ahuja and Orlin [2001]) If an optimization problems is polynomially solvable for each linear cost function, then the corresponding inverse problems under ℓ_1 and ℓ_∞ norms are polynomially solvable.*

Ahuja and Orlin [2001] use Theorem 1 of Grötschel et al. [1993] to conclude that inverse LP, in particular, is polynomially solvable. The separation problem in this case is an LP of polynomial size and is hence polynomially solvable. The theorem of Grötschel et al. [1993] is applicable, since Karp and Cook reductions are equivalent for problems in P. Theorem 2 also indicates that if a given MILP is polynomially solvable, then the associated inverse problem is also polynomially solvable.

4.2 General Case

In the general case, the MILP constituting the forward problem is not polynomially solvable and so we now consider MILPs whose decision versions are complete for NP. Applying the results of Grötschel et al. [1993] straightforwardly, as Ahuja and Orlin [2001] did, we can easily show that (INVMILP-L1) and (INVMILP-INF) can be solved in polynomial time, given an oracle for the MLBVP, as stated in the following theorem.

Theorem 3. *Given an oracle for the MLBVP,* (INVMILP-L1) *and* (INVMILP-INF) *are solvable in time polynomial in n, the vertex complexity of $\mathcal{S}$, and $\langle c \rangle$.*

The above result directly implies that IMILP under ℓ_1 and ℓ_∞ norms is in fact in the complexity Δ_2^{P}, but stronger results are possible, as we show. In the remainder of this section, we assume the norm used in a p-norm, as this is needed for some results (in particular, Proposition 1 is crucially applied).

Definitions. We next define decision versions of the inverse MILP analogous to those we defined in the case of MILP. These similarly attempt to verify that a given bound on the objective value is a lower bound, an upper bound, or an exact optimal value. The upper-bound verification problem for inverse MILP is as follows.

Definition 6. ***Inverse MILP Upper Bound Verification Problem (IMUBVP):***

- *INPUT: $\gamma \in \mathbb{Q}$, $d \in \mathbb{Q}^n$, $x^0 \in \mathbb{Q}^n$, $A \in \mathbb{Q}^{m \times n}$, $c \in \mathbb{Q}^n$, $b \in \mathbb{Q}^m$, and $r \in \mathbb{N}$, where (A, b, r) is an enconding of the set $\mathcal{S}$ in* (MILP) *and $(c, \mathcal{S}, x^0)$ are input data for problem* (INVMILP).
- *OUTPUT: YES, if $\exists d \in \mathcal{D}(x^0)$ such that $\|c - d\| \leq \gamma$, i.e., $\mathcal{K}(\gamma) \cap \mathcal{D}(x^0) \neq \emptyset$, NO otherwise.*

Similarly, we have the lower bound verification problem for inverse MILP.

Definition 7. ***Inverse MILP Lower Bound Verification Problem (IMLBVP):***

- *INPUT: $\gamma \in \mathbb{Q}$, $d \in \mathbb{Q}^n$, $x^0 \in \mathbb{Q}^n$, $A \in \mathbb{Q}^{m \times n}$, $c \in \mathbb{Q}^n$, $b \in \mathbb{Q}^m$, and $r \in \mathbb{N}$, where (A, b, r) is an enconding of the set $\mathcal{S}$ in* (MILP) *and $(c, \mathcal{S}, x^0)$ are input data for problem* (INVMILP).
- *OUTPUT: YES, if $\gamma \leq \|c - d\|$ for all $d \in \mathcal{D}(x^0)$, i.e., $\operatorname{int}(\mathcal{K}(\gamma)) \cap \mathcal{D}(x^0) = \emptyset$, NO otherwise.*

Finally, we have the optimal value verification problem.

Definition 8. ***Inverse MILP Optimal Value Verification Problem (IMOVVP):***

- *INPUT:* $\gamma \in \mathbb{Q}$, $d \in \mathbb{Q}^n$, $x^0 \in \mathbb{Q}^n$, $A \in \mathbb{Q}^{m\times n}$, $c \in \mathbb{Q}^n$, $b \in \mathbb{Q}^m$, *and* $r \in \mathbb{N}$, *where* (A, b, r) *is an enconding of the set* $\mathcal{S}$ *in* (MILP) *and* $(c, \mathcal{S}, x^0)$ *are input data for problem* (INVMILP).
- *OUTPUT: YES, if* $\min_{d\in\mathcal{K}(y)\cap\mathcal{D}(x^0)} y = \gamma$, *NO otherwise.*

In the rest of the section, we formally establish the complexity class membership of each of the above problems and in so doing, illustrate the relationships of the above problem to each other and to their MILP counterparts.

Informal Discussion. Before presenting the formal proofs, which are somewhat technical, we informally describe the relationship of the three problems above to each other and to their MILP analogues. Suppose we are given a value γ and we wish to determine whether it is an upper bound, a lower bound, or the exact optimal value of (MILP) with objective function vector $c \in \mathbb{Q}^n$. For simplicity, let us assume that $\text{conv}(\mathcal{S})$ is full-dimensional and that $\|c\| = 1$. Roughly speaking, we can utilize an algorithm for solving (INVMILP) to make the determination, as follows. We first construct a target vector

$$x^0 = \gamma c$$

which has an objective function value of γ by construction. Now suppose we solve (INVMILP) with this x^0 as the target vector. Note that $x^0 \notin \mathcal{S}$ in general. Solving this inverse problem will yield one of two results.

1. If $c^\top x \leq c^\top x^0 = \gamma$ for all $x \in \mathcal{S}$, then c is a feasible objective and optimal value will be zero.
2. If $\gamma < \max_{x\in\mathcal{S}} c^\top x$, then 0 is the only feasible objective and the optimal value is $\|c\| = 1$.

This does not precisely establish the status of γ as a bound because it cannot distinguish between when γ is a strict upper bound (and hence not a lower bound) and when γ is the exact optimal value. However, this can be overcome by appealing to Proposition 1 to reformulate the strict "$>$" to a "$\geq$" using an appropriately selected perturbation.

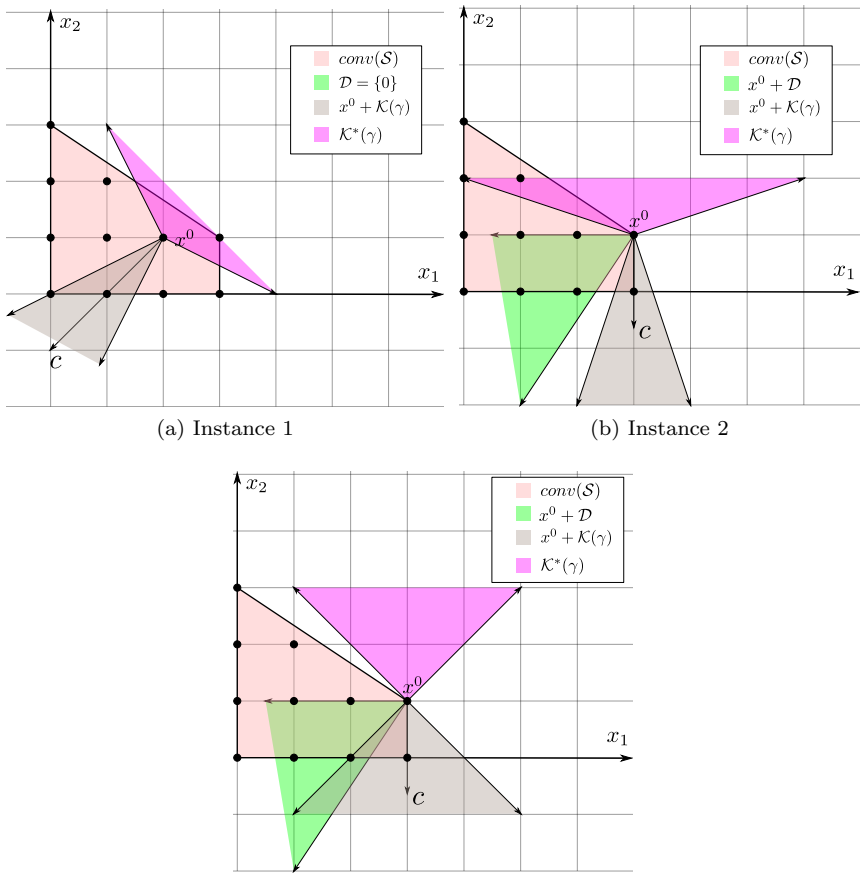
Required Lemmas. In the proofs that follow, the conic set consisting of points, not necessarily in $\mathcal{S}$, that have an objective function value greater than or equal to that of x^0 for *all* the vectors in $\mathcal{K}(\gamma)$ will play an important role. This set is formally defined as follows.

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

Another way of describing $\mathcal{K}^*(\gamma)$ is as the radial cone obtained by translating the dual of $\mathcal{K}(\gamma)$ from the origin to x^0. Intuitively, $\mathcal{K}^*(\gamma)$ and $\mathcal{S}$ can be thought of as being in the "primal space," i.e., the space of primal solution vectors, whereas the cones $\mathcal{D}(x^0)$ and $\mathcal{K}(\gamma)$ can be thought of as being in the "dual space", the space of directions. Figure 6 show how the various cones and sets introduced so far are related.

Figure 6 displays sets $\text{conv}(\mathcal{S})$, $\mathcal{D}(x^0)$, $\mathcal{K}(\gamma)$ and $\mathcal{K}^*(\gamma)$ for three different two-dimensional inverse problems with Euclidean norm. The output to the IMUBVP for these three instances are NO, NO, and YES respectively. $\mathcal{K}(\gamma) \cap \mathcal{D}(x^0)$ is empty for the first two instances, and nonempty for the third instance. $\text{conv}(\mathcal{S}) \cap \text{int}\,(\mathcal{K}^*(\gamma))$ is nonempty for the first two instances and empty for the third instance.

In the proofs that follow, we will use the following lemma to characterize precisely when γ is a lower bound for (INVMILP).



(a) Instance 1

(b) Instance 2

(c) Instance 3

Figure 6: Pictorial illustration of sets $\text{conv}(\mathcal{S})$, $\mathcal{D}(x^0)$, $\mathcal{K}(\gamma)$, $\mathcal{K}^*(\gamma)$ for 3 IMILP instances

Lemma 1. $\mathcal{K}(\gamma) \cap \mathcal{D}(x^0) = \emptyset$ *if and only if* $\text{conv}(\mathcal{S}) \cap \text{int}\,(\mathcal{K}^*(\gamma)) \neq \emptyset$.

Proof. ($\Rightarrow$) For the sake of contradiction, let us assume that both $\mathcal{K}(\gamma) \cap \mathcal{D}(x^0) = \emptyset$ and $\text{conv}(\mathcal{S}) \cap \text{int}\,(\mathcal{K}^*(\gamma)) = \emptyset$. Since $\text{conv}(\mathcal{S})$ and $\text{int}\,(\mathcal{K}^*(\gamma))$ are both convex sets, there exists a hyperplane separating them. In particular, there exists $a \in \mathbb{R}^n$ such that

$$\max_{x \in \text{conv}(\mathcal{S})} a^\top x \leq \min_{x \in K^*(\gamma)} a^\top x = \inf_{x \in \text{int}(K^*(\gamma))} a^\top x. \tag{SEPi}$$

The problem on the right-hand side is unbounded when $a \notin \mathcal{K}(\gamma)$, since then there must exist $x \in \text{int}\,(\mathcal{K}^*(\gamma))$ with $a^\top x > a^\top x^0$, which means that $x - x^0$ is a ray with positive objective value (recall $\mathcal{K}^*(\gamma)$ is a cone). Therefore, we must have $a \in \mathcal{K}(\gamma)$ and it follows that x^0 is an optimal solution for the problem on the right-hand side (one could also argue directly that whenever a is a hyperplane separating a convex set from a convex pointed cones, then the extreme points of the pointed cone must always be optimal when optimizing with objective function a). Therefore, we have

$$\max_{x \in \text{conv}(\mathcal{S})} a^\top x \leq a^\top x^0.$$

Since $a \in \mathcal{K}(\gamma)$, then by assumption, $a \notin \mathcal{D}(x^0)$, so there exists an $\hat{x} \in \mathcal{S}$ such that $a^\top(x^0 - \hat{x}) < 0$. So

finally, we have

$$a^\top x^0 < a^\top \hat{x} \leq \max_{x \in \text{conv}(\mathcal{S})} a^\top x \leq a^\top x^0,$$

which is a contradiction. This completes the proof of the forward direction.

($\Rightarrow$) For the reverse direction, we assume there exists $\overline{x} \in \text{conv}(\mathcal{S})$ and $\text{int}\,(\mathcal{K}^*(\gamma))$. Since $\overline{x} \in \text{conv}(\mathcal{S})$, there exists $\{x^1, x^2, \ldots, x^k\} \subseteq \mathcal{S}$ and $\lambda \in \mathbb{Q}_+^k$ such that $\overline{x} = \sum_{i=1}^k \lambda_i x^i$, $\sum_{i=1}^k \lambda_i = 1$, and $k \leq n+1$. Now, let an arbitrary $d \in \mathcal{K}(\gamma)$ be given. Since $\overline{x} \in \text{int}\,(\mathcal{K}^*(\gamma))$, we have that

$$\begin{aligned}
d^\top \left(x^0 - \overline{x}\right) < 0 &\Leftrightarrow d^\top x^0 - \left(\sum_{i=1}^k \lambda_i x^i\right) < 0 \\
&\Leftrightarrow d^\top \left(\sum_{i=1}^k \lambda_i x^0 - \sum_{i=1}^k \lambda_i x^i\right) < 0 \\
&\Leftrightarrow \sum_{i=1}^k \lambda_i d^\top \left(x^0 - x^i\right) < 0 \\
&\Rightarrow \exists j \in \{1, \ldots, k\} \text{ such that } d^\top (x^0 - x^j) < 0 \\
&\Rightarrow d \notin \mathcal{D}(x^0)
\end{aligned}$$

Since d was chosen arbitrarily, we have that $\mathcal{K}(\gamma) \cap \mathcal{D}(x^0) = \emptyset$. This completes the proof of the reverse direction.

□

In the same vein, we can also easily show the following result characterizing when γ is *not an upper bound*, using a similar proof, which we omit here because it is almost identical to the proof of the above.

Lemma 2. *For $\gamma \in \mathbb{Q}$, we have that* $\text{int}\,(\mathcal{K}(\gamma)) \cap \mathcal{D}(x^0) = \emptyset$ *if and only if* $\text{conv}(\mathcal{S}) \cap \mathcal{K}^*(\gamma) \neq \emptyset$

Before diving into the formal proofs, we briefly describe the intuition behind these lemmas and what role they play in the reductions used to formally prove the complexity results that follow. As an examples, let us consider the first result, which shows that IMUBVP is in coNP. To do so, we demonstrate the existence of a certificate when the output to the IMUBVP is NO. The output NO means that x^0 is suboptimal with respect to any objective $d \in \mathcal{K}(\gamma)$ (the cone consisting of the set of objectives or positive multiples of objectives within distance γ of c). To verify the output NO, we must show, in principle, that for each direction $d \in \mathcal{K}(\gamma)$, there exists an $x \in \mathcal{S}$ such that $d^\top(x^0 - x) < 0$. This appears to be impossible to do in polynomial time, but Lemma 1 gives us a way to do it. To see this, observe that the NO answer is equivalent to $\mathcal{K}(\gamma) \cap \mathcal{D}(x^0) = \emptyset$, which we know from Lemma 1 holds if and only if there exists $\overline{x} \in \text{conv}(\mathcal{S}) \cap \text{int}\,(\mathcal{K}^*(\gamma))$. Thus, given such an $\overline{x}$, we need only verify its membership in $\text{conv}(\mathcal{S}) \cap \text{int}\,(\mathcal{K}^*(\gamma))$. $\text{int}\,(\mathcal{K}^*(\gamma))$ is a cone whose membership problem is known to be polynomially solvable [Glineur and Terlaky, 2004]. Membership problem in $\text{conv}(\mathcal{S})$ is NP-complete, but as such, has its own certificate, can be certified by expressing $\overline{x}$ as a convex combination of extreme points of $\text{conv}(\mathcal{S})$. As the proof of Lemma 1 shows, it is then the decomposition of $\overline{x}$ into a convex combination of extreme points of $\text{conv}(\mathcal{S})$ that serves as the certificate.

Formal Proofs. We now present the formal proofs. In the remainder of the paper, we assume for simplicity that $\|c\| = 1$ and $\text{conv}(\mathcal{S})$ is full-dimensional.

Theorem 4. *The IMUBVP is in* coNP.

Proof. We prove the theorem by showing the existence of a short certificate when the output to the problem is NO. Therefore, let an instance $(\gamma, x^0, c, A, b, r)$ of the IMUBVP for which the output is NO be given along with relevant input data. Since the output is NO, we must have that $\|c\| = 1 > \gamma$, since $d = 0$ is a valid solution otherwise. Furthermore, if $\gamma = 0$, the problem reduces to the MUBVP with input $(c^\top x^0, c, A, b, r)$ (is $c^\top x^0$ an upper bound for maximization over $\mathcal{S}$ with objective function vector c), which is a problem already known to be in coNP. Therefore, we assume that $0 < \gamma < \|c\| = 1$.

As already informally discussed, The NO answer is equivalent to the condition $\mathcal{K}(\gamma) \cap \mathcal{D}(x^0) = \emptyset$. Both $\mathcal{K}(\gamma)$ and $\mathcal{K}^*(\gamma)$ are full-dimensional and pointed, since $0 < \gamma < \|c\| = 1$. From Lemma 1, we then have that the NO answer is also equivalent to the existence of $\overline{x} \in \operatorname{conv}(\mathcal{S}) \cap \operatorname{int}(\mathcal{K}^*(\gamma))$, but we have noted earlier that simply providing such $\overline{x}$ is not itself a short certificate, since we cannot verify its membership in $\operatorname{conv}(\mathcal{S})$ polynomial time. We complete the proof by showing that there exists a subset of extreme points of $\mathcal{S}$ whose convex combination yields $\overline{x}$ and *is* a certificate, as in the proof of Lemma 1. Moreover, this certificate is short, since at most $n + 1$ extreme points are needed. □

We next show that not only is IMUBVP in coNP, it is complete for it.

Theorem 5. *The IMUBVP is complete for* coNP.

Proof. We show that the MUBVP can be Karp-reduced to the IMUBVP. Let an instance (γ, c, A, b, r) of the MUBVP be given. Then we claim this MUBVP can be decided by deciding an instance of the IMUBVP with inputs $(0, \gamma c, c, A, b, r)$. The IMUBVP with this input asks whether $\{d \in \mathbb{R}^n \mid \|c - d\| \leq 0\} \cap \{d \in \mathbb{R}^n \mid d^\top (x - \gamma c) \leq 0 \; \forall x \in \mathcal{S}\}$ is non-empty. The first set contains a single point, $d = c$. The intersection is non-empty if and only if c is in the cone given by the second set. c is in this cone if and only if

$$\begin{aligned} c^\top (x - \gamma c) \leq 0 \; \forall x \in \mathcal{S} &\Leftrightarrow c^\top x - \gamma \leq 0 \; \forall x \in \mathcal{S}, \\ &\Leftrightarrow c^\top x \leq \gamma \; \forall x \in \mathcal{S}. \end{aligned}$$

The last line above means the output of the MUBVP is YES. This indicates that the output of the original instance of the MUBVP is YES if and only if the output of the constructed instance of the IMUBVP is YES. □

Figure 7 illustrates the reduction from the MUBVP to the IMUBVP for three different γ values. In the proof, the point γc is constructed so as to have objective function value γ. The inverse problem is then just to determine whether the inequality $c^\top x \leq \gamma$ is valid for $\operatorname{conv}(\mathcal{S})$, so the equivalence to the MUBVP follows straightforwardly. The output to both the MUBVP and the IMUBVP is NO for γ_1, and YES for both γ_2 and γ_3. Note that this proof crucially depends on the fact that we do not assume $x^0 \in \mathcal{S}$ for the IMUBVP. Now, we consider IMLBVP, for which the proofs take a similar approach.

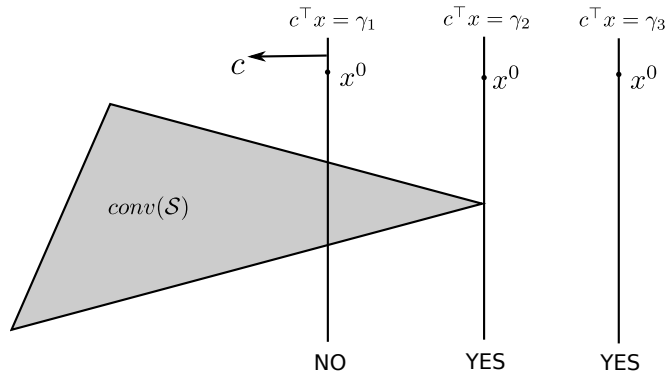
Theorem 6. *The IMLBVP is in* NP.

Proof. The proof is almost identical to the proof the IMUBVP is in coNP. We show that there exists a short certificate that can validate the output YES. Let an instance $(\gamma, x^0, c, A, b, r)$ of the IMLBVP be given such that the output is YES. As in the proof of Theorem 4, it is enough to consider the case where $0 < \gamma \leq \|c\| = 1$. When the output is YES, $\operatorname{int}(\mathcal{K}(\gamma)) \cap \mathcal{D}(x^0) = \emptyset$, which holds if and only if there exists $\overline{x} \in \operatorname{conv}(\mathcal{S}) \cap \mathcal{K}^*(\gamma)$ by Lemma 2. The remainder of the proof follows the same steps as that of Theorem 5. A subset of $\mathcal{S}$ whose convex combination yields $\overline{x}$ is a short certificate, since for any $d \in \operatorname{int}(\mathcal{K}(\gamma))$, we have that $d^\top (x^0 - \overline{x}) < 0 \Rightarrow d \notin \mathcal{D}(x^0)$. □

Lemma 3. *Let $\gamma \in \mathbb{Q}$ and let $\epsilon := 2^{-\max\{\langle c \rangle + \nu, \langle \gamma \rangle\} - 1}$ and $\delta := 2^{-\langle x^0 \rangle - \nu - 1}$.*

(i) If $\gamma > \max_{x \in \mathcal{S}} c^\top x$, then $\gamma > \max_{x \in \mathcal{S}} c^\top x + \epsilon$.

Figure 7: Reduction Example



(ii) If $\gamma \leq \max_{x\in\mathcal{S}} c^\top x$, then $\|c-d\| > \epsilon\delta$ for all $d \in \mathcal{D}((\gamma-\epsilon)\,c)$.

Proof. (i) First, we have that the encoding length of $\max_{x\in\mathcal{S}} c^\top x$ is bounded by $\langle c\rangle + \nu$. Then by Proposition 1, we have that $\gamma - \max_{x\in\mathcal{S}} c^\top x > \epsilon$.

(ii) For this part, let $\overline{x} \in \operatorname{argmax}_{x\in\mathcal{S}} c^\top x$ such that $\overline{x}$ is an extreme point of $\operatorname{conv}(\mathcal{S})$ and set $x^0 = (\gamma-\epsilon)c$. For an arbitrary $d \in \mathcal{D}(x^0)$, we have that

$$\begin{aligned}
d^\top(x^0-\overline{x}) \geq 0 &\Leftrightarrow d^\top(x^0-\overline{x}) - c^\top(x^0-\overline{x}) + c^\top(x^0-\overline{x}) \geq 0 \\
&\Leftrightarrow (d-c)^\top(x^0-\overline{x}) \geq c^\top\overline{x} - c^\top x^0 \\
&\Leftrightarrow (d-c)^\top(x^0-\overline{x}) \geq c^\top\overline{x} - \gamma + \epsilon \qquad (4)\\
&\Leftrightarrow (d-c)^\top(x^0-\overline{x}) \geq \epsilon \qquad (5)\\
&\Leftrightarrow \|c-d\|\|x^0-\overline{x}\| \geq \epsilon \qquad (6)\\
&\Leftrightarrow \|c-d\| \geq \frac{\epsilon}{\|x^0-\overline{x}\|} \\
&\Leftrightarrow \|c-d\| > \epsilon\delta \qquad (7)
\end{aligned}$$

Equation (4) follows by substituting $(\gamma-\epsilon)c$ for x^0; (5) follows from nonnegativity of $c^\top\overline{x}-\gamma$; and (6) follows from the Cauchy–Schwarz inequality. Equation (7) follows from the fact that $\|x^0-\overline{x}\| < 2^{<x^0>+\nu-1}$, again by Proposition 1, since the encoding length of $x^0-\overline{x}$ is bounded by the vertex complexity ν of $\operatorname{conv}\mathcal{S}$ and $\langle x^0\rangle$.

□

Theorem 7. *The IMLBVP is complete for* NP.

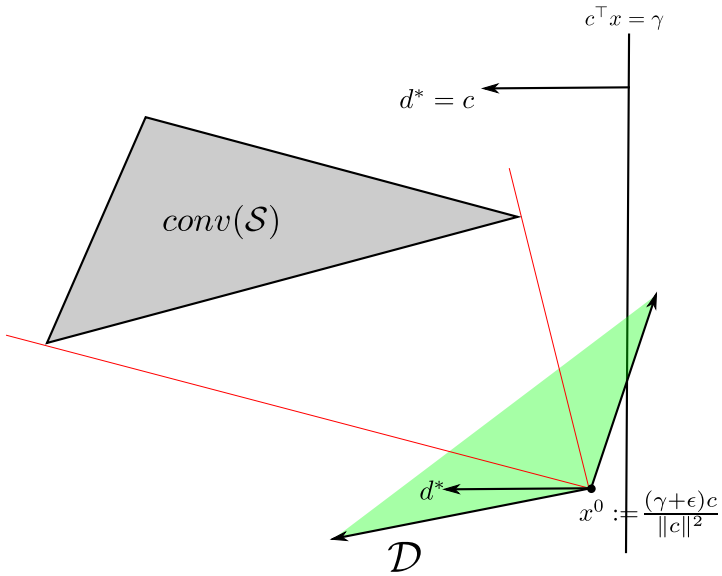
Proof. We show that the MLBVP can be Karp-reduced to the IMLBVP. Therefore, let an instance (γ, c, A, b, r) of the MLBVP be given and let ϵ and δ be given as in Lemma 3. Then we claim that the MLBVP can be resolved by deciding the IMLBVP with inputs $(\epsilon\delta, (\gamma-\epsilon)c, A, b, r)$. The IMLBVP asks whether the set $\operatorname{int}(\mathcal{K}(\epsilon\delta)) \cap \mathcal{D}((\gamma-\epsilon)c)$ is empty. Note that $c \in \operatorname{int}(\mathcal{K}(\epsilon\delta))$. If the output to the IMLBVP is YES, then $\operatorname{int}(\mathcal{K}(\epsilon\delta)) \cap \mathcal{D}((\gamma-\epsilon)c)$ is empty. This means $c \notin \mathcal{D}((\gamma-\epsilon)c)$, i.e.,

$$c \notin \left\{d \in \mathbb{R}^n \mid d^\top\left(x - (\gamma-\epsilon)\,c\right) \leq 0\ \forall x \in \mathcal{S}\right\}.$$

This in turn means that there exists an $\overline{x} \in \mathcal{S}$ such that

$$\begin{aligned}
c^\top\left(\overline{x} - (\gamma-\epsilon)\,c\right) > 0 &\Leftrightarrow c^\top\overline{x} - \gamma + \epsilon > 0 \\
&\Leftrightarrow c^\top\overline{x} > \gamma - \epsilon.
\end{aligned}$$

Figure 8: Lemma 3 on a Simple Example



By contraposition of Lemma 3, part (i), we then have that $c^\top \overline{x} \geq \gamma$. This means that the output to the MLBVP is YES.

When the output to the IMLBVP is NO, there exists $\overline{d} \in \text{int}\,(\mathcal{K}(\epsilon\delta)) \cap \mathcal{D}((\gamma - \epsilon)c)$. Since $\overline{d} \in \mathcal{D}((\gamma - \epsilon)c)$ and $\|c - \overline{d}\| \leq \epsilon\delta$, using contraposition of Lemma 3, part (ii), there is no $\overline{x} \in \mathcal{S}$ such that $c^\top \overline{x} \geq \gamma$, i.e., $c^\top x < \gamma$ for all $x \in \mathcal{S}$. This means the output to the MLBVP is NO. □

Note that the reduction presented in Theorem 7 can also be used in Theorem 5. The one presented in Theorem 5 is just simpler and does not require the introduction of ϵ (or at least it can be considered to be 0).

Figure 8 illustrates the case described in Lemma 3, inequality $c^\top x < \gamma$ holds for all $x \in \mathcal{S}$. Figure displays the cone of feasible d directions and optimal d as d^*. Output to both the MLBVP and the IMLBVP is negative for the inputs given in Theorem 7.

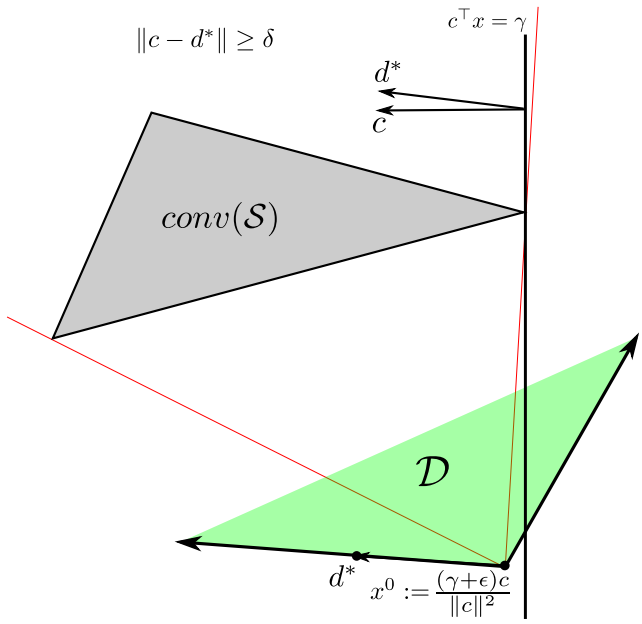
Figure 9 illustrates a case where optimal value of MILP is exactly γ. It can also be considered as an illustration of the case described in Lemma 3, forward problem optimal value is exactly gamma. Inverse optimal value is denoted by d^*. $\epsilon\delta$ as described in Lemma 3 is a lower bound for the inverse problem. Output to both the MLBVP and the IMLBVP for the displayed inputs is positive. From the figure it is easy to see that result in Lemma 3 holds when the forward problem optimal value is not exactly γ but strictly less.

Theorem 8. *The IMOVVP is complete for* D^P.

Proof. As noted before, the reduction presented in Theorem 7 can be used to reduce both the MUBVP and the MLBVP to the IMUBVP and the IMLBVP respectively. Using this reduction, the language of the IMOVVP can then be expressed as the intersection of the languages of the IMUBVP and the IMLBVP that are in coNP and NP, respectively. This proves that the IMOVVP is in class D^P. The IMOVVP is complete for D^P, since the MOVVP can be reduced to the IMOVVP using the same reduction. □

Note that optimal value verification of both inverse and forward problems are complete for D^P, optimizing both forward and inverse problems are in the same complexity class.

Figure 9: Lemma 3 on a Simple Example



5 Conclusion and Future Directions

In this paper, we formally defined various problems related to the inverse MILP problem in which we try to derive an objective function d closest to a given estimate c that make a given solution x^0 optimal over the feasible region $\mathcal{S}$ to an MILP. This problem can be seen as an optimization problem over the set of all inequalities valid for $\mathcal{S}$ and satisfied at equality by x^0. Alternatively, it can also be seen as optimization over the 1-polar with some additional constraints. Both these characterization make the connection the separation problem associated with $\mathcal{S}$ evident.

After defining the problem formally, we gave a cutting plane algorithm for solving it under the ℓ_1 and ℓ_∞ norms and observed that the separation problem for the feasible region is equivalent to the original forward problem, enabling us to conclude by the framework of Grötschel et al. [1993] that the problem can be solved with a polynomial number of calls to an oracle for solving the forward problem.

This algorithm places the decision version of inverse MILP in the complexity class Δ_2^p, but it is possible to prove a stronger result. The main contribution of this study is to show that this decision problem is complete for the class coNP, which is on the same level of the polynomial-time hierarchy of that of the forward problem. We proved the problem is in coNP by giving a short certificate for the output NO and then show it is complete for coNP by reducing the MUBVP to the inverse MILP decision problem. We also provide a reduction for the inverse lower bound problem. Finally, we show that the inverse optimal value verification problem is complete to the class D^P, which is precisely the same class containing the MILP optimal value verification problem.

Theorem 1 states that an optimization problem (over a convex set) can be solved in polynomial time given an oracle for the separation problem. Technically, this does not allow us to place the optimization and separation problems on precisely the same level of the polynomial hierarchy. It is likely that the result of Grötschel et al. [1993] can be modified slightly in order to show that optimization and separation are indeed on the same level of the hierarchy. There are also some interesting open questions remaining to be explored with respect to complexity.

Finally, we have implemented the algorithm and a computationally oriented study is left as future work. Such a study would reveal the practical performance of the separation-optimization procedure and investigate the possible relationship between the number of iterations (oracle calls) and the polyhedral complexity (vertex/facet complexity), among other things. This may provide practical estimates for the number of iterations required to solve certain classes of problems.

References

Ravindra K. Ahuja and James B. Orlin. Inverse optimization. *Operations Research*, 49(5):771–783, 2001. doi: 10.1287/opre.49.5.771.10607.

S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2007.

R. Beigel and L. Fortnow. Are cook and karp ever the same? In *Proceedings of the* 18th *Annual IEEE Conference on Computational Complexity*, pages 333–336, 2003.

E. Andrew Boyd. Fenchel cutting planes for integer programs. *Operations Research*, 42(1):53–64, 1994. doi: 10.1287/opre.42.1.53. URL `http://dx.doi.org/10.1287/opre.42.1.53`.

M. C. Cai, X. G. Yang, and J. Z. Zhang. The complexity analysis of the inverse center location problem. *Journal of Global Optimization*, 15(2):213–218, September 1999. ISSN 1573-2916. doi: 10.1023/A:1008360312607. URL `https://doi.org/10.1023/A:1008360312607`.

Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, pages 151–158, New York, NY, USA, 1971. ACM. doi: 10.1145/800157.805047. URL `http://doi.acm.org/10.1145/800157.805047`.

Jack Edmonds. Matroids and the greedy algorithm. *Mathematical programming*, 1(1):127–136, 1971.

Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979. ISBN 0716710447.

François Glineur and Tamas Terlaky. Conic formulation for l p-norm optimization. *Journal of Optimization theory and Applications*, 122(2):285–307, 2004.

M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Algorithms and Combinatorics. Springer, second corrected edition edition, 1993.

Clemens Heuberger. Inverse combinatorial optimization: a survey on problems, methods, and results. *Journal of Combinatorial Optimization*, 8(3):329–361, September 2004. ISSN 1573-2886. doi: 10.1023/B:JOCO.0000038914.26975.9b. URL `https://doi.org/10.1023/B:JOCO.0000038914.26975.9b`.

Siming Huang. *Inverse problems of some NP-complete problems*, pages 422–426. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. ISBN 978-3-540-32440-9. URL `https://doi.org/10.1007/11496199_45`.

Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

Mark W. Krentel. The complexity of optimization problems. *Journal of Computer and System Sciences*, 36(3):490 – 509, 1988. ISSN 0022-0000. doi: https://doi.org/10.1016/0022-0000(88)90039-6. URL `http://www.sciencedirect.com/science/article/pii/0022000088900396`.

Mark William Krentel. *The Complexity of Optimization Problems*. PhD thesis, Cornell University, Ithaca, New York, May 1987a.

M.W. Krentel. *The Complexity of Optimization Problems*. PhD thesis, Cornell University, 1987b.

C. H. Papadimitriou and M. Yannakakis. The complexity of facets (and some facets of complexity). In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, STOC '82, pages 255–260, New York, NY, USA, 1982. ACM. ISBN 0-89791-070-2. doi: 10.1145/800070.802199. URL `http://doi.acm.org/10.1145/800070.802199`.

Christos H. Papadimitriou. Computational complexity. In *Encyclopedia of Computer Science*, pages 260–265. John Wiley and Sons Ltd., Chichester, UK, 2003. ISBN 0-470-86412-5. URL `http://dl.acm.org/citation.cfm?id=1074100.1074233`.

Andrew J. Schaefer. Inverse integer programming. *Optimization Letters*, 3(4):483–489, September 2009. ISSN 1862-4480. doi: 10.1007/s11590-009-0131-z. URL `https://doi.org/10.1007/s11590-009-0131-z`.

Alexander Schrijver. *Theory of integer and linear programming*. Wiley, Chichester, 1986.

L.J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1 – 22, 1976a. ISSN 0304-3975. doi: https://doi.org/10.1016/0304-3975(76)90061-X. URL `http://www.sciencedirect.com/science/article/pii/030439757690061X`.

L.J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3:1–22, 1976b.

Alan M Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London mathematical society*, 2(1):230–265, 1937.

H. Vollmer and K.W. Wagner. Complexity classes of optimization functions. *Information and Computation*, 120(2):198 – 219, 1995. ISSN 0890-5401. doi: https://doi.org/10.1006/inco.1995.1109. URL `http://www.sciencedirect.com/science/article/pii/S0890540185711091`.

Lizhi Wang. Cutting plane algorithms for the inverse mixed integer linear programming problem. *Operations Research Letters*, 37(2):114 – 116, 2009. ISSN 0167-6377. doi: http://dx.doi.org/10.1016/j.orl.2008.12.001. URL `http://www.sciencedirect.com/science/article/pii/S0167637708001326`.