

An Algorithm for Solving Inverse Integer Programs

Aykut Bulut¹

Joint work with:
Ted Ralphs¹

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

INFORMS Annual Meeting
October 2012

1 Inverse IP

2 Complexity

3 Algorithm

4 Computational Results

1 Inverse IP

2 Complexity

3 Algorithm

4 Computational Results

- 1 Inverse IP
- 2 Complexity
- 3 Algorithm
- 4 Computational Results

- 1 Inverse IP
- 2 Complexity
- 3 Algorithm
- 4 Computational Results

Definitions

We consider an IP

$$z_{IP} = \min_{x \in \mathcal{P}} d^T x, \quad (1)$$

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

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

For a given $c \in \mathbb{R}^n$, $x^0 \in \mathcal{P}$, the inverse problem is defined as follows.

$$\begin{aligned} & \min \|c - d\| \\ & \text{s.t.} \\ & d^T x^0 \leq d^T x \quad \forall x \in \mathcal{P}. \end{aligned} \quad (2)$$

Assumption: \mathcal{P} is bounded.

$$\begin{aligned} & \min \|c - d\| \\ & s.t. \\ & d^T x^0 \leq d^T x \quad \forall x \in \mathcal{P}. \end{aligned} \tag{4}$$

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

Inverse IP with l_1 norm

$$z_{IP}^1 = \min \sum_{i=1}^n \theta_i$$

s.t.

$$c_i - d_i \leq \theta_i$$

$$\forall i \in \{1, 2, \dots, n\} \quad (3)$$

$$d_i - c_i \leq \theta_i$$

$$\forall i \in \{1, 2, \dots, n\}$$

$$d^T x^0 \leq d^T x$$

$$\forall x \in \mathcal{E}.$$

Inverse IP with l_∞ norm

$$z_{IP}^\infty = \min y$$

s.t.

$$c_i - d_i \leq y \quad \forall i \in \{1, 2, \dots, n\} \quad (4)$$

$$d_i - c_i \leq y \quad \forall i \in \{1, 2, \dots, n\}$$

$$d^T x^0 \leq d^T x \quad \forall x \in \mathcal{E}.$$

For the remainder of the presentation, we deal with the case of l_∞ norm. Let \mathcal{S} represent feasible set of the inverse IP, defined as

$$\mathcal{S} = \{(y, d) \in \mathbb{R} \times \mathbb{R}^n \mid y \geq \|c - d\|_\infty, d^T(x^0 - x) \leq 0 \forall x \in \mathcal{E}\}.$$

Note that \mathcal{S} is a polyhedron.

Polynomially Solvable Forward Problems

Ahuja and Orlin [1] determines the complexity of inverse problem when the forward problem is polynomially solvable.

Theorem

(Ahuja and Orlin [1]) If a forward problem is polynomially solvable for each linear cost function, then corresponding inverse problems under l_1 and l_∞ are polynomially solvable.

Forward Problems

Define the following problems related to IP and inverse IP.

Definition

IP decision problem: Given $\gamma \in \mathbb{Q}$ decide whether $d^T x \leq \gamma$ holds for some $x \in \mathcal{P}$.

Definition

IP optimization problem: Find solution vector x^* such that $x^* \in \operatorname{argmin}_{x \in \mathcal{P}} d^T x$ or decide the problem is unbounded or decide the problem is infeasible.

Definition

Inverse IP decision problem: Given $\gamma \in \mathbb{Q}$ decide whether $y \leq \gamma$ holds for some $(y, d) \in \mathcal{S}$.

Definition

Inverse IP optimization problem: Find solution vector (y^*, d^*) , such that $(y^*, d^*) \in \operatorname{argmin}_{(y,d) \in \mathcal{S}} y$.

Definition

Inverse IP separation problem: Given a vector $(\bar{y}, \bar{d}) \in \mathbb{Q} \times \mathbb{Q}^n$, decide whether (\bar{y}, \bar{d}) is in \mathcal{S} , and if not, find a hyperplane that separates (\bar{y}, \bar{d}) from \mathcal{S} , i.e., find $\pi \in \mathbb{Q}^{n+1}$ such that $\pi^T \begin{bmatrix} \bar{y} \\ \bar{d} \end{bmatrix} < \min \left\{ \pi^T \begin{bmatrix} y \\ d \end{bmatrix} \mid (y, d) \in \mathcal{S} \right\}$.

Component 1—An Observation

Recall inverse IP optimization problem.

$$z_{IP}^{\infty} = \min y$$

s.t.

$$c_i - d_i \leq y$$

$$\forall i \in \{1, 2, \dots, n\}$$

$$d_i - c_i \leq y$$

$$\forall i \in \{1, 2, \dots, n\}$$

$$d^T x^0 \leq d^T x$$

$$\forall x \in \mathcal{E}.$$

Definition

Inverse IP separation problem: Given a vector $(\bar{y}, \bar{d}) \in \mathbb{Q} \times \mathbb{Q}^n$, decide whether (\bar{y}, \bar{d}) is in \mathcal{S} , and if not, find a hyperplane that separates (\bar{y}, \bar{d}) from \mathcal{S} , i.e., find $\pi \in \mathbb{Q}^{n+1}$ such that $\pi^T \begin{bmatrix} \bar{y} \\ \bar{d} \end{bmatrix} < \min \left\{ \pi^T \begin{bmatrix} y \\ d \end{bmatrix} \mid (y, d) \in \mathcal{S} \right\}$.

Component 2–GLS Theorem

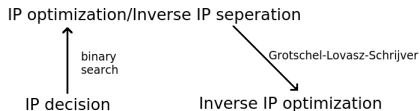
The following theorem by Grötschel et al. indicates the relationship between separation and optimization problems.

Theorem

(Grötschel et al. [2]) *Given an oracle for the separation problem, the optimization problem over a given polyhedron with linear objective can be solved in time, polynomial in φ , n and the **encoding length of objective coefficient vector**, where facet complexity of polyhedron is at most φ .*

Complexity of IP optimization/decision problems

Figure: Problem relations



Theorem

Inverse IP optimization problem under l_∞/l_1 norm is solvable in time polynomial of φ , $n + 1/2n$, and encoding length of $(1, 0, \dots, 0)/(1, \dots, 1, 0, \dots, 0)$, given an oracle for the IP decision problem.

Corollary

Inverse IP decision problem is in Δ_2^P .

Proposed Algorithm

First, we define two parametric problems named P_k and $InvP_k$ as follows

$$\min_{x \in \mathcal{P}} d^{kT} x \quad (P_k)$$

min y

s.t.

$$c_i - d_i \leq y \quad \forall i \in \{1, 2, \dots, n\} \quad (InvP_k)$$

$$d_i - c_i \leq y \quad \forall i \in \{1, 2, \dots, n\}$$

$$d^T x^0 \leq d^T x \quad \forall \mathcal{E}^{k-1}$$

where \mathcal{E}^{k-1} is the set of extreme points found so far.

Algorithm 1

```
 $k \leftarrow 1$   
 $d^k \leftarrow c$   
Solve  $P_k, x^k \leftarrow x^*$   
while  $d^{kT}(x^0 - x^k) > 0$  do  
     $k \leftarrow k + 1$   
    Solve  $InvP_k, d^k \leftarrow d^*$   
    Solve  $P_k, x^k \leftarrow x^*$   
end while
```

- x^k is an optimal solution to P_k , whereas d^k is an optimal solution to $invP_k$.
- The algorithm stops when a generated cut is not violated by the current solution.

Computational Results

- We used MIPLIB3 benchmark problems to create our inverse problems.
- We perturbed objective function coefficients, solved the problems and used the resulting solutions as x^0 's.
- Using generated x^0 's we created inverse IP problems.
- Experiments are conducted using COIN-OR Branch and Cut (Coin-Cbc) and COIN-OR Open Solver Interface (Coin-Osi) tools with Condor on a cluster running Debian operating system.

Table: Computational results– l_∞

Problem name	Iter.	$\ c - d^*\ _\infty$	$\ c\ _\infty$	Time
air03	1	0	13746	05:05:24
bell3a	4	4.49554	60000	00:09:17
blend2	9	0.00508924	24.0142	00:05:00
cap6000	2	13.9	91110	01:40:01
dcmulti	867	7	1800	00:25:38
egout	7	1.38485	43.71	00:09:10
fiber	2	27.7829	729670	08:21:59
gesa3	1	0	1548890	00:00:37
gt2	1	0	7797	00:00:05
khb05250	6	0.75	2500000	00:01:05
l152lav	5	3.28571	268	00:02:02
lseu	1	0	517	00:00:17
mas74	1	0	1	02:28:05
misc03	1	0	1	00:00:02
misc06	1163	0.410302	1	01:01:38
mod008	2	4.625	87	00:00:11
mod010	4	3.78947	266	00:03:31
noswot	4	1	1	1–09:33:20
p0201	1	0	9600	00:00:00
p0282	1	0	160646	00:00:00
pk1	11	0.138533	1	01:16:51
pp08aCUTS	2380	4.69136	500	08:42:09
qiu	20	16.947	114.034	00:08:11
qnet1	6	0.41136	1	00:01:59
rgn	3	1	3	00:00:00
rout	10	0.555358	1	06:52:02
stein27	1	0	1	00:00:00
vpm1	4	0.111111	1	1–02:55:03

References



Ravindra K. Ahuja and James B. Orlin.

Inverse optimization.

Operations Research, 49(5):771–783, September/October 2001.



Martin Grötschel, László Lovász, and Alexander Schrijver.

Geometric Algorithms and Combinatorial Optimization, volume 2 of *Algorithms and Combinatorics*.

Springer, second corrected edition edition, 1993.

This is end of presentation!

Thank you for listening!

A Small Example:

Define forward problem as follows and let $x^0 = (2, 1)$.

$$\min 2x_1 + x_2$$

s.t.

$$2x_1 + 3x_2 \geq 1$$

$$-x_1 + 2x_2 \leq 3$$

$$5x_1 + 5x_2 \leq 16$$

$$2x_1 - 4x_2 \leq 3$$

$$x_1, x_2 \geq 0$$

$$x_1, x_2 \text{ integer}$$

A Small Example:

k , d^k and x^k values through iterations are given in Table 2.

Table: k , d^k and x^k values through iterations

	k	d^k	x^k
initialization	1	(2, 1)	(0, 1)
iteration 1	2	(0, 3)	(1, 0)
iteration 2	3	(0, -1)	(1, 2)
iteration 3	4	(0, 0)	(0, 1)

Inverse IP optimal value is $y^* = \|c - d^4\|_\infty = 2$. Inverse IP optimal solution is $d^4 = (0, 0)$.

Table: Computational results– l_1

Problem name	Iter.	$\ c - d^*\ _1$	$\ c\ _1$	Time
bell3a	1	26.7727	1.78994e+06	
cap6000	2	19	1.29696e+07	
dcmulti	1652	193.312	30648.9	
egout	34	28.472	793.391	
enigma	1	0	45	
fiber	1	3.28735	4.80778e+07	
flugpl	1	0	25380	
gesa2	1	0	9.09347e+07	
gesa3	1	0	8.73813e+07	
gt2	1	0	291998	
khh05250	3	8	6.5264e+07	
1152lav	21	85	382524	
lseu	1	0	15494	
mas74	1	0	1.0015	
misc03	1	0	1	
misc06	10	1	1	
mod008	6	23	23554	
mod010	60	101	489211	
p0201	1	50	99900	
p0282	1	0	1.30262e+06	
pk1	1	1	1	
qnet1	25	2.07111	124	
rgn	53	20	160	
rout	7	1	1	
stein27	1	0	27	