Integrality

Wednesday, June 21, 2023 1:56 PM

Consider LP $\begin{pmatrix} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{pmatrix}$ where $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$

¿What if we restrict (some of) the variables to be integers?

Why?

- Variables may be modeling an integer quantity
  ex) How many units of product to produce
- Variables may model a binary (yes or no) decision
  ex) Whether a factory should be located in a particular city

Integer Program (IP) (or ILP)

$$\begin{pmatrix} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \\ & \underline{x \in \mathbb{Z}^n} \end{pmatrix} \leftarrow \text{Integrality constraints}$$

0-1 Programming / Binary Programming

$$\begin{pmatrix} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \\ & x \in \{0,1\}^n \end{pmatrix} \leftarrow \text{binary constraints}$$

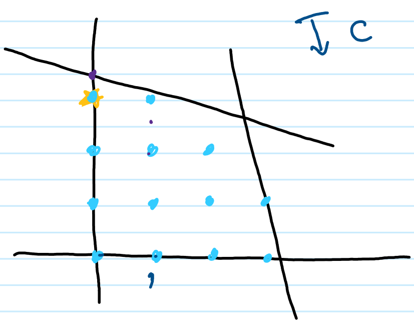
(or MILP)

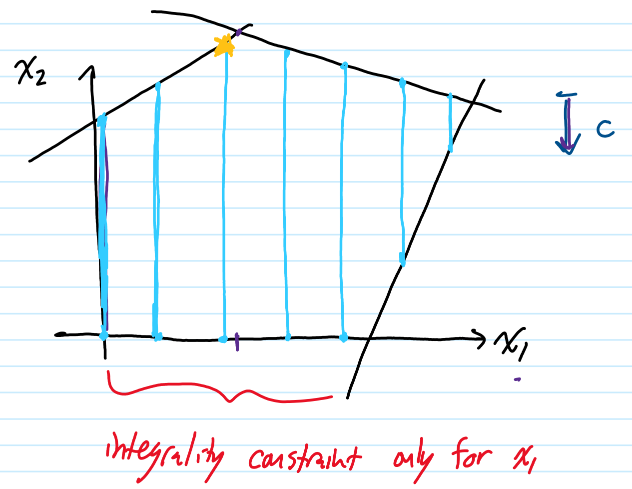
Mixed Integer Program (MIP)

$$\left(\begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \\ & x_j \in \mathbb{Z}^n, \ j \in \mathcal{I} \end{array}\right)$$

Indexing set of integral variables

Graphical Examples





Fun IP Example

Wednesday, June 21, 2023 2:33 PM

Sudoku

$X_{(1,1),5} = 1,\ X_{(1,1),k} = 0$

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4		6	8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

- An IP feasibility problem:

find filled board

s.t.
- every row contains each digit 1-9 exactly once
- every col'm contains each digit 1-9 exactly once
- every 3x3 box contains each digit 1-9 exactly once
- every clue is used

Let's try to formulate as an IP...

Naive Formulation

$$
\begin{aligned}
\text{Find } & X \in \mathbb{R}^{9\times 9} \\
\text{s.t. } & \sum_{i=k}^{k+2}\sum_{j=l}^{l+2} X_{ij} = \underline{45} && \forall k,j \in \{1,4,7\} && \leftarrow \text{"every 3x3 box contains each digit"} \\
& \sum_{j=1}^{9} X_{kj} = 45 && \forall k \in \{1,\dots,9\} && \leftarrow \text{"rows contain each digit"} \\
& \sum_{i=1}^{9} X_{il} = 45 && \forall l \in \{1,\dots,9\} && \leftarrow \text{"col.s contain each digit"} \\
& X_{ij} = c_{ij} && (i,j) \in C && \leftarrow \text{"each hint used"} \\
& 1 \le X_{kl} \le 9 && \forall k,l \in \{1,\dots,9\} && \leftarrow \text{variable bounds} \\
& X_{kl} \in \mathbb{Z} && \forall k,l \in \{1,\dots,9\} && \leftarrow \text{integrality}
\end{aligned}
$$

IP Formulation

$X_{(i,j),k}$: number k is used in cell (i,j)

$$
\left(
\begin{array}{lll}
\text{Find} & X \in \mathbb{R}^{9\times 9\times 9} & \\
\text{s.t.} & \sum_k X_{(i,j),k} = 1 & \forall (i,j) \in \{1,\dots,9\}^2 \\
 & \sum_j X_{(i,j),k} = 1 & \forall i,k \in \{1,\dots,9\} \\
 & \sum_i X_{(i,j),k} = 1 & \forall j,k \in \{1,\dots,9\} \\
 & \sum_{i=S_i}^{S_i+2} \sum_{j=S_j}^{S_j+2} X_{(i,j),k} = 1 & \forall S_i,S_j \in \{1,4,7\}, \forall k \in \{1,\dots,9\} \\
 & X_{(i,j),k} = \text{clues}(i,j,k) & \\
 & X_{(i,j),k} \in \{0,1\} & \forall i,j,k \in \{1,\dots,9\}
\end{array}
\right)
$$

every cell has only one #

every row uses each digit

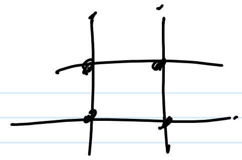
every col. uses each digit

every 3x3 box uses each digit

every clue is used

IP's are Hard

Wednesday, June 21, 2023 4:54 PM



Difficulty to Solve

- In some special circumstances every vertex of the underlying polyhedron (ie the feasible region w/out the integrality constraints) is an integer $\Rightarrow$ Can solve IP via LP methods (Simplex, interior pt)
- In general, IPs (MIPs) are NP Hard

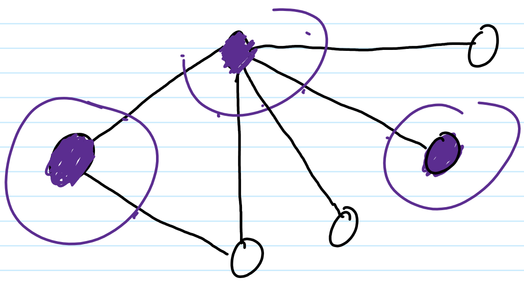
  Meaning: IPs cannot be solved efficiently (provided $P \neq NP$)

Lets Prove this...

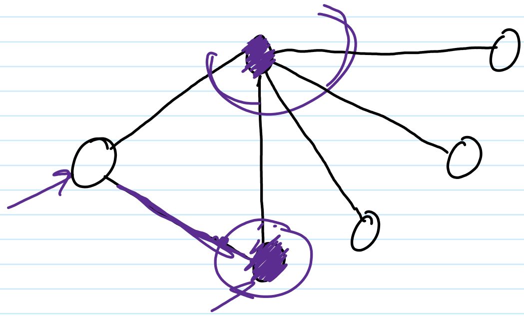
To prove that a problem (P_1) is NP-Hard, we take another problem (P_2) known to be NP-Hard (e.g. vertex cover) and show that P_2 can be efficiently formulated as P_1

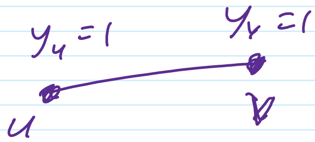
Vertex Cover Problem

Given a graph $G(V, E)$, a vertex cover is a set of vertices that includes @ least one endpoint of every edge



Problem: Find vertex cover of minimum cardinality





IP Formulation

$y \qquad y_v, \ \forall v$ ~~vertex~~

$$\begin{pmatrix} \min & \sum_{v \in V} y_v & \\ \text{s.t.} & y_v + y_u \geq 1 & \forall uv \in E \\ & 0 \leq y_v \leq 1 & \forall v \in V \\ & y_v \in \mathbb{Z} & \forall v \in V \end{pmatrix}$$

← min sum of "used" vertices

← y includes @ least one endpoint $\forall$ edges

y is set of vertices

$\Downarrow$

Soln is min vertex cover

$y_v \in \{0,1\}^n$

Branch and Bound

Wednesday, June 21, 2023 5:27 PM

Finds exact soln in inefficient time

$$\begin{pmatrix} \min & c^T x \\ & Ax \leq b \\ & x \in \mathbb{Z}^n \end{pmatrix} \text{IP}$$

$\downarrow$ Linear Relaxation

$$\begin{pmatrix} \min & c^T x \\ & Ax \leq b \end{pmatrix} \text{LP}$$

$\downarrow$ solve

Soln x^{LR}

If $x^{LR} \in \mathbb{Z}^n$ $\rightarrow$ Solved!

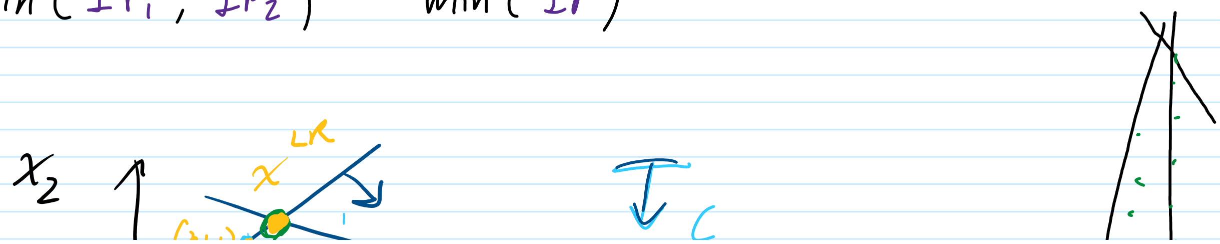
Otherwise $\rightarrow$ $\exists i$ s.t. $x_i^{LR} \notin \mathbb{Z}$

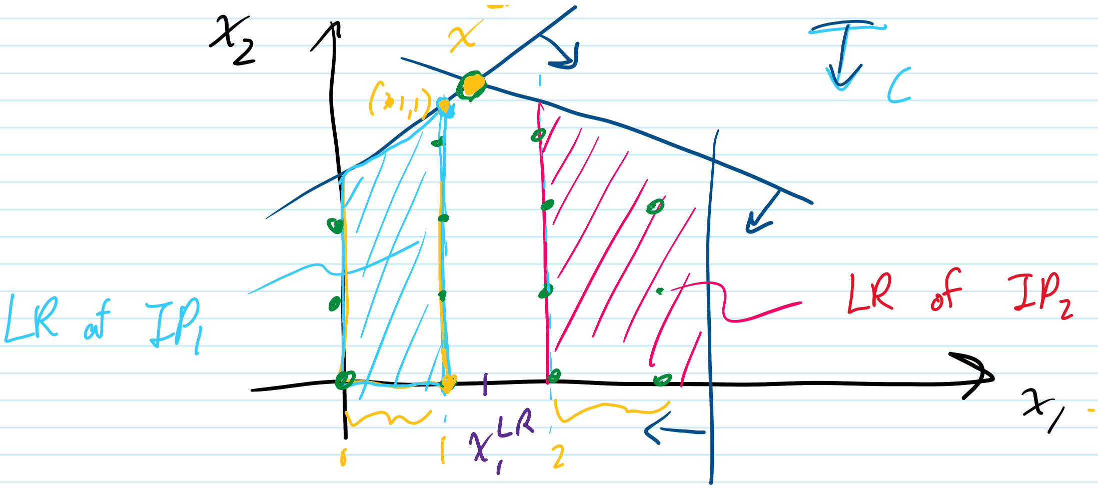
Create Subproblems

$$\begin{pmatrix} \min & c^T x \\ & Ax \leq b \\ & x_i^{LR} \leq \\ & x \in \mathbb{Z}^n \end{pmatrix} \text{IP}_1 \qquad \begin{pmatrix} \min & c^T x \\ & Ax \leq b \\ & x_i \geq \lceil x_i^{LR} \rceil \\ & x \in \mathbb{Z}^n \end{pmatrix} \text{IP}_2$$

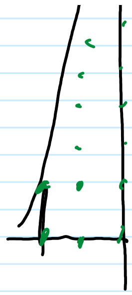
$\vdots$

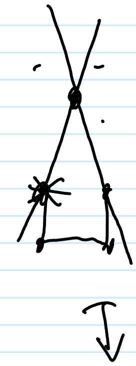
Note: $\min(\text{IP}_1, \text{IP}_2) = \min(\text{IP})$

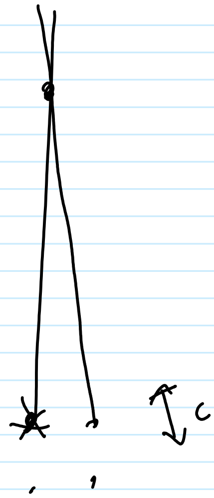




x_2
$(x_1, 1)$
c
LR of IP$_1$
LR of IP$_2$
x_1
0
1
x_1^{LR}
2







c

Large Scale LPs

Wednesday, June 21, 2023 5:27 PM

$$\begin{pmatrix} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{pmatrix}_{LP}$$

Most practitioners solve LP via Simplex or Interior Point Methods

Pros:

- typically find high accuracy solutions in reasonable amt. of time
- Reliable implementations widely availble (Gurobi, CPLEX, ...)

Both methods rely on factorization (i.e. matrix inversion)

Cons:

- Speed heavily depends on the sparsity pattern of the linear system
- For large enough problems factorization will run out of memory (even when original problem can fit into mem)
- Challenging to parallelize or distribute across multiple machines

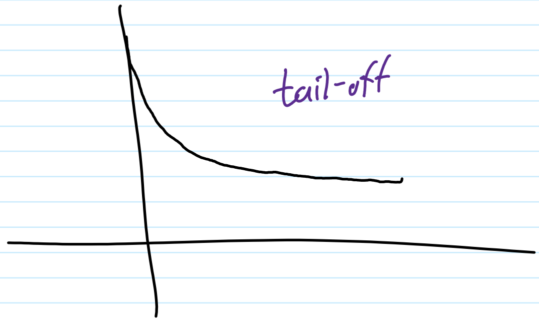
Becoming a bigger issue as problem sizes keep growing to meet applications!

Solution: Develop LP methods that rely on matrix-vector multiplication instead

e.g. First order methods like gradient descent

Pros:

- Easily parallelized
- Low memory footprint
- Each matrix-vector mult. reliably fast



Con:

- Struggle to obtain high accuracy solution in reasonable time frame

Modern Challenges: Overcoming this

Best attempts use variety of techniques (PDLP):

- diagonal preconditioning
- presolving
- adaptive step sizes
- adaptive restarts ⟵

Restarts:

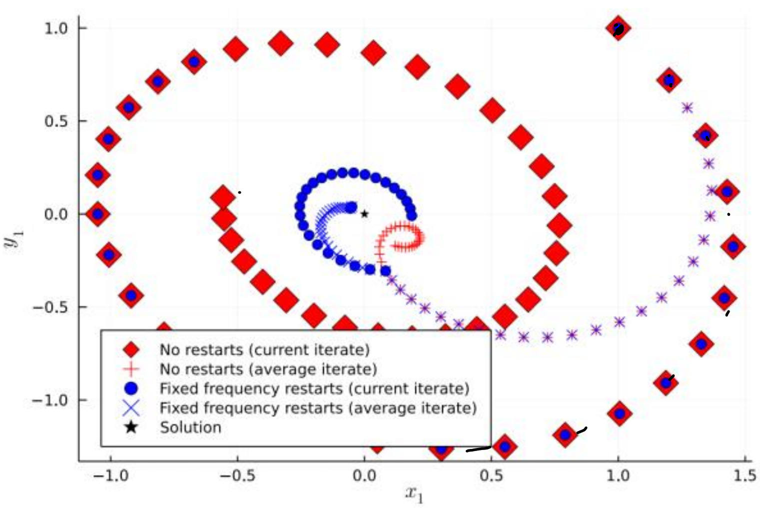


Figure 1: Plot of the first 50 iterates of non-restarted and restarted PDHG for a simple two-dimensional bilinear problem $\mathcal{L}(x, y) = xy$ with $\eta = 0.2$. The restart length is for the fixed frequency restarts is 25. The unique optimal solution is $(0, 0)$.

[Applegate et al, '22]