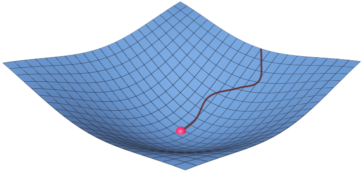
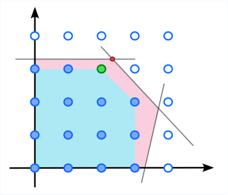
Optimization

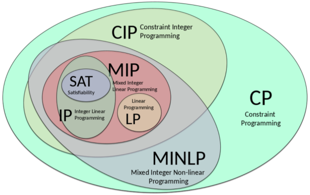


Paul J. Atzberger

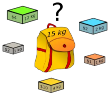
University of California Santa Barbara (UCSB)

Integer Linear Programming (ILP)





Gleixner 2018



Vidyalay 2020

Kuo 2024

Motivations

- Many problems only allow for a discrete set of possibilities.
- Examples include the number of people, products, or vehicles in planning.
- Also, distinct types, categories, or outcomes.

Integer Linear Program (ILP)

$$\begin{array}{ll} \min & \mathbf{c}^T\mathbf{x} \\ \text{subject} & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq 0, \quad \mathbf{x} \in \mathbb{Z}. \end{array}$$

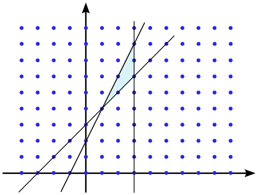
Approaches to ILPs

- Relax the $\mathbf{x} \in \mathbb{Z}$ condition and solve LP for $\tilde{\mathbf{x}}^*$.
- Approximate the solution by $\bar{\mathbf{x}}^* = [\tilde{\mathbf{x}}^*]$.
- Use LP to approximate the solution and if not in $\mathbb{Z}$ introduce successive new constraints to rule-out non-integer points (*Gomory Cuts*).
- Other strategies include: branch-and-bound, other cutting-plane rules, and heuristic search.

Examples and Applications

- Planning / scheduling: Traveling Salesman Problem (TSP).
- Theorem provers / logic: Satisfiability (SAT), 0-1 ILP.
- Resource allocation: cell phone networks, job scheduling.

Integer Linear Programming: Example



Update

$$y_{ij}^{+} = y_{ij} - \frac{y_{iq}}{y_{pq}} y_{pj}, \quad \text{if } i \neq p$$

$$y_{pj}^{+} = \frac{y_{pj}}{y_{pq}}, \quad \text{if } i = p,$$

$$s_N^T = c_N^T - c_B^T B^{-1} N.$$

ILP

$$\begin{array}{ll} \min & \mathbf{c}^T \mathbf{x} \\ \text{subject} & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq 0, \quad \mathbf{x} \in \mathbb{Z}. \end{array}$$

Example (LP1)

$$\begin{array}{ll} \min & -x_1 - 2x_2 \\ \text{subject} & -2x_1 + x_2 + x_3 = 2 \\ & -x_1 + x_2 + x_4 = 3 \\ & x_1 + x_5 = 3 \\ & \mathbf{x} \geq 0, \quad \mathbf{x} \in \mathbb{Z}. \end{array}$$

Canonical Tableau

	a_1	a_2	a_3	a_4	a_5	b
	−2	1	1	0	0	2
	−1	1	0	1	0	3
	1	0	0	0	1	3
c^T	−1	−2	0	0	0	0

$\mathcal{B} = \{3, 4, 5\}$
(canonicalize)

→

	a_1	a_2	a_3	a_4	a_5	b
	−2	1	1	0	0	2
	−1	1	0	1	0	3
	1	0	0	0	1	3
r^T	−1	−2	0	0	0	0

$\mathcal{B} = \{3, 4, 5\}, \quad q = 2$
$\frac{y_{i0}}{y_{iq}} = [2, 3], i = [1, 2], p = 1.$

iteration:1

	a_1	a_2	a_3	a_4	a_5	b
	−2	1	1	0	0	2
	1	0	−1	1	0	1
	1	0	0	0	1	3
r^T	−5	0	2	0	0	4

$\mathcal{B} = \{2, 4, 5\}, \quad q = 1,$
$\frac{y_{i0}}{y_{iq}} = [1, 3], i = [2, 3], p = 2.$

→

iteration:2

	a_1	a_2	a_3	a_4	a_5	b
	0	1	−1	2	0	4
	1	0	−1	1	0	1
	0	0	1	−1	1	2
r^T	0	0	−3	5	0	9

$\mathcal{B} = \{2, 1, 5\}, \quad q = 3,$
$\frac{y_{i0}}{y_{iq}} = [2], i = [3], p = 3.$

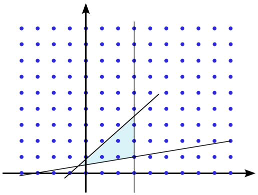
iteration:3

	a_1	a_2	a_3	a_4	a_5	b
	0	1	0	1	1	6
	1	0	0	0	1	3
	0	0	1	−1	1	2
r^T	0	0	0	2	3	15

$\mathcal{B} = \{2, 1, 3\}, \quad \text{(final)}$

→ **solution (valid)**
$\mathbf{x} = [3, 6, 2, 0, 0]$

Integer Linear Programming: Example



Update

$$y_{ij}^{+} = y_{ij} - \frac{y_{iq}}{y_{pq}} y_{pj}, \quad \text{if } i \neq p$$

$$y_{pj}^{+} = \frac{y_{pj}}{y_{pq}}, \quad \text{if } i = p,$$

$$s_N^T = c_N^T - c_B^T B^{-1} N.$$

Example (LP2)

$$\begin{array}{ll} \min & -2x_1 - 3x_2 \\ \text{subject} & x_1 + x_3 = 3 \\ & x_1 - 6x_2 + x_4 = -3 \\ & -9x_1 + 8x_2 + x_5 = 1 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0. \end{array}$$

initial-canonical

	a_1	a_2	a_3	a_4	a_5	b
	1	0	1	0	0	3
	1	−6	0	1	0	−3
	−9	8	0	0	1	1
r^T	−2	−3	0	0	0	0

$\mathcal{B} = \{3, 4, 5\}$, $q = 2$

$\frac{y_{i0}}{y_{iq}} = [\frac{1}{8}]$, $i = [3]$, $p = 3$

→

iteration 1

	a_1	a_2	a_3	a_4	a_5	b
	1	0	1	0	0	3
	$-\frac{23}{4}$	0	0	1	$\frac{3}{4}$	$-\frac{9}{4}$
	$-\frac{9}{8}$	1	0	0	$\frac{1}{8}$	$\frac{1}{8}$
r^T	$-\frac{43}{8}$	0	0	0	$\frac{3}{8}$	$\frac{3}{8}$

$\mathcal{B} = \{3, 4, 2\}$, $q = 1$

$\frac{y_{i0}}{y_{iq}} = [3]$, $i = [1]$, $p = 1$

→

iteration 2

	a_1	a_2	a_3	a_4	a_5	b
	1	0	1	0	0	3
	0	0	$\frac{23}{4}$	1	$\frac{3}{4}$	15
	0	1	$\frac{9}{8}$	0	$\frac{1}{8}$	$\frac{7}{2}$
r^T	0	0	$\frac{43}{8}$	0	$\frac{3}{8}$	$\frac{33}{2}$

$\mathcal{B} = \{1, 4, 2\}$,

(final)

→

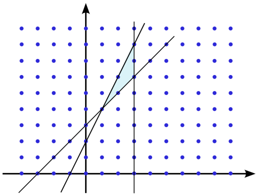
solution (fail):

$\mathbf{x} = \left[3, \frac{7}{2}, 0, 15, 0\right].$

Integer Linear Programming: Relaxing to LP

ILP

$$\begin{aligned} \text{min} \quad & \mathbf{c}^T\mathbf{x} \\ \text{subject} \quad & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq 0, \quad \mathbf{x} \in \mathbb{Z}. \end{aligned}$$



Update

$$y_{ij}^{+} = y_{ij} - \frac{y_{iq}}{y_{pq}} y_{pj}, \quad \text{if } i \neq p$$

$$y_{pj}^{+} = \frac{y_{pj}}{y_{pq}}, \quad \text{if } i = p,$$

$$s_N^T = c_N^T - c_B^T B^{-1} N.$$

LP1

	a_1	a_2	a_3	a_4	a_5	b
	0	1	0	1	1	6
	1	0	0	0	1	3
	0	0	1	-1	1	2
r^T	0	0	0	2	3	15

$\mathcal{B} = \{2, 1, 3\}$, (final)

$\rightarrow$ **solution (valid):** $\mathbf{x} = [3, 6, 2, 0, 0]$.

LP2

	a_1	a_2	a_3	a_4	a_5	b
	1	0	1	0	0	3
	0	0	$\frac{23}{4}$	1	$\frac{3}{4}$	15
	0	1	$\frac{9}{8}$	0	$\frac{1}{8}$	$\frac{7}{2}$
r^T	0	0	$\frac{43}{8}$	0	$\frac{3}{8}$	$\frac{33}{2}$

$\mathcal{B} = \{1, 4, 2\}$, (final)

$\rightarrow$ **solution (fails):** $\mathbf{x} = \left[3, \frac{7}{2}, 0, 15, 0\right]$.

Relaxing ILPs to LPs

- **Suppose we ignore the integer constraint?**
- Often we still obtain integer solutions, **but this can fail!**
- **When can we relax integer constraints?** Still guarantee integer solutions $\mathbf{x}^* \in \mathbb{Z}$?

Modifcations of LPs

- When LP fails, how can we augment the problem to continue search for integer solutions?
- There are many strategies used in practice.

Integer Linear Programming (ILP)

ILP

$$\begin{array}{ll} \text{min} & \mathbf{c}^T\mathbf{x} \\ \text{subject} & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq 0, \quad \mathbf{x} \in \mathbb{Z}. \end{array}$$

Theorem (Cramer's Rule)

For any $m \times m$ matrix B with $\det B \neq 0$ the solution $\mathbf{x} = B^{-1}\mathbf{b}$ can be expressed as

$$x_j = \frac{1}{\det B}\det(\hat{B}_j), \quad \textit{where } \hat{B}_j \textit{ has } j^{th} \textit{ column replaced by } \mathbf{b}.$$

Proof: Let $D_j(\mathbf{v}) = D_j(v_1, v_2, \ldots, v_m) = \det(\hat{B}_j(\mathbf{v}))$, where $\hat{B}_j(\mathbf{v}))$ denotes replacing column j with the vector $\mathbf{v}$. If we take $\mathbf{v} = \mathbf{B}_j$ then $D_j(\mathbf{B}_j) = \det(B)$. Since the determinant is linear in any given column j, we have $D_j(v_1, v_2, \ldots, v_m) = C_{1j}v_1 + \cdots + C_{mj}v_m$.

Another useful property is that if we let $\mathbf{v} = \mathbf{B}_k$ with $k \neq j$ then $D_j(\mathbf{B}_k) = 0$. We consider the linear system $B\mathbf{x} = \mathbf{b}$,

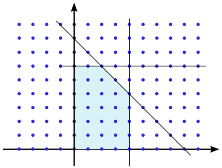
$$\begin{array}{ccccccc} B_{11}x_1 & + & B_{12}x_2 & + \cdots + & B_{1m}x_m & = & b_1 \\ B_{21}x_1 & + & B_{22}x_2 & + \cdots + & B_{2m}x_m & = & b_2 \\ \vdots & & \vdots & & \vdots & & \vdots \\ B_{m1}x_1 & + & B_{m2}x_2 & + \cdots + & B_{mm}x_m & = & b_m \end{array}$$

If we multiply the i^{th} row by C_{ij} and sum, this yields $\sum_k D_j(\mathbf{B}_k)x_k = \det(B)x_j = D_j(\mathbf{b}) = \det(\hat{B}_j)$. ∎

Integer Linear Programming:

- For basic feasible solution we have $\mathbf{x}_B = B^{-1}\mathbf{b}$.
- Consider matrix B and $\mathbf{b}$ with only integer coefficients.
- In this case, $\det(B)$ and $\det(\hat{B}_j)$ are integers.
- If $\det(B) = \pm 1$ then x_j is an integer.

Integer Linear Programming (Unimodular Case)



ILP (Case I)

$$\begin{array}{ll} \text{min} & \mathbf{c}^T\mathbf{x} \\ \text{subject} & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq 0, \quad \mathbf{x} \in \mathbb{Z}. \end{array}$$

ILP (Case II)

$$\begin{array}{ll} \text{min} & \mathbf{c}^T\mathbf{x} \\ \text{subject} & A\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq 0, \quad \mathbf{x} \in \mathbb{Z}. \end{array}$$

Theorem (Cramer's Rule)

For any $m \times m$ matrix B with $\det B \neq 0$ the solution $\mathbf{x} = B^{-1}\mathbf{b}$ can be expressed as

$$x_j = \frac{1}{\det B}\det(\hat{B}_j), \quad \textit{where } \hat{B}_j \textit{ has } j^{th} \textit{ column replaced by } \mathbf{b}.$$

Def: A matrix $A \in \mathbb{Z}^{m\times n}$ is called **unimodular** if **all m^{th}-order minors are ± 1.**

Lemma: If A is unimodular and B is a matrix selecting any m columns of A, then $B\mathbf{x} = \mathbf{b}$ has solutions $\mathbf{x} \in \mathbb{Z}^m$.

Corollary: All basic solutions for the LP have $\mathbf{x}_B = B^{-1}\mathbf{b} \in \mathbb{Z}^m$.

Consequence: For A that is unimodular in Case I, we are guaranteed that the related **LP has only integer solutions.**

Case II: When constraints given as $A\mathbf{x} \leq \mathbf{b}$.

Slack variables $\boldsymbol{\xi}$ put into standard form $A\mathbf{x} + \boldsymbol{\xi} = \mathbf{b}$.

Express as $\tilde{A}\tilde{\mathbf{x}} = \mathbf{b}$, where $\tilde{A} = [A; I]$ and $\tilde{\mathbf{x}} = [\mathbf{x}; \boldsymbol{\xi}]$.

Def: A matrix $A \in \mathbb{Z}^{m\times n}$ is called **totally unimodular** if all minors of any order k are ± 1.

Lemma: If A is totally unimodular, then $\tilde{A}$ is unimodular.

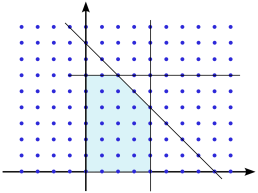
In these cases standard LP can be used to solve the ILP!

Integer Linear Programming (Unimodular Case)

Example

$$\min \; -2x_1 - 5x_2$$

$$\text{subject } x_1 + x_3 = 4$$
$$x_2 + x_4 = 6$$
$$x_1 + x_2 + x_5 = 8$$
$$x_1, x_2, x_3, x_4, x_5 \geq 0$$
$$x_1, x_2, x_3, x_4, x_5 \in \mathbb{Z}$$



Update

$$y_{ij}^{+} = y_{ij} - \frac{y_{iq}}{y_{pq}} y_{pj}, \quad \text{if } i \neq p$$
$$y_{pj}^{+} = \frac{y_{pj}}{y_{pq}}, \quad \text{if } i = p,$$
$$s_N^T = c_N^T - c_B^T B^{-1} N.$$

Canonical Tableau

	a_1	a_2	a_3	a_4	a_5	b
	1	0	1	0	0	4
	0	1	0	1	0	6
	1	1	0	0	1	8
c^T	−2	−5	0	0	0	0

$\mathcal{B} = \{3, 4, 5\}$
(canonicalize)

→

	a_1	a_2	a_3	a_4	a_5	b
	1	0	1	0	0	4
	0	1	0	1	0	6
	1	1	0	0	1	8
r^T	−2	−5	0	0	0	0

$\mathcal{B} = \{3, 4, 5\}, q = 2$
$\frac{y_{i0}}{y_{iq}} = [6, 8], i = [2, 3], p = 2.$

iteration: 1

	a_1	a_2	a_3	a_4	a_5	b
	1	0	1	0	0	4
	0	1	0	1	0	6
	1	0	0	−1	1	2
r^T	−2	0	0	5	0	30

$\mathcal{B} = \{3, 2, 5\}, q = 1$
$\frac{y_{i0}}{y_{iq}} = [4, 2], \; i = [1, 3], \; p = 3$

→

iteration: 2

	a_1	a_2	a_3	a_4	a_5	b
	0	0	1	1	−1	2
	0	1	0	1	0	6
	1	0	0	−1	1	2
r^T	0	0	0	3	2	34

$\mathcal{B} = \{3, 2, 1\}$
(final)

Solution: $\mathbf{x} = [2, 6, 2, 0, 0]$.

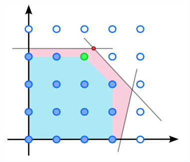
Integer Linear Programming: Gomory Cuts

ILP

min $\quad \mathbf{c}^T\mathbf{x}$

subject $\quad A\mathbf{x} = \mathbf{b}$

$\qquad \mathbf{x} \geq 0, \quad \mathbf{x} \in \mathbb{Z}^n.$



Gomory Cutting-Plane Method

For any feasible solution $\mathbf{x}$

$$x_i + \sum_{j=m+1}^{n} y_{ij}x_j = y_{i0} \;\leftarrow \text{from } [A\mathbf{x} = \mathbf{b}]_i.$$

The **optimal solution $\mathbf{x}^*$ of LP** is given by $x_i^* = y_{i0},\; i \leq m.$

If $\mathbf{x}^*$ is **not an integer** how can we modify the LP to reject this solution, but retain all integer feasible points?

For any feasible point

$$x_i + \sum_{j=m+1}^{n} \lfloor y_{ij} \rfloor x_j \leq y_{i0}.$$

Follows since $\lfloor y_{ij} \rfloor \leq y_{ij}$ and $x_i \geq 0$.

For any **integer** feasible point

$$x_i + \sum_{j=m+1}^{n} \lfloor y_{ij} \rfloor x_j \leq \lfloor y_{i0} \rfloor .$$

For the **non-integer** LP optimal basic point $\mathbf{x}^*$, we have $x_i^* = y_{i0} > \lfloor y_{i0} \rfloor,\; i \leq m$ and $x_i^* = 0,\; i \geq m+1,$ so

$$x_i^* + \sum_{j=m+1}^{n} \lfloor y_{ij} \rfloor x_j^* > \lfloor y_{i0} \rfloor .$$

We introduce the **Gomory Constraint** to the LP

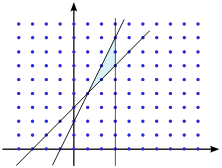
$$\sum_{j=m+1}^{n} \left(y_{ij} - \lfloor y_{ij} \rfloor\right) x_j \geq y_{i0} - \lfloor y_{i0} \rfloor .$$

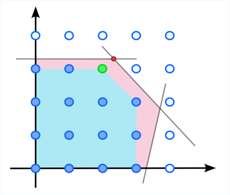
This rejects the $\mathbf{x}^*$ of the LP while preserving all **integer** feasible points $\mathbf{x}$.

In practice, we solve the modified LP with new slack variable x_{n+1} and add constraint to $A\mathbf{x} = \mathbf{b}$,

$$\sum_{j=m+1}^{n} \left(y_{ij} - \lfloor y_{ij} \rfloor\right) x_j - x_{n+1} = y_{i0} - \lfloor y_{i0} \rfloor .$$

Integer Linear Programming: Summary





ILP

$$\begin{array}{ll} \min & \mathbf{c}^T\mathbf{x} \\ \text{subject} & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq 0, \quad \mathbf{x} \in \mathbb{Z}. \end{array}$$

LP Approaches Discussed

- ILPs can in general be challenging to solve.
- How might we use LPs to approach solving ILPs?
- When can we relax the $\mathbf{x} \in \mathbb{Z}$ conditions and still obtain integer solutions? (Unimodularity)
- If this fails, what modifications of LPs can be used to continue search for integer solutions? (Gomory Cuts)

Summary

- ILPs arise in many problems with discrete possibilities.
- Given mathematical structure in many applications the ILPs can be relaxed and solved or approximated by modified LPs.
- Other strategies include: branch-and-bound, other cutting-plane rules, machine learning methods, and heuristic search rules.