Introduction to Linear Programming (LP):

[Source : Tim Roughgarden's lecture notes]

- Mathematical model for optimization of a linear objective subject to linear inequality constraints.

Linear programming is a remarkable sweet spot between power/generality and computational efficiency.

polytime solvable

many interesting problems are obtained as special cases.

- used to solve many problems exactly (e.g. max-flow/min-cut, bipartite matching)

[Total unimodularity, total dual integrality (TDI)]

- can be used to solve NP-hard problems approximately. (e.g. set cover, bin packing)

[Deterministic/randomized rounding :

fractional → integral

- LP Duality gives a refined understanding for many problems.

• Consider simple case when all "$\lesseqgtr$" are "=".

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1$$
$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2$$
$$\vdots$$
$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m$$

Easy: There are exactly 1 or 0 solns.

Gaussian elimination does it in poly time. [$O(n^3)$ arithmetic operations].

– either returns the soln. or correctly reports that no feasible soln. exists.

LP is harder – There can be multiple (infinite) feasible solutions, we need to compute the "best".

• What is LP?

Ingredients of a Linear Program

1. *Decision variables* $x_1, \ldots, x_n \in \mathbb{R}$.
2. *Linear constraints*, each of the form

   constants

$$\sum_{j=1}^{n} a_j x_j \quad (*) \quad b_i,$$

   where (*) could be $\le$, $\ge$, or $=$.

3. A *linear objective function*, of the form

$$\max \sum_{j=1}^{n} c_j x_j$$

   or

$$\min \sum_{j=1}^{n} c_j x_j.$$

Example

$$\max x_1 + x_2$$

$$4x_1 + x_2 \le 2$$
$$x_1 + 2x_2 \le 1$$
$$x_1 \ge 0$$
$$x_2 \ge 0.$$

Not allowed:
x_j^2, $x_j x_k$, $\log(1 + x_j)$.

$a = b \iff \begin{matrix} a \geq b \\ a \leq b \end{matrix} \iff \begin{matrix} a \leq b \\ -a \leq -b \end{matrix}$

$a \geq b \iff -a \leq -b$.

$\max \sum c_j x_j \iff \min -\sum c_j x_j$

$a \geq b$
$\iff a = b + c,\ c \geq 0$

• A closer view of the example.

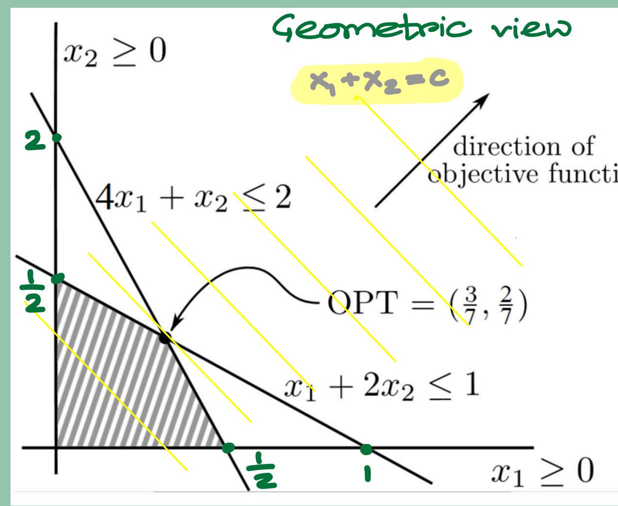
Algebraic view

objective $\max x_1 + x_2$ (1)

subject to:

constraints
$$\begin{cases} 4x_1 + x_2 \leq 2 & (2) \\ x_1 + 2x_2 \leq 1 & (3) \\ x_1 \geq 0 & (4) \\ x_2 \geq 0. & (5) \end{cases}$$

Geometric view



1. A linear constraint in n dimensions corresponds to a halfspace in $\mathbb{R}^n$. Thus a feasible region is an intersection of halfspaces, the higher-dimensional analog of a polygon.[3]

2. The level sets of the objective function are parallel $(n-1)$-dimensional hyperplanes in $\mathbb{R}^n$, each orthogonal to the coefficient vector $\mathbf{c}$ of the objective function.

3. The optimal solution is the feasible point furthest in the direction of $\mathbf{c}$ (for a maximization problem) or $-\mathbf{c}$ (for a minimization problem). Equivalently, it is the last point of intersection (traveling in the direction $\mathbf{c}$ or $-\mathbf{c}$) of a level set of the objective function and the feasible region.

4. When there is a unique optimal solution, it is a vertex (i.e., "corner") of the feasible region.

constraint $\Rightarrow$ halfspace.

Feasible regions $\Rightarrow$ polytope / polyhedron

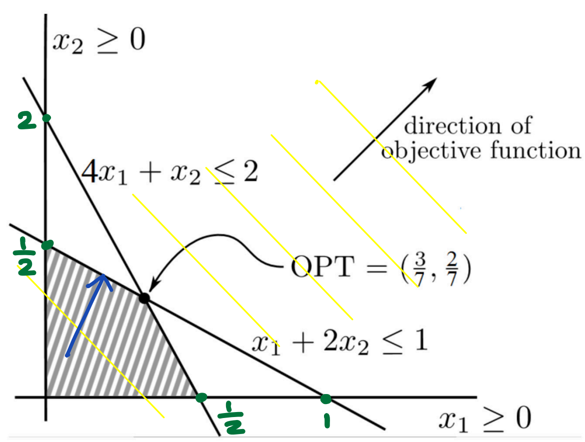
OPT $\Rightarrow$ vertex

Edge cases:

1. There might be no feasible solutions at all. For example, if we add the constraint $x_1 + x_2 \geq 1$ to our toy example, then there are no longer any feasible solutions. Linear programming algorithms correctly detect when this case occurs.

2. The optimal objective function value is unbounded ($+\infty$ for a maximization problem, $-\infty$ for a minimization problem). Note a necessary but not sufficient condition for this case is that the feasible region is unbounded. For example, if we dropped the constraints $2x_1 + x_2 \leq 1$ and $x_1 + 2x_2 \leq 1$ from our toy example, then it would have unbounded objective function value. Again, linear programming algorithms correctly detect when this case occurs.

3. The optimal solution need not be unique, as a "side" of the feasible region might be parallel to the levels sets of the objective function. Whenever the feasible region is bounded, however, there always exists an optimal solution that is a vertex of the feasible region.[4]



- Towards LP-Duality:

objective $\max x_1 + x_2$ (1)

subject to:

constraints
$$4x_1 + x_2 \leq 2 \quad (2)$$
$$x_1 + 2x_2 \leq 1 \quad (3)$$
$$x_1 \geq 0 \quad (4)$$
$$x_2 \geq 0. \quad (5)$$

From geometric viewpoint:

$\text{OPT} = \frac{5}{7}$ for $x_1 = \frac{3}{7}$, $x_2 = \frac{2}{7}$.

But how do we know that it is optimal?

Show obj $\leq 5/7$.

Attempt 1:
$$x_1 + x_2 \leq 4x_1 + x_2 \leq 2 \quad \text{[Using (4) \& (2)]}$$
$$x_1 + x_2 \leq x_1 + 2x_2 \leq 1 \quad \text{[using (3) \& (5)]}$$

Can we do better?

Attempt 2:
$$x_1 + x_2 \leq \frac{1}{7}(4x_1 + x_2) + \frac{3}{7}(x_1 + 2x_2) \underset{\text{[Using (2) \& (3)]}}{\leq} \frac{1}{7} \cdot 2 + \frac{3}{7} \cdot 1 = \frac{5}{7}. \; ↯$$

Q. How do we find such values $1/7$ & $3/7$?

→ via LP duality.

Standard linear program: Primal LP (P)

$$\max \sum_{j=1}^{n} c_j x_j \quad \to \text{obj.} \tag{7}$$

subject to

$$\sum_{j=1}^{n} a_{1j} x_j \leq b_1 \tag{8}$$
$$\sum_{j=1}^{n} a_{2j} x_j \leq b_2 \tag{9}$$
$$\vdots \leq \vdots \tag{10}$$
$$\sum_{j=1}^{n} a_{mj} x_j \leq b_m \tag{11}$$
$$x_1, \dots, x_n \geq 0. \tag{12}$$

Matrix-vector notation:

$$\max \underset{n\times 1}{\mathbf{c}}^T \underset{n \times 1}{\mathbf{x}}$$

subject to:

$$\underset{m\times n}{\mathbf{A}} \underset{n\times 1}{\mathbf{x}} \leq \underset{m\times 1}{\mathbf{b}}$$
$$\mathbf{x} \geq 0,$$

This linear program has n nonnegative decision variables $x_1, \dots, x_n$ and m constraints (not counting the nonnegativity constraints). The a_{ij}'s, b_i's, and c_j's are all part of the input (i.e., fixed constants).[1]

Goal: Derive upper bound on obj.

Approach: Take nonnegative linear combination of the constraints that (componentwise) dominates obj.

$\Rightarrow$ Find $y_1, \dots, y_m \geq 0$ s.t.

$\sum_{i=1}^{m} y_i a_{ij} \geq c_j \quad \forall j \in [n]$

matrix notation:

$A^T y \geq c$

$\downarrow$ nonneg m-dim vector

Then for every feasible soln $(x_1, \dots, x_n)$ of (P):

$$\underbrace{\sum_{j=1}^{n} c_j x_j}_{\text{x's obj fn}} \leq \sum_{j=1}^{n} \left(\sum_{i=1}^{m} y_i a_{ij} \right) x_j \tag{14}$$

Reversing the order of sum

$$= \sum_{i=1}^{m} y_i \cdot \left(\sum_{j=1}^{n} a_{ij} x_j \right) \tag{15}$$

constraints of (P) + nonnegativity of y_i's.

$$\leq \underbrace{\sum_{i=1}^{m} y_i b_i}_{\text{upper bound}} . \tag{16}$$

In matrix-vector notation:

$c^T x \leq (A^T y)^T x$

$= y^T (Ax) \leq y^T b$

- Upshot: OPT of $(P) \leq \sum_{i=1}^{m} b_i y_i$.

Here's the key point: *the tightest upper bound on OPT is itself the optimal solution to a linear program.* Namely:

$$\min \sum_{i=1}^{m} b_i y_i$$

Dual program

subject to

$$\sum_{i=1}^{m} a_{i1} y_i \geq c_1$$
$$\sum_{i=1}^{m} a_{i2} y_i \geq c_2$$
$$\vdots \geq \vdots$$
$$\sum_{i=1}^{m} a_{in} y_i \geq c_n$$
$$y_1, \dots, y_m \geq 0.$$

Or, in matrix-vector form: (D)

$$\min \mathbf{b}^T \mathbf{y}$$

subject to

$$\mathbf{A}^T \mathbf{y} \geq \mathbf{c}$$
$$\mathbf{y} \geq 0.$$

This linear program is called the *dual* to (P), and we sometimes denote it by (D).

Primal

objective $\max x_1 + x_2$ (1)

subject to:

constraints

$$4x_1 + x_2 \leq 2 \quad (2)$$
$$x_1 + 2x_2 \leq 1 \quad (3)$$
$$x_1 \geq 0 \quad (4)$$
$$x_2 \geq 0. \quad (5)$$

$\max \mathbf{c}^T\mathbf{x}$

(P)

$\mathbf{Ax} \leq \mathbf{b}$

$\mathbf{x} \geq 0,$

$A = \begin{bmatrix} 4 & 1 \\ 1 & 2 \end{bmatrix}$

$A^T = \begin{bmatrix} 4 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$

Dual

min $2y_1 + y_2$

subject to:

$$4y_1 + y_2 \geqslant 1$$
$$y_1 + 2y_2 \geqslant 1$$
$$y_1, y_2 \geqslant 0$$

Recipe for conversion:

Primal	Dual
variables $x_1, \ldots, x_n$	n constraints
m constraints	variables $y_1, \ldots, y_m$
objective function $\mathbf{c}$	right-hand side $\mathbf{c}$
right-hand side $\mathbf{b}$	objective function $\mathbf{b}$
$\max \mathbf{c}^T\mathbf{x}$	$\min \mathbf{b}^T\mathbf{y}$
constraint matrix $\mathbf{A}$	constraint matrix $\mathbf{A}^T$
ith constraint is "$\leq$"	$y_i \geq 0$
ith constraint is "$\geq$"	$y_i \leq 0$
ith constraint is "$=$"	$y_i \in \mathbb{R}$
$x_j \geq 0$	jth constraint is "$\geq$"
$x_j \leq 0$	jth constraint is "$\leq$"
$x_j \in \mathbb{R}$	jth constraint is "$=$"

Dual of Dual

= Primal

Theorem 5.1 (Weak Duality) *For every maximization linear program (P) and corresponding dual linear program (D),*

$$\text{OPT value for (P)} \le \text{OPT value for (D)};$$

for every minimization linear program (P) and corresponding dual linear program (D),

$$\text{OPT value for (P)} \ge \text{OPT value for (D)}.$$

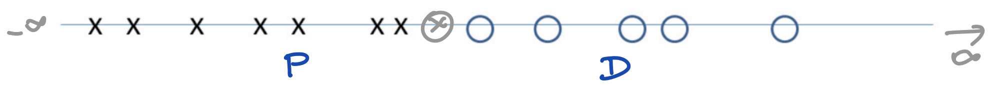


Figure 3: visualization of weak duality. X represents feasible solutions for P while O represents feasible solutions for D.

Weak duality already has some very interesting corollaries.

Corollary 5.2 *Let (P),(D) be a primal-dual pair of linear programs.*

(a) *If the optimal objective function value of (P) is unbounded, then (D) is infeasible.*

(b) *If the optimal objective function value of (D) is unbounded, then (P) is infeasible.*

(c) *If* $\mathbf{x}, \mathbf{y}$ *are feasible for (P),(D) and* $\mathbf{c}^T\mathbf{x} = \mathbf{y}^T\mathbf{b}$*, then both* $\mathbf{x}$ *and* $\mathbf{y}$ *are both optimal.*

$$\begin{array}{lll} \text{maximize} & \displaystyle\sum_{j=1}^{n} c_j x_j & \\ \text{subject to} & \displaystyle\sum_{j=1}^{n} a_{ij} x_j \le b_i, & i = 1, \dots, m \\ & x_j \ge 0, & j = 1, \dots, n \end{array}$$

$$\begin{array}{lll} \text{minimize} & \displaystyle\sum_{i=1}^{m} b_i y_i & \\ \text{subject to} & \displaystyle\sum_{i=1}^{m} a_{ij} y_i \ge c_j, & j = 1, \dots, n \\ & y_i \ge 0, & i = 1, \dots, m \end{array}$$

2 Complementary Slackness Conditions

2.1 The Conditions

Next is a corollary of ~~Corollary 1.1.~~ Theorem 5.1 It is another sufficient (and as we'll see later, necessary) condition for optimality.

Corollary 2.1 (Complementary Slackness Conditions) *Let (P),(D) be a primal-dual pair of linear programs. If* $\mathbf{x}, \mathbf{y}$ *are feasible solutions to (P),(D), and the following two conditions hold then both* $\mathbf{x}$ *and* $\mathbf{y}$ *are both optimal.*

(1) *Whenever* $x_j \neq 0$, $\mathbf{y}$ *satisfies the* j*th constraint of (D) with equality.*

(2) *Whenever* $y_i \neq 0$, $\mathbf{x}$ *satisfies the* i*th constraint of (P) with equality.*

The conditions assert that no decision variable and corresponding constraint are simultaneously "slack" (i.e., it forbids that the decision variable is not 0 and also the constraint is not tight).

Proof of Corollary 2.1: We prove the corollary for the case of primal and dual programs of the form (P) and (D) in Section 1; the other cases are all the same.

The first condition implies that

$$\text{Primal} = \sum_{j=1}^{n} c_j x_j = \sum_{j=1}^{n} \left(\sum_{i=1}^{m} y_i a_{ij} \right) x_j$$

for each $j = 1, \dots, n$ (either $x_j = 0$ or $c_j = \sum_{i=1}^{m} y_i a_{ij}$). Hence, inequality (1) holds with equality. Similarly, the second condition implies that

$$\sum_{i=1}^{m} y_i \left(\sum_{j=1}^{n} a_{ij} x_i \right) = \sum_{i=1}^{m} y_i b_i = \text{Dual}$$

for each $i = 1, \dots, m$. Hence inequality (3) also holds with equality. Thus $\mathbf{c}^T\mathbf{x} = \mathbf{y}^T\mathbf{b}$, and Corollary 1.1 implies that both $\mathbf{x}$ and $\mathbf{y}$ are optimal. ■

2.2 Physical Interpretation

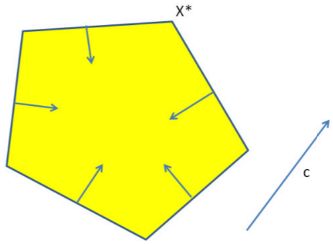


Figure 2: Physical interpretation of complementary slackness. The objective function pushes a particle in the direction c until it rests at x^*. Walls also exert a force on the particle, and complementary slackness asserts that only walls touching the particle exert a force, and sum of forces is equal to 0.

We offer the following informal physical metaphor for the complementary slackness conditions, which some students find helpful (Figure 2). For a linear program of the form (P) in Section 1, think of the objective function as exerting "force" in the direction $\mathbf{c}$. This pushes a particle in the direction $\mathbf{c}$ (within the feasible region) until it cannot move any further in this direction. When the particle comes to rest at position $\mathbf{x}^*$, the sum of the forces acting on it must sum to 0. What else exerts force on the particle? The "walls" of the feasible region, corresponding to the constraints. The direction of the force exerted by the ith constraint of the form $\sum_{j=1}^n a_{ij}x_j \leq b_i$ is perpendicular to the wall, that is, $-\mathbf{a}_i$, where $\mathbf{a}_i$ is the ith row of the constraint matrix. We can interpret the corresponding dual variable y_i as the magnitude of the force exerted in this direction $-\mathbf{a}_i$. The assertion that the sum of the forces equals 0 corresponds to the equation $\mathbf{c} = \sum_{i=1}^n y_i \mathbf{a}_i$. The complementary slackness conditions assert that $y_i > 0$ only when $\mathbf{a}_i^T \mathbf{x} = b_i$ — that is, only the walls that the particle touches are allowed to exert force on it.

Theorem 4.1 (Strong LP Duality) *When a primal-dual pair (P),(D) of linear programs are both feasible,*

$$\text{OPT for (P)} = \text{OPT for (D)}.$$

Corollary 4.2 (LP Optimality Conditions) *Let* $\mathbf{x}, \mathbf{y}$ *are feasible solutions to the primal-dual pair (P),(D) be a = primal-dual pair, then*

$\mathbf{x}, \mathbf{y}$ *are both optimal* *if and only if* $\mathbf{c}^T\mathbf{x} = \mathbf{y}^T\mathbf{b}$

if and only if *the complementary slackness conditions hold.*

Follows from strong duality

Follows from compl. slackness conditions.

Theorem 4.4 (Farkas's Lemma) *Given a matrix* $\mathbf{A} \in \mathbb{R}^{m \times n}$ *and a right-hand side* $\mathbf{b} \in \mathbb{R}^m$, *exactly one of the following holds:*

(i) There exists $\mathbf{x} \in \mathbb{R}^n$ *such that* $\mathbf{x} \geq 0$ *and* $\mathbf{A}\mathbf{x} = \mathbf{b}$;

(ii) There exists $\mathbf{y} \in \mathbb{R}^m$ *such that* $\mathbf{y}^T\mathbf{A} \geq 0$ *and* $\mathbf{y}^T\mathbf{b} < 0$.

Existence of y imply Ax = b is infeasible.

- Application / example of LP Duality :

Max-flow in a network:

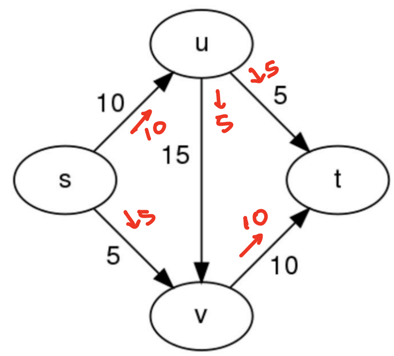
Given :

Directed graph $G=(V,E)$, source $s\in V$, sink $t\in V$.

+ve arc capacities $c: E\to\mathbb{R}^+$.

Goal :

Find the max flow that can be sent from s to t subject to :



A flow network, with source *s* and sink *t*. The numbers next to the edge are the capacities.

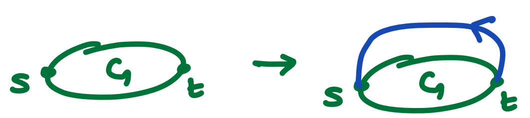
1. Capacity constraints :

For each arc e, the flow sent through e (f_e) is bounded by its capacity, $\boxed{f_e \le c_e}$

2. Flow conservation : $\forall v \in V \setminus \{s,t\}$,

total flow into v = total flow out of v.

- Formulate as an LP :



Flow → Conservation

Add a fictitious arc of infinite capacity from t to s.

⇒ flow conservation @ s,t too.

$$
\begin{array}{lll}
\text{maximize} & f_{ts} & \\
\text{subject to} & f_{ij} \le c_{ij}, & (i,j)\in E \\
& \displaystyle\sum_{j:\ (j,i)\in E} f_{ji} - \sum_{j:\ (i,j)\in E} f_{ij} \le 0, & i \in V \\
& f_{ij} \ge 0, & (i,j)\in E
\end{array}
$$

d_{ij} cap. cons.

p_i

flow cons. in fact, holds by '=' as deficit in flow balance in one node imply surplus in flow balance in another

$$\max \quad 1\cdot f_{ts} + \sum_{e\in E} 0\cdot f_e \qquad m = |E|+1. \quad n = |E| + |V|.$$

s.t.

$$1\cdot f_{ij} + \sum_{e\in E\setminus ij} 0\cdot f_e \le c_{ij} \quad \forall\, ij \in E \qquad \leftarrow d_{ij}$$

$$\sum_{\substack{j:(j,i)\\ \in E}} 1\cdot f_{ij} + \sum_{\substack{j:(i,j)\\ \in E}} (-1)\cdot f_{ij} \le 0 \quad \forall\, i \in V. \qquad \leftarrow p_i$$

incoming edges; outgoing edges

$$f_{ij} \geq 0.$$

maximize $\sum_{i=1}^{m} b_i y_i$

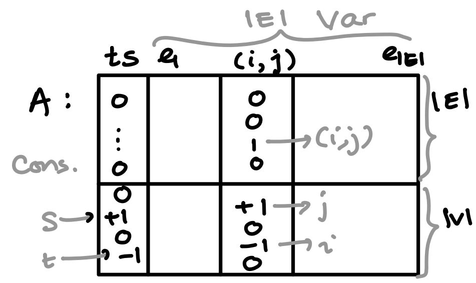
subject to $\sum_{i=1}^{m} a_{ij} y_i \le c_j, \quad j = 1,\dots,n$

$y_i \ge 0, \quad i = 1,\dots,m$

minimize $\sum_{j=1}^{n} c_j x_j$

subject to $\sum_{j=1}^{n} a_{ij} x_j \ge b_i, \quad i = 1,\dots,m$

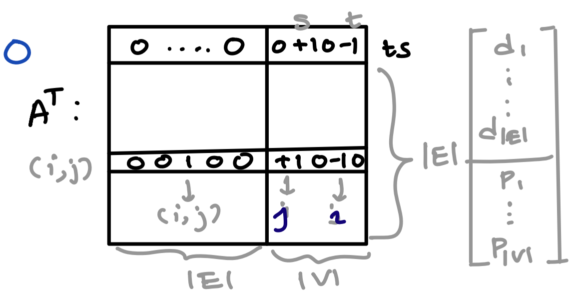
$x_j \ge 0, \quad j = 1,\dots,n$



$$\min \quad \sum_{i\in V} 0\cdot p_i + \sum_{ij\in E} c_{ij}\cdot d_{ij}$$

s.t. (for ts) $\quad 1\cdot p_s + (-1)\cdot p_t \geq 1$

(for $E\setminus ts$) $\quad d_{ij} + p_j - p_i \geq 0$



Dual :

$$\text{minimize} \quad \sum_{(i,j)\in E} c_{ij} d_{ij}$$

subject to

$d_{ij} - p_i + p_j \geq 0, \quad (i,j) \in E$ — *

$p_s - p_t \geq 1$ — **

$d_{ij} \geq 0 \quad (i,j) \in E$

$p_i \geq 0 \quad i \in V$

- Intuitive understanding of the dual program.

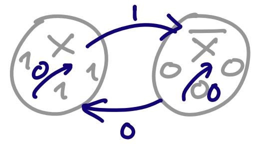
For now consider $d_{ij} \in \{0,1\}$ & $p_i \in \{0,1\}$.

$d_{ij} \rightarrow$ distance label on arcs.

$p_i \rightarrow$ potentials on nodes.

Let (d^*, p^*) be an opt soln of this integer program.

Then, $P_s^* - P_t^* \geq 1 \Rightarrow P_s^* = 1,\ P_t^* = 0.$



This defines an s-t cut $(X, \bar{X})$, where

X is set of potential 1 nodes.

$\bar{X}$ is set of potential 0 nodes.

Consider an arc (i,j) with $i \in X, j \in \bar{X}$.

then, $d_{ij} \geq P_i - P_j = 1 \Rightarrow d_{ij} = 1.$

For arc (i,j) with $i \in X, j \in X$ or $i \in \bar{X}, j \in \bar{X}$ or $i \in \bar{X}, j \in X$,

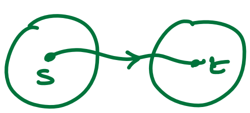
$d_{ij} \geq 0$, thus can be set 1 or 0.

To minimize the objective, we should set them 0.

Thus, objective of dual = min s-t cut

[i.e., partition of V into $[X, \bar{X}]$ s.t. number of arcs going from X to $\bar{X}$ is minimized].

Observation:



Any path from s to t in G contains at least one edge of C.

So dual can be interpreted as fractional s-t cut:

The distance labels assigned to arcs by the dual satisfy the property that distance labels on any s-t path $(s = v_0, v_1, \ldots, v_k = t)$ sum to ≥ 1.

$$\sum_{i=0}^{k-1} d_{v_i v_{i+1}} \overset{(*)}{\geq} \sum_{i=0}^{k-1} (P_{v_i} - P_{v_{i+1}}) = P_s - P_t \overset{(**)}{\geq} 1$$

- Relating max-flow & min-cut.

Due to capacity constraints, capacity of any s-t cut is an upper bound on any feasible flow.

i.e., max-flow $\leq$ min-cut

This is also evident from weak LP-duality.

max-flow (primal LP) $\leq$ dual LP $\leq$ integer program for mincut

$\downarrow$ weak LP-duality

$\downarrow$ LP relaxation (Note: upper bounds are redundant)

Surprisingly, we have stronger property.

max-flow (primal LP) $=$ dual LP $=$ integer program for mincut

$\downarrow$ strong LP-duality

$\downarrow$ integral polyhedron

$\Rightarrow$ Max-flow min cut theorem.

- How to show integrality of LP:

A **totally unimodular matrix**[1] (TU matrix) is a matrix for which every square non-singular submatrix is unimodular. Equivalently, every square submatrix has determinant 0, +1 or −1. A totally unimodular matrix need not be square itself. From the definition it follows that any submatrix of a totally unimodular matrix is itself totally unimodular (TU). Furthermore it follows that any TU matrix has only 0, +1 or −1 entries. The converse is not true, i.e., a matrix with only 0, +1 or −1 entries is not necessarily unimodular. A matrix is TU if and only if its transpose is TU.

Totally unimodular matrices are extremely important in polyhedral combinatorics and combinatorial optimization since they give a quick way to verify that a linear program is integral (has an integral optimum, when any optimum exists). Specifically, if A is TU and b is integral, then linear programs of forms like $\{\min cx \mid Ax \geq b, x \geq 0\}$ or $\{\max cx \mid Ax \leq b\}$ have integral optima, for any c. Hence if A is totally unimodular and b is integral, every extreme point of the feasible region (e.g. $\{x \mid Ax \geq b\}$) is integral and thus the feasible region is an integral polyhedron.

In mathematical optimization, **total dual integrality** is a sufficient condition for the integrality of a polyhedron. Thus, the optimization of a linear objective over the integral points of such a polyhedron can be done using techniques from linear programming.

A linear system $Ax \leq b$, where A and b are rational, is called totally dual integral (TDI) if for any $c \in \mathbb{Z}^n$ such that there is a feasible, bounded solution to the linear program

$$\begin{aligned} \max c^{\mathrm{T}} x \\ Ax \leq b, \end{aligned}$$

there is an integer optimal dual solution.[1][2][3]

Edmonds and Giles[2] showed that if a polyhedron P is the solution set of a TDI system $Ax \leq b$, where b has all integer entries, then every vertex of P is integer-valued. Thus, if a linear program as above is solved by the simplex algorithm, the optimal solution returned will be integer. Further, Giles and Pulleyblank[1] showed that if P is a polytope whose vertices are all integer valued, then P is the solution set of some TDI system $Ax \leq b$, where b is integer valued.

Note that TDI is a weaker sufficient condition for integrality than total unimodularity.[4]

- Understanding complementary slackness:

- Let f^* be opt primal soln (max flow).
  (d^*, p^*) be opt dual soln (min cut: defined by $X, \bar{X}$).

- Say arc (i,j) has $i \in X, j \in \bar{X}$.
  then $d^*_{ij} = 1$ i.e. $d^*_{ij} \neq 0$
  Then c.s. $\Rightarrow f^*_{ij} = c_{ij}$.

- Now say arc (k,l) has $k \in \bar{X}, l \in X$.
  then $p^*_k - p^*_l = -1$ and $d^*_{kl} \in \{0,1\}$.
  Hence, $d^*_{kl} - p^*_k + p^*_l \geq 0$ must be strict inequality
  So, c.s. $\Rightarrow f^*_{kl} = 0$.

$\Rightarrow$ Arcs $X \to \bar{X}$ are saturated by f^*.
Arcs $\bar{X} \to X$ carry no flow.

$\Rightarrow$ max-flow $=$ min-cut.

For max-flow we consider an alternate LP based on path decomposition.

Advantage: no need to explicitly state conservation constraints. Will still have capacity & nonneg constraints.

Let $\mathcal{P}$ denote the set of all s-t paths.

One can show following two LPs are equivalent:

$$\max \sum_{P \in \mathcal{P}} f_P \tag{18}$$

subject to

$$\underbrace{\sum_{P \in \mathcal{P} : e \in P} f_P}_{\text{total flow on } e} \leq u_e \qquad \text{for all } e \in E \xleftarrow{} \ell_e \tag{19}$$

$$f_P \geq 0 \qquad \text{for all } P \in \mathcal{P}. \tag{20}$$

$$\begin{array}{lll}
\text{maximize} & f_{ts} & \\
\text{subject to} & f_{ij} \leq c_{ij}, & (i,j) \in E \\
& \sum_{j:\ (j,i) \in E} f_{ji} - \sum_{j:\ (i,j) \in E} f_{ij} \leq 0, & i \in V \\
& f_{ij} \geq 0, & (i,j) \in E
\end{array}$$

Dual:

$$\min \sum_{e \in E} u_e \ell_e$$

subject to

$$\sum_{e \in P} \ell_e \geq 1 \qquad \text{for all } P \in \mathcal{P} \tag{21}$$

$$\ell_e \geq 0 \qquad \text{for all } e \in E.$$

Again we can show dual corresponds to mincut.

For a fix cut $(X, \overline{X})$ with $s \in X$, $y \in \overline{X}$ set

$$\ell_{ij} = \begin{cases} 1 & \text{if } i \in X, j \in \overline{X} \\ 0 & \text{else} \end{cases}$$

Every s-t path has at least one edge in cut $[X, \overline{X}]$.

$\Rightarrow \ell_{ij} = 1 \Rightarrow (21)$ holds.

Objective value $= \sum_{e \in E} u_e \ell_e = \sum_{\substack{i \in X \\ j \in \overline{X}}} u_{ij} = \text{cut}[X, \overline{X}]$.

Hence, again

$$\underset{\text{(primal LP)}}{\text{max-flow}} \underset{\substack{\text{strong} \\ \text{LP-duality}}}{=} \text{dual LP} \underset{\substack{\text{integral} \\ \text{polyhedron}}}{=} \underset{\text{for mincut}}{\text{integer program}}$$

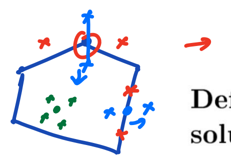
- Many other interesting theorems/algorithms can be viewed as consequence of LP duality

  - Minimax theorem.
  - Hungarian algorithms.

  $\Big\}$ see TR notes

• Properties of extreme point solutions :

[Lau-Ravi-Singh, Ch 2]

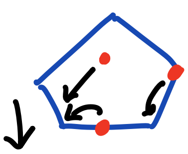


Definition 1.2.1 *Let $P = \{x : Ax = b, x \geq 0\} \subseteq \mathbb{R}^n$. Then $x \in \mathbb{R}^n$ is an* **extreme point solution** *of P if there does not exist a non-zero vector $y \in \mathbb{R}^n$ such that $x + y, x - y \in P$.*

— also known as vertex soln / basic feasible soln.

Definition 2.1.1 *Let P be a polytope and let x be an extreme point solution of P then x is* ***integral*** *if each co-ordinate of x is an integer. The polytope P is called* ***integral*** *if every extreme point of P is integral.*

LP relaxation is exact.



Lemma 2.1.2 *Let $P = \{x : Ax \geq b, x \geq 0\}$ and assume that $\min\{c^T x : x \in P\}$ is finite. Then for every $x \in P$, there exists an extreme point solution $x' \in P$ such that $c^T x' \leq c^T x$, i.e., there is always an extreme point optimal solution.*

Basic feasible solution :

Consider the linear program

$$\begin{array}{lrcl} \text{minimize} & c^T x & & \\ \text{subject to} & Ax & \geq & b \\ & x & \geq & 0 \end{array}$$

By introducing slack variables s_j for each constraint, we obtain an equivalent linear program in *standard form*.

$$\begin{array}{lrcl} \text{minimize} & c^T x & & \\ \text{subject to} & Ax + s & = & b \\ & x & \geq & 0 \\ & s & \geq & 0 \end{array}$$

Henceforth, we study linear program in standard form: $\{\min cx : Ax = b, x \geq 0\}$. Without loss of generality, we can assume that A is of full row rank. If there are dependent constraints, we can remove them without affecting the system or its optimal solution.

A subset of columns B of the constraint matrix A is called a *basis* if the matrix of columns corresponding to B, i.e. A_B, is invertible. A solution x is called *basic* if and only if there is a basis B such that $x_j = 0$ if $j \notin B$ and $x_B = A_B^{-1}b$. If in addition to being basic, it is also feasible, i.e., $A_B^{-1}b \geq 0$, it is called a *basic feasible solution* for short. The correspondence between bases and basic feasible solutions is not one to one. Indeed there can be many bases which correspond to the same basic feasible solution. The next theorem shows the equivalence of extreme point solutions and basic feasible solutions.

eqn(hyperplane) var(dim)

Theorem 2.1.5 *Let A be a $m \times n$ matrix with full row rank. Then every feasible x to $P = \{x : Ax = b, x \geq 0\}$ is a basic feasible solution if and only if x is an extreme point solution.*

Basic solution : A_B is invertible. rank = rank(A) = m.

Nonbasic variable

So, put (n − m) variables to 0,
Other m variables are called basic variables.
Resulting system is $A_B x_B = b$ & if A_B is invertible.
its solution is a basic soln. (BS)

If all variables in BS are ≥ 0, then it is
Basic feasible solution (BFS).
Otherwise, it is called infeasible soln.

If some basic variables are 0 in BFS → degenerate
" all " " are >0 " → non-degenerate

Thm 2.1.5 : Basic feasible soln = extreme point soln = vertex soln.

Lem 2.1.2 : ∃ an extreme point optimal soln.

⇒ Optimal soln uses a basic feasible solution.

\# Vertices (extreme points) $\leq$ BFS $\leq$ BS $\leq$ ${}^nC_m$.

Problem. Write all basic solution to the following system and find optimal solution. Maximize $Z = 2x_1 + 3x_2$

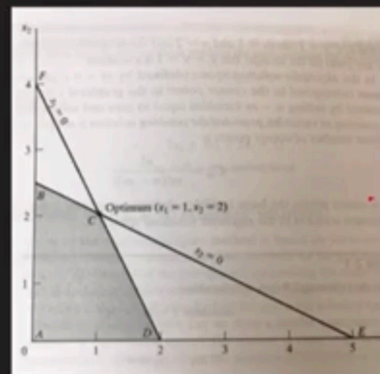
$$2x_1 + x_2 \le 4$$
$$x_1 + 2x_2 \le 5 \quad \text{and} \quad x_1,\ x_2 \ge 0$$

Solution. Convert inequalities into equality by adding slack/surplus variable

$$2x_1 + x_2 + s_1 = 4$$
$$x_1 + 2x_2 + s_2 = 5 \quad \text{and} \quad x_1,\ x_2,\ s_1,\ s_2 \ge 0$$

Non Basic Variables	Basic Variables	Basic solution	Associated corner point	Feasibility	Objective Value
(x_1, x_2)	(s_1, s_2)	(4, 5)	A	Yes	0
(x_1, s_1)	(x_2, s_2)	(4, -3)	F	No	-
(x_1, s_2)	(x_2, s_1)	(2.5, 1.5)	B	Yes	7.5
(x_2, s_1)	(x_1, s_2)	(2, 3)	D	Yes	4
(x_2, s_2)	(x_1, s_1)	(5, -6)	E	No	-
(s_1, s_2)	(x_1, x_2)	(1, 2)	C	Yes	8 (Optimum)

Graphical and algebraic solution connection



$$\max \quad 2x_1 + 3x_2$$

s.t. $\quad 2x_1 + x_2 \le 4$ ✓

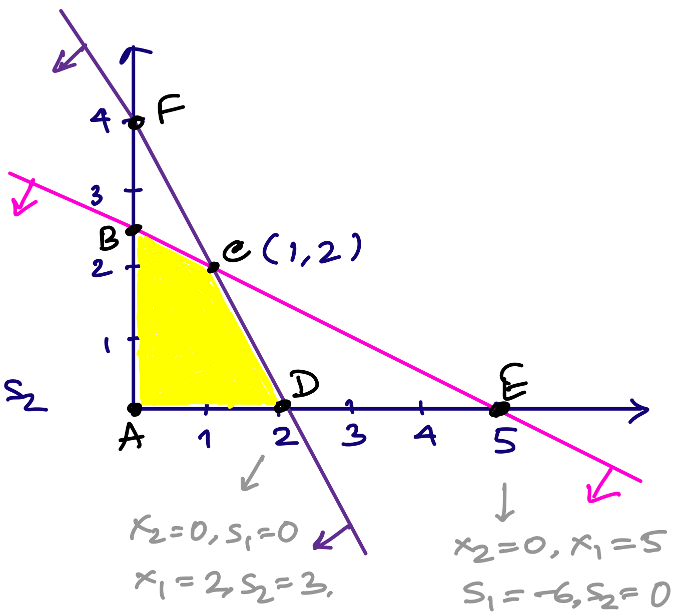
$\quad x_1 + 2x_2 \le 5$ ✓

$\quad x_1, x_2 \geq 0$

$\Updownarrow$

$$\max \quad 2x_1 + 3x_2 + 0s_1 + 0s_2$$

$$2x_1 + x_2 + s_1 = 4$$
$$x_1 + 2x_2 + s_2 = 5$$
$$x_1, x_2, s_1, s_2 \geq 0$$



$$A: \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & s_1 & s_2 \end{array} \\ \begin{bmatrix} 2 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \end{bmatrix} \end{array}$$

OPT = a BFS

No degen BFS.

Vertices (extreme points) $\le$ BFS $\underset{<}{\le}$ BS $\le$ ${}^nC_m$.

4 Corners $\quad$ 4 BFS $\quad$ 6 BS $\quad$ $\binom{4}{2} = 6$

Example 2:

$$\max \; 2x_1 + x_2$$
$$x_1 + x_2 \le 3 \quad \checkmark$$
$$x_1 - x_2 \le 0 \quad \checkmark$$
$$x_1, x_2 \ge 0$$

$$\Downarrow$$

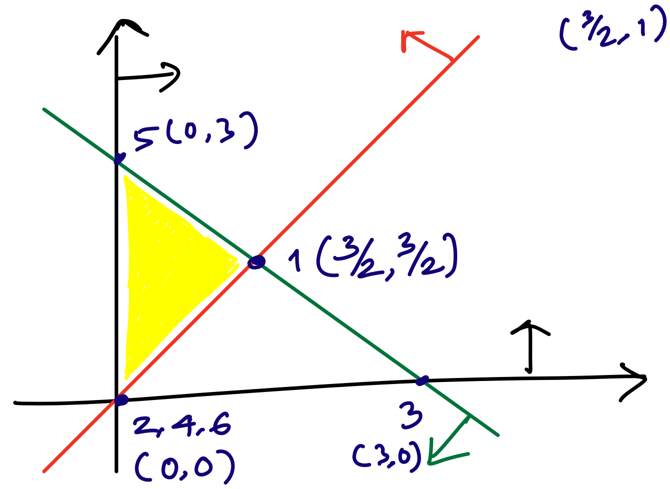
$$\max \; 2x_1 + x_2$$
$$x_1 + x_2 + x_3 = 3$$
$$x_1 - x_2 + x_4 = 0$$
$$x_1, x_2, x_3, x_4 \ge 0.$$

$$\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \\ \end{array}$$
$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{bmatrix}$$

$\checkmark 1 \quad 1$

$\checkmark 1 \quad -1$

$(3/2, 1)$



Soln:

	Basic Variable	Nonbasic Var.	Soln. $x_1\ x_2\ x_3\ x_4$			
1	x_1, x_2	$x_3 = x_4 = 0$	$(3/2, 3/2, 0, 0)$	BS	BFS	nondeg.
2	x_1, x_3	$x_2 = x_4 = 0$	$(0, 0, 3, 0)$	BS	BFS	deg.
3	x_1, x_4	$x_2 = x_3 = 0$	$(3, 0, 0, -3)$	BS	$\times$	
4	x_2, x_3	$x_1 = x_4 = 0$	$(0, 0, 3, 0)$	BS	BFS	deg
5	x_2, x_4	$x_1 = x_3 = 0$	$(0, 3, 0, 3)$	BS	BFS	nondeg
6	x_3, x_4	$x_1 = x_2 = 0$	$(0, 0, 3, 0)$	BS	BFS	deg.

Vertices (extreme points) $\le$ BFS $\overset{<}{\le}$ BS $\le$ ${}^nC_m$.

3 Corners $\quad$ 5 BFS $\quad$ 6 BS $\quad$ $\binom{4}{2} = 6$

For nondeg. BFS there is one to one correspondence between BFS & vertex.

But not for deg. BFS.

At any BFS there are n lin indep tight constraints.

Problem. Determine the optimum solution for following LP by enumerating all the basic solution.

$$\text{Maximize } Z = 2x_1 - 4x_2 + 5x_3 - 6x_4$$
$$x_1 + 4x_2 - 2x_3 + 8x_4 \le 2$$
$$-x_1 + 2x_2 + 3x_3 + 4x_4 \le 1$$
$$x_1,\ x_2,\ x_3,\ x_4 \ge 0$$

$$\text{max } 2x_1 - 4x_2 + 5x_3 - 6x_4$$
$$x_1 + 4x_2 - 2x_3 + 8x_4 + s_1 = 0$$
$$-x_1 + 2x_2 + 3x_3 + 4x_4 + s_2 = 0$$
$$x_1, x_2, x_3, x_4, s_1, s_2 \geq 0.$$

Solution: Introduce slack variables in the constraints.

Cases	B.V.	Non-B.V.	Solution $(x_1,x_2,x_3,x_4,s_1,s_2)$	Value of Z	
1	x_1, x_2	$x_3 = x_4 = s_1 = s_2 = 0$	(0, 1/2, 0, 0, 0, 0)	-2	A
2	x_1, x_3	$x_2 = x_4 = s_1 = s_2 = 0$	(8, 0, 3, 0, 0, 0)	31 (Optimal)	B
3	x_1, x_4	$x_2 = x_3 = s_1 = s_2 = 0$	(0, 0, 0, 1/4, 0, 0)	-1.5	C
4	x_1, s_1	$x_2 = x_3 = x_4 = s_2 = 0$	(-1, 0, 0, 0, 3, 0)	Not a BFS	
5	x_1, s_2	$x_2 = x_3 = x_4 = s_1 = 0$	(2, 0, 0, 0, 0, 3)	4	D
6	x_2, x_3	$x_1 = x_4 = s_1 = s_2 = 0$	(0, 1/2, 0, 0, 0, 0)	-2	A
7	x_2, x_4	$x_1 = x_3 = s_1 = s_2 = 0$	Not a part of BS	Linear dependent columns	
8	x_2, s_1	$x_1 = x_3 = x_4 = s_2 = 0$	(0, 1/2, 0, 0, 0, 0)	-2	A
9	x_2, s_2	$x_1 = x_3 = x_4 = s_1 = 0$	(0, 1/2, 0, 0, 0, 0)	-2	A
10	x_3, x_4	$x_1 = x_2 = s_1 = s_2 = 0$	(0, 0, 0, 1/4, 0, 0)	-1.5	C
11	x_3, s_1	$x_1 = x_2 = x_4 = s_2 = 0$	(0, 0, 1/3, 0, 8/3, 0)	5/3=1.6	E
12	x_3, s_2	$x_1 = x_2 = x_4 = s_1 = 0$	(0, 0, -1, 0, 0, 4)	Not a BFS	
13	x_4, s_1	$x_1 = x_2 = x_3 = s_2 = 0$	(0, 0, 0, 1/4, 0, 0)	-1.5	C
14	x_4, s_2	$x_1 = x_2 = x_3 = s_1 = 0$	(0, 0, 0, 1/4, 0, 0)	-1.5	C
15	s_1, s_2	$x_1 = x_2 = x_3 = x_4 = 0$	(0, 0, 0, 0, 2, 1)	0	F

$$\times \quad \begin{array}{cc} x_2 & x_4 \\ \end{array}$$
$$\begin{bmatrix} 4 & 8 \\ 2 & 4 \end{bmatrix}$$

vertices (extreme points) $\leq$ BFS $\leq$ BS $\leq {}^{n}C_{m}$.

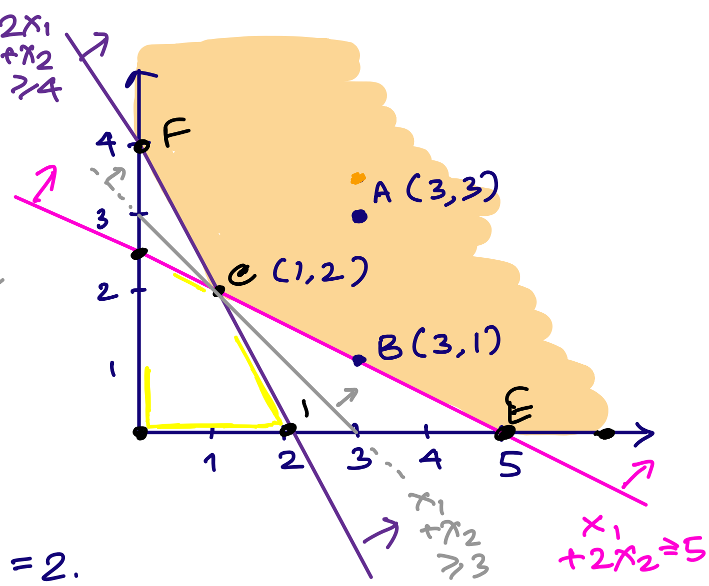
6 Corners $<$ 12 BFS $<$ 14 BS $< \binom{6}{2} = 15$

• Linear independence: $v_1, v_2, \ldots, v_n \, (\neq 0)$ are linearly independent imply $\sum_i \alpha_i v_i \neq 0$ unless all $\alpha_i = 0$.

The next theorem relates extreme point solutions to corresponding non-singular columns of the constraint matrix.

Lemma 2.1.3 *Let $P = \{x : Ax \geq b, x \geq 0\}$. For $x \in P$, let $A^=$ be the submatrix of A restricted to rows which are at equality at x, and let $A^=_x$ denote the submatrix of $A^=$ consisting of the columns corresponding to the nonzeros in x. Then x is an extreme point solution if and only if $A^=_x$ has linearly independent columns (i.e., $A^=_x$ has full column rank).*

$$
\begin{aligned}
\min \quad & 2x_1 + 3x_2 \\
\text{s.t.} \quad & 2x_1 + x_2 \geq 4 \quad ✓ \\
& x_1 + 2x_2 \geq 5 \quad ✓ \\
& x_1 + x_2 \geq 3 \quad ✓ \\
& x_1, x_2 \geq 0
\end{aligned}
$$



$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \\ 1 & 1 \end{bmatrix}$ rank(A) = 2.

$\begin{bmatrix} 2 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} = 3 \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$

<u>At C:</u> $A^= = \begin{bmatrix} 2 & 1 \\ 1 & 2 \\ 1 & 1 \end{bmatrix}$, $A^=_x = \begin{bmatrix} 2 & 1 \\ 1 & 2 \\ 1 & 1 \end{bmatrix}$ lin indep cols. $\Rightarrow$ Extreme point

<u>At F:</u> $A^= = [2 \; 1]$, $A^=_x = [1]$. $\Rightarrow$ Extreme point

<u>At A:</u> $A^=$ = 0×0 matrix. $\Rightarrow$ Not extreme point.

<u>At B:</u> $A^= = [1 \; 2]$, $A^=_x = [1 \; 2] \rightarrow$ Not lin indep $\Rightarrow$ Not extreme point

Rank Lemma:

$m \times n$

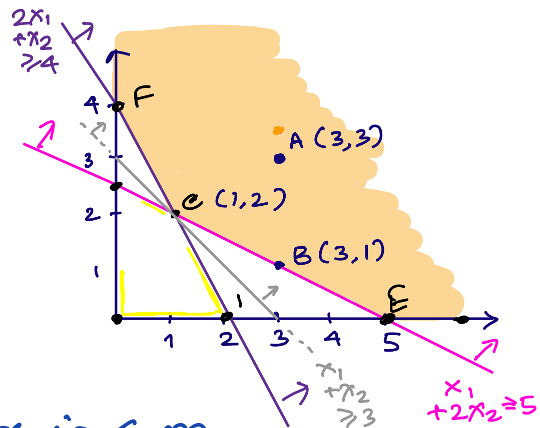
Lemma 2.1.4 (Rank Lemma) *Let $P = \{x : Ax \geq b, x \geq 0\}$ and let x be an extreme point solution of P such that $x_i > 0$ for each i. Then any maximal number of linearly independent tight constraints of the form $A_i x = b_i$ for some row i of A equals the number of variables.*

Proof Since $x_i > 0$ for each i, we have $A^=_x = A^=$. From Lemma 2.1.3 it follows that $A^=$ has full column rank. Since the number of columns equals the number of non-zero variables in x and row rank of any matrix equals the column rank†, we have that row rank of $A^=$ equals the number of variables. Then any maximal number of linearly independent tight constraints is exactly the maximal number of linearly independent rows of $A^=$ which is exactly the row rank of $A^=$ and hence the claim follows. □

atc: $x_i > 0 \ \forall i$, $A^=_x = \begin{bmatrix} 2 & 1 \\ 1 & 2 \\ 1 & 1 \end{bmatrix}$, Rank$(A^=_x) = 2$. (n)
So, maximal no. of lin. indep constraints is 2.

Rank lemma basically says if # variables is n & # constraints is $(m+n)$.
we have n linearly independent constraints (from A + nonnegativity constraints) that gets tight in an extreme point.

If all $x_i > 0$, all these n constraints come from A i.e. nontrivial constraints. And the n constraints that gets tight are linearly independent.



Useful fact:
If $n > m$, the support size of x is $\leq m$.
(As $n-m$ nonnegativity constraints need to be satisfied).

→ Rank lemma is one of the key ingredient in iterative methods. (see Book by Lau-Ravi-Singh)

- **How to solve LP ?**

A General Algorithm Design Paradigm

1. $\mathbf{x}$ is feasible for (P).
2. $\mathbf{y}$ is feasible for (D).
3. $\mathbf{x}, \mathbf{y}$ satisfy the complementary slackness conditions (Corollary 2.1).

Pick two of these three conditions to maintain at all times, and work toward achieving the third.

- **Simplex method [Dantzig '47]**

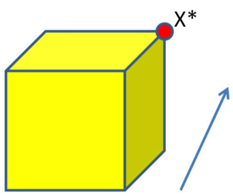
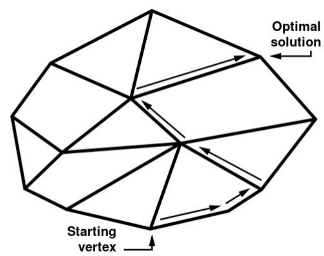
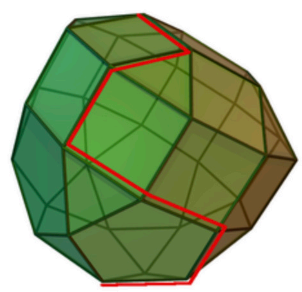


Figure 1: Illustration of a feasible set and an optimal solution x^*. We know that there always exists an optimal solution at a vertex of the feasible set, in the direction of the objective function.



A system of linear inequalities defines a polytope as a feasible region. The simplex algorithm begins at a starting vertex and moves along the edges of the polytope until it reaches the vertex of the optimal solution.

- start from a "pivot" vertex
- Local search: if there is any better neighbor vertex, move there.



Polyhedron of simplex algorithm in 3D

Maintain 1 & 3, works towards 2.

- Pros: mostly good in practice.
- Cons: worst-case exponential time.
  [n-dim polytope can have $\Omega(2^n)$ vertices]

• Ellipsoid method [Khachiyan '79]

- Pros: first polytime algo. (Proves LP is in P)
  - might even solve LPs with exponentially many constraints.
- Cons: slow for practice

maintains 1 & 2, works towards 3.

Optimization = Feasibility = Separation.

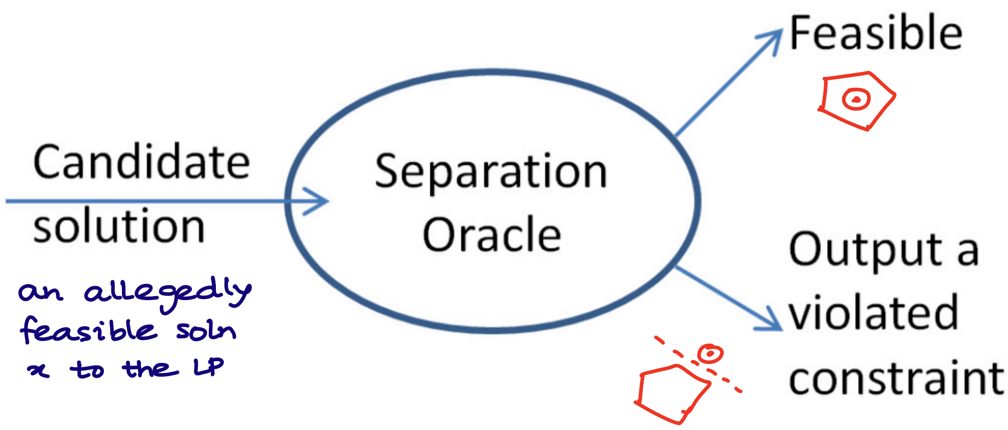


Figure 2: The responsibility of a separation oracle.

- It might even handle exponentially many constraints. [Book by Grötschel, Lovasz, Schrijver]

Consider min-cut again.

$$\min \sum_{e \in E} u_e \ell_e$$

$$\sum_{e \in P} \ell_e \geq 1 \qquad \text{for all } P \in \mathcal{P} \qquad (21)$$

$$\ell_e \geq 0 \qquad \text{for all } e \in E.$$

set of all s-t paths

can be exponential

Poly time sep. oracle:

Given ℓ_e', either feasible or returns some path P s.t. $\sum_{e \in P} \ell_e' < 1$ or some $\ell_e' < 0$.

→ Given ℓ_e', check if all $\ell_e' \geq 0$. Else return $\ell_e' < 0$.

Then run Dijkstra's algo to compute shortest s-t path, using ℓ_e' as edge lengths.

If shortest path P has length < 1.
return violated constraint $\sum_{e \in P} l_e < 1$.

Else all s-t paths have length ≥ 1
$\Rightarrow l_e'$ is a feasible solution.

- Separation oracle is heavily used in solving LPs in approximation algorithms.

W-S Book:

Ch 4.2 : minimize weighted sum of completion times on a single machine.

Ch 4.4 : the prize-collecting Steiner tree.
[separation problem is solved using max-flow]

Ch 4.6 : the bin packing problem
[configuration LP has exponentially many variables, dual LP has exponentially many constraints, Separation problem of the dual is the knapsack problem, which can be solved using an FPTAS. using that we can solve configuration LP within $(1+\epsilon)$-factor]

Ch 8.3 : the multicut problem.
[separation problem is solved using shortest path]

Ch 8.7 : linear arrangement problem.

Ch 11.2 : min-cost degree bounded spanning tree.
[separation problem is solved using max-flow]

Ch 11.3 : survivable network design.
[separation problem is solved using max-flow]

Ch 15.2 : Oblivious routing.

- To solve an LP :

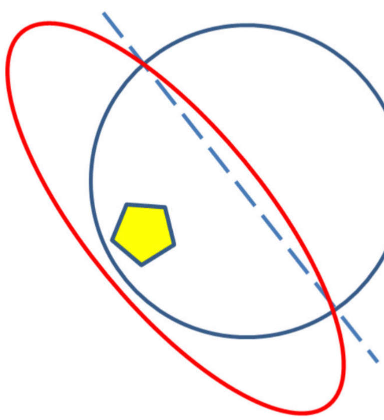
step 1 : OPT → FEASIBILITY.

Replace obj (max $c^T x$) by $c^T x \geq M$ → some target objective function value

So feasible ⇒ OPT ≥ M.

(do a binary search)

step 2 : FEASIBILITY → SEPARATION



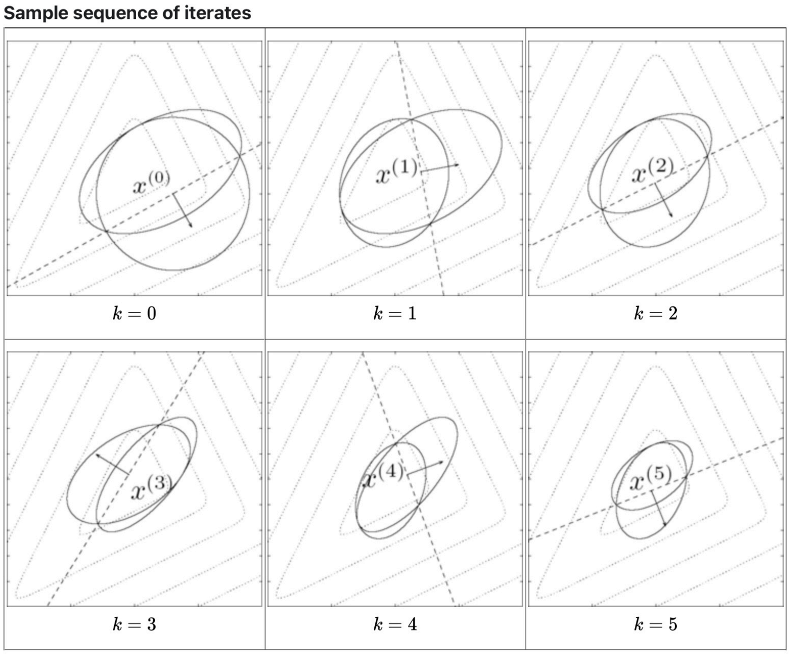


Figure 3: The ellipsoid method first initializes a huge sphere (blue circle) that encompasses the feasible region (yellow pentagon). If the ellipsoid center is not feasible, the separation oracle produces a violated constraint (dashed line) that splits the ellipsoid into two regions, one containing the feasible region and one that does not. A new ellipsoid (red oval) is drawn that contains the feasible half-ellipsoid, and the method continues recursively.

Elementary but tedious calculations show that the volume of the current ellipsoid is guaranteed to shrink at a certain rate at each iteration, and this yields a polynomial bound on the number of iterations required. The algorithm stops when the current ellipsoid is so small that it cannot possibly contain a feasible point (given the precision of the input data).

- Interior-point methods [Karmarkar '84]

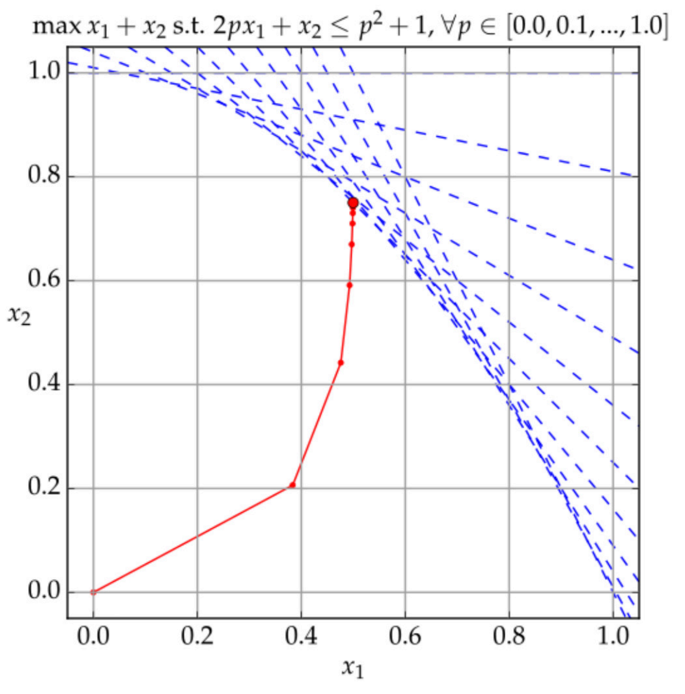
— works well in practice.

— also runs in polytime in the worst case.

"central path methods"

$$\text{maximize} \qquad \mathbf{c}^T\mathbf{x} - \lambda \cdot \underbrace{f(\text{distance between } \mathbf{x} \text{ and boundary})}_{\text{barrier function}},$$

where $\lambda \geq 0$ is a parameter and f is a "barrier function" that blows up (to $+\infty$) as its argument goes to 0 (e.g., $\log \frac{1}{z}$). Initially, one sets λ so big that the problem becomes easy (when $f(x) = \log \frac{1}{x}$, the solution is the "analytic center" of the feasible region, and can be computed using e.g. Newton's method). Then one gradually decreases the parameter λ, tracking the corresponding optimal point along the way. (The "central path" is the set of optimal points as λ varies from ∞ to 0.) When $\lambda = 0$, the optimal point is an optimal solution to the linear program, as desired.



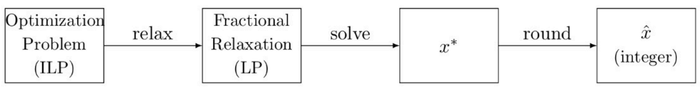
Example search for a solution. Blue lines show constraints, red points show iterated solutions.

• Integrality gap:

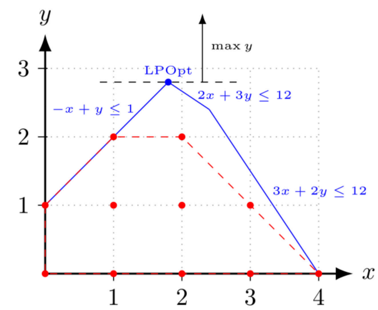
LP Relaxation and Rounding

Recall that many combinatorial problems of interest can be encoded as integer linear programs. Solving integer linear programs is in general NP-hard, so we nearly always *relax* the integrality requirement into a linear constraint like nonnegativity during our analysis.

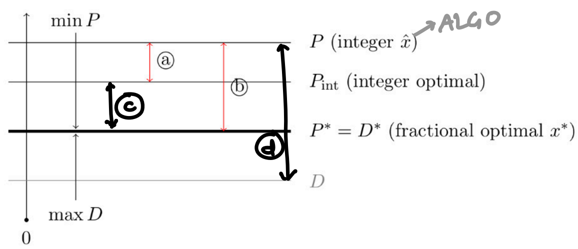
In LP rounding, we will directly round the fractional LP solution to generate an integral combinatorial solution. In most cases, this rounding will incur some loss on the solution value, so the results from LP rounding are often approximate.



we control the gap between the fractional optimal solution and our rounded solution to bound the gap between the rounded solution and the integer optimal.



A (general) integer program and its LP-relaxation



The approximation factor is ⓐ. It is often difficult to analyze this directly, so use the upper bound provided by ⓑ (LP rounding) or ⓓ (dual fitting and primal-dual). Both of these gaps, however, include the extra gap ⓒ, which we call the **integrality gap**, which is the ~~difference~~ ratio between the integer and fractional optimal solutions.

The integrality gap is a structural property of the LP, so we cannot avoid it if our approximation uses that particular LP relaxation. In fact, for most of the examples we have done, the approximation ratio we derived *was* the integrality gap (modulo constant factors).

Figure 1: LP rounding solves for x^*, the fractional optimal solution, and rounds it to an integer feasible solution $\hat{x}$. The approximation ratio is the ratio between $\hat{x}$ and the integer optimal solution (ⓐ). We bound this using the ratio between $\hat{x}$ and x^* (ⓑ).

5 Integrality Gap

We'll soon study a few approximation algorithms for set cover that are $(\log n)$-approximate. This analysis is tight in the sense that the *integrality gap* for the set cover linear program is indeed $(\log n)$ (i.e. there are examples we can construct where the optimal integer solution is $(\log n)$ times more expensive than the fractional optimal).

Definition 1. *For an integer minimization problem, let* $\mathrm{OPT}_{\mathrm{int}}$ *be its integer optimal solution and* OPT^* *be the optimal solution to a fractional relaxation. Let the possible instances to the problem be the set of* I*. The* ***integrality gap*** *of this relaxation of the problem is:*

$$\max_I \frac{\mathrm{OPT}_{\mathrm{int}}(I)}{\mathrm{OPT}^*(I)}$$

A similar form exists for maximization problems.

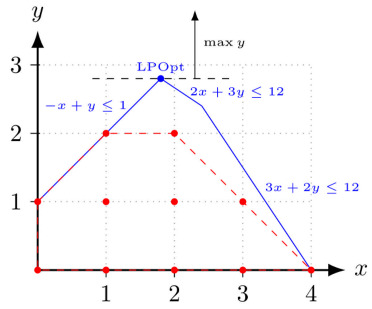
In other words, any integer approximation which relies on a bound against the fractional optimal of this program will incur this penalty in the approximation ratio. There are some problem/relaxation pairs for which there is no gap, however, and the LP optimal admits integer solutions; a few examples we have already seen include maximum flow and maximum bipartite matching. In these cases, we say that the linear program is *exact*.

Finally, integrality gaps are often *unconditional* – they do not rely on assumptions such as P! = NP, unlike many other approximation lower bounds (for example, lower bounds derived from PCP theorem). However, they only affect the specific relaxation, and may not apply to other approximation algos for the optimization problem.

• A problem can have many different LP relaxations.

E.g. Bin packing has two commonly used LPs

- assignment LP
- configuration LP (very small integrality gap)



A (general) integer program and its LP-relaxation

• Also SDPs (semidefinite program) generalize LPs & sometimes are used to obtain approximation guarantees that are not possible to obtain via LPs.

• So finding a right LP/SDP relaxation is critical. Hierarchies help here.

Instead of following the heuristic approach of finding inequalities that may be helpful for an LP or SDP, there is a more systematic (and potentially more powerful) approach lying in the use of *LP or SDP hierarchies*. In particular there are procedures by *Balas, Ceria, Cornuéjols* [BCC93]; *Lovász, Schrijver* [LS91] (with LP-strengthening *LS* and an SDP-strengthening LS_+); *Sherali, Adams* [SA90] or *Lasserre* [Las01a, Las01b]. On the t-th level, they all use $n^{O(t)}$ additional variables to strengthen an initial relaxation $K = \{x \in \mathbb{R}^n \mid Ax \geq b\}$ (thus the term *Lift-and-Project Methods*) and they all can be solved in time $n^{O(t)}$. Moreover, for $t = n$ they define the integral hull K_I and for any set of $|S| \leq t$ variables, a solution x can be written as convex combinations of vectors from K that are integral on S.