E0 249: Approximation Algorithms, Spring 2022 **Jan 19**

Week 3

Introduction to Linear Programming & LP duality

Instructors: Arindam Khan *Scribe: Aditya Lonkar*

1 Introduction to Linear Programming

We will begin with a brief description of Linear Programming (LP) and state why it is such an important tool in optimization (exact definition will be covered later). Later sections will cover other related aspects of LP like LP duality, application of LP to Max-flow Min cut, algorithms for LP and extreme points. A general reference for this part is [Rou12b; Rou12c; Rou12d; Rou12a]. Most of the figues have been adopted from [Rou12b; Rou12c; Rou12d; Rou12a].

Linear Programming (LP) is a mathematical model for optimization of a linear objective subject to linear inequality constraints. It is a very useful tool because it is polynomial time solvable (See Section 4) and can be used to model a lot of problems in practice, i.e., it is general enough and can be solved efficiently. To state some of its merits,

1. It can be used to solve many problems exactly like Max-Flow/Min-Cut, Bipartite matching, etc where properties like Total unimodularity or Total dual integrality are used to show that the LP in fact, admits an optimal integral solution.

2. It can be used to solve NP-complete problems approximately by reducing any such problem to Integer Programming (IP), another NP-complete problem where the decision variables are restricted to have only integer values. The corresponding LP (which is a relaxation of IP) in which the decision variables are real numbers between 0 and 1, can be solved first to get a fractional solution and, we can further round the values to integers using techniques like Deterministic rounding and Randomized rounding to get an integral solution.

3. LP Duality (Section 2) gives a refined understanding for many problems.

Consider the problem where unlike in LP where we have a linear objective function to optimize with respect to inequalities, we have a system of linear equations with no objective, i.e., given a $m \times n$ matrix A and a $m \times 1$ vector b, we want to find whether a solution exists for

$$Ax = b$$

for vectors $x \in \mathbb{R}^n$.

It turns out that this problem is significantly easier than LP and can be solved by the well known algorithm of Gaussian elimination, which runs in $O(n^3)$ time. Gaussian elimination either correctly returns a feasible solution (there can be many) or reports that no solution exists. For an LP, we can have infinite feasible solutions and need to output the best with respect to the given objective function which is a harder problem.

Let us now define an LP more formally.

Ingredients of a Linear Program

1. *Decision variables* $x_1, ..., x_n \in \mathbb{R}$
2. *Linear constraints*, each of the form
$$\sum_{j=1}^{j=n} a_j x_j \ (*) \ b_i$$
where (*) could be $\geq, \leq$ or $=$.
3. A *linear objective function*, of the form
$$\max \sum_{j=1}^{j=n} c_j x_j$$
or
$$\min \sum_{j=1}^{j=n} c_j x_j$$

Just to give some examples of expressions that are not allowed in an LP: $x_j^2, x_j x_k, \log(1 + x_j)$ for some variables x_j, x_k. Some other points to note here are

- An equality constraint can be converted to two inequalities. For eg: $a = b$ can be equivalently written as $a \leq b$ and $b \leq a$
- $a \geq b$ can be equivalently written as $-a \leq -b$
- A maximization objective can be written equivalently as a minimization objective. That is, $\max \sum_j c_j x_j$ is equivalently $\min - \sum_j c_j x_j$.
- Inequalities can be expressed as equalities along with a non-negativity constraint as follows: $a \geq b$ can be equivalently written as $a = b + c$ and $c \geq 0$ where c is a new variable.

This leads us to a definition of a standard LP : Primal LP.

Standard LP: Primal (P)

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

subject to

for $i \in [m]$,

$$a_{ij} x_j \leq b_i$$

$$x_1, ..., x_n \geq 0$$

The linear program has n non-negative decision variables $x_1, ..., x_n$ and m constraints (not counting the non-negativity ones). The a_{ij}s, b_js and c_js are all part of the input (fixed constants).

Equivalently, in matrix-vector notation the LP can be formulated as follows:

$$\max c^T x$$

subject to:

$$Ax \leq b$$
$$x \geq 0$$

Let's look at a concrete example of a LP.

Example 1.1. Here is an algebraic view of a LP.

$$\max x_1 + x_2 \tag{1}$$

subject to

$$4x_1 + x_2 \leq 2 \tag{2}$$
$$x_1 + 2x_2 \leq 1 \tag{3}$$
$$x_1 \geq 0 \tag{4}$$
$$x_2 \geq 0 \tag{5}$$

Below is a geometric view of the same example.

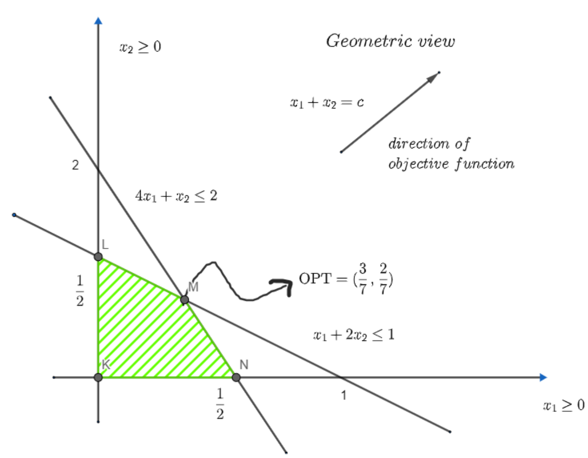


Figure 1: A geometric view of the example

We will state some points regarding correspondence between the algebraic and geometric perspective of an LP:

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

3. The optimal solution is the feasible point furthest in the direction of c (for a maximization problem).
4. When there is a unique optimal solution, it is a vertex (corner) of the feasible region.

There can be some edge cases with respect to solution of an LP.

i There might be no feasible solutions at all.

ii The optimal objective function value is unbounded. Note that a necessary but not sufficient condition for this case is that the feasible region is unbounded.

iii The optimal solution need not be unique, as a 'side' of the feasible region might be parallel to the level sets of the objective function. Whenever the feasible region is bounded, there always exists an optimal solution that is a vertex of the feasible region.

2 LP duality

In Example 1.1 the proposed optimal solution had value $\frac{5}{7}$ and occurred at $(\frac{3}{7}, \frac{2}{7})$. But we cannot be sure that it is indeed the maximum value of the objective without giving a formal proof, for which we get to the notion of LP-duality. We will show an upper bound of $\frac{5}{7}$ on the objective value of the function in the example which together with the fact that we have a point at which the same value occurs, implies that it is indeed the optimum value.

For our first attempt, observe that

$$x_1 + x_2 \leq x_1 + 2x_2 \leq 1 \qquad \text{[From (3)]}$$

The first inequality holds because of non-negativity of x_2. For our second attempt,

$$x_1 + x_2 \leq \frac{1}{7}(4x_1 + x_2) + \frac{3}{7}(x_1 + x_2) \leq \frac{1}{7} \cdot 2 + \frac{3}{7} \cdot 1 = \frac{5}{7}$$

Note that the second inequality follows from (2) and (3) in the example. Thus, OPT $\leq \frac{5}{7}$ and we have a feasible solution with value $\frac{5}{7}$ which implies that it is indeed the optimal feasible solution. Generalizing this approach to bound the value of the optimal solution motivates the definition of the dual LP.

Thus, the problem is to get an upper bound on the value of the optimal objective and the approach is to take a linear combination of the constraints that componentwise dominate the objective function.

Dual LP (D)

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

such that

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

$$y_1, ..., y_m \geq 0$$

Alternatively, in the matrix-vector notation:

$$\min b^T y$$

such that

$$A^T y \geq c$$
$$y \geq 0$$

For every feasible solution $(x_1, ..., x_n)$ of (P),

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

$$= \sum_{i=1}^{i=m} y_i \cdot \left(\sum_{j=1}^{j=n} a_{ij} x_j \right) \qquad \text{(Reversing the order of the sum)}$$

$$\leq \underbrace{\sum_{i=1}^{i=m} y_i b_i}_{\text{upper bound}} \qquad \text{(constraints of P + non-negativity of } y_i\text{s)}$$

Thus, we have that OPT of P $\leq \sum_{i=1}^{i=m} b_i y_i$.

We state a simple recipe for conversion between P and D.

Primal	Dual
variables $x_1, ..., x_n$	n constraints
m constraints	variables $y_1, ..., y_m$
objective function c	right-hand side c
right-hand side b	objective function b
$\max c^T x$	$\min b^T y$
constraint matrix A	constraint matrix A^T
ith constraint is "$\leq$"	$y_i \geq 0$
ith constraint is "$\geq$"	$y_i \leq 0$
ith constraint is "$=$"	$y_i = 0$
$x_i \geq 0$	jth constraint is "$\leq$"
$x_i \leq 0$	jth constraint is "$\geq$"
$x_i = 0$	jth constraint is "$=$"

Note that it can be shown that the Dual of the Dual is the Primal.

Theorem 2.1 (Weak Duality). *For every maximization linear program (P) and corresponding dual linear program (D),*

$$\textit{OPT value for (P)} \leq \textit{OPT value for (D)}$$

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

$$\textit{OPT value for (P)} \geq \textit{OPT value for (D)}$$

Weak duality has some interesting corollaries.

Corollary 2.2. *Let (P), (D) be primal-dual pairs 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 x, y are feasible for (P), (D) and $c^T x = b^T y$, then x, y are respectively optimal for (P) and (D).*

2.1 Complementary Slackness

Now we give a corollary of Theorem 2.1. It is a sufficient and a necessary condition for optimality (as we will see later).

Corollary 2.3 (Complementary slackness conditions). *Let (P), (D) be a primal-dual pair of linear programs. If x, y are feasible solutions to (P), (D) respectively and the following two conditions hold, then they are both optimal.*

(1) *Whenever $x_j \neq 0$, y satisfies the jth constraint of (D) with equality.*

(2) *Whenever $y_i \neq 0$, x satisfies the ith constraint of (P) with equality.*

The conditions assert that no decision variable and the corresponding constraint are simultaneously "slack" (that is, it does not allow for both of them to not be tight at the same time).

Proof: We prove the corollary for (P) and (D) in standard forms as mentioned (the proof follows for any variation in the way the LPs are described).

The first condition implies that

$$c_j x_j = \left(\sum_{i=1}^{i=m} y_i a_{ij} \right) x_j \tag{1}$$

for each $j = 1..., n$. Similarly the second condition implies that

$$y_i \left(\sum_{j=1}^{j=n} a_{ij} x_j \right) = y_i b_i \tag{2}$$

for each $i = 1..., m$. Summing equation (1) over $j = 1$ to $j = n$ and equation (2) over $i = 1$ to $i = m$ implies $c^T x = b^T y$ and Corollary 2.2 implies that both x and y are optimal. ∎

Theorem 2.4 (Strong LP Duality). *When a primal-dual pair (P), (D) of linear programs are both feasible,*

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

Corollary 2.5 (LP Optimality conditions). *Let x and y be feasible solutions to a primal-dual pair (P), (D) of linear programs. Then x and y are both optimal*

- *if and only if $c^T x = b^T y$*
- *if and only if complementary slackness conditions hold.*

Theorem 2.4 is proved using Farkas's Lemma which we state now although we do not go into the proof of either of these statements.

Lemma 2.6 (Farkas's Lemma). *Given a matrix $A \in \mathbb{R}^{m \times n}$ and a right-hand side $b \in \mathbb{R}^m$, exactly one of the following holds.*

1. *There exists $x \in \mathbb{R}^n$ such that $x \geq 0$ and $Ax = b$.*
2. *There exists $y \in \mathbb{R}^m$ such that $y^T A \geq 0$ and $y^T b < 0$.*

3 Application of LP-Duality

In this section, we study an application of LP-Duality in the context of Max-Flow Min-Cut. Recall the Max-Flow problem: Given a directed graph $G = (V, E)$, source $s \in V$, sink $t \in V$ and positive arc capacities $c : E \to \mathbb{R}^+$, the goal is to find a max-flow that can be sent from s to t subject to the following constraints:

- **Capacity constraints:** For each arc e, the flow sent through e is bounded by its capacity, i.e., $f_e \leq c_e$ where f_e denotes the flow through arc e.
- **Flow conservation:** $\forall v \in V \setminus \{s, t\}$,

$$\text{total flow into } v = \text{total flow out of } v$$

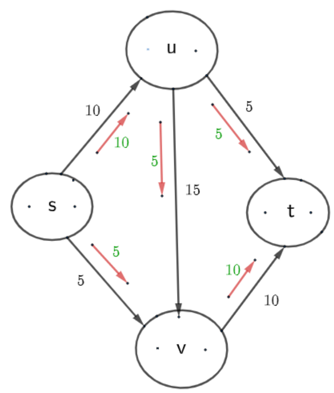


Figure 2: A flow network with source s and sink t. The numbers next to the edges in black are the capacities. The numbers in red and the corresponding arrows indicate the amount of flow in that edge.

Now, we will formulate Max-Flow as an LP. We will add a fictitious arc from t to s so that we can conserve flow even at s and t.

$$\max \quad f_{ts}$$

subject to

$$f_{ij} \leq c_{ij}, \qquad (i,j) \in E$$

$$\sum_{j:(j,i)\in E} f_{ji} - \sum_{i:(i,j)\in E} f_{ij} \leq 0, \qquad i \in V$$

$$f_{ij} \geq 0, \qquad (i,j) \in E$$

where f_{ij} represents the flow on edge $(i, j) \in E$.

The second constraint should be an equality for conservation of flow at a node. However, since sum of all those constraints over $i \in V$ is 0. it implies that R.H.S of none of them can be strictly negative. Now, we formulate the same LP keeping the matrix format in mind.

3.1 Primal LP (P) for Max-Flow

$$\max \quad 1 \cdot f_{ts} + \sum_{e \in E} 0 \cdot f_e$$

such that

$$1 \cdot f_{ij} + \sum_{e \in E \setminus \{(i,j)\}} 0 \cdot f_e \leq c_{ij} \qquad (i,j) \in E \qquad (1)$$

$$\sum_{j:(j,i) \in E} 1 \cdot f_{ji} + \sum_{i:(i,j) \in E} (-1) \cdot f_{ij} \leq 0 \qquad i \in V \qquad (2)$$

$$f_{ij} \geq 0$$

Note that here $m = |E| + |V|$, that is, the number of constraints in the primal and $n = |E| + 1$ which is the number of variables. So we have $|E|$ variables in the dual corresponding to constraint (1) in the primal and for any such constraint corresponding to $(i,j) \in E$ we have a corresponding variable d_{ij} in (D). Similarly, we have $|V|$ variables in the dual corresponding to constraint (2) in the dual and for any such constraint corresponding to $i \in V$ we have a corresponding variable p_i in (D).

3.2 Dual LP (D) for Max-Flow

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

such that

$$\begin{aligned}
\text{for } (t,s), \qquad & 1 \cdot p_s + (-1) \cdot p_t \geq 1 \\
\text{for } E \setminus (t,s), \qquad & d_{ij} + p_j - p_i \geq 0 \\
& d_{ij} \geq 0 \\
& p_i \geq 0
\end{aligned}$$

3.3 Intuitive understanding of (D)

We will see how the dual (D) formulated in the last subsection has an intuitive understanding w.r.t cuts in the graph. To that end, for now, assume that $d_{ij} \in \{0,1\}$ and $p_i \in \{0,1\}$, i.e, we are now interested in the integer program of (D). Let d_{ij} be some distance label on arcs and p_i be potentials on nodes. Let (d^*, p^*) be an optimal solution to this integer program (I). Since (d^*, p^*) is an optimal solution to the IP, it is a feasible solution to (D) as well. Hence, we have that

$$p_s^* - p_t^* \geq 1 \implies p_s^* = 1, p_t^* = 0$$

This defines a $s-t$ cut $(X, \overline{X})$ where X is the set of potential 1 nodes and $\bar{X}$ is the set of potential 0 nodes. Note that having higher integral potentials for the nodes does not make sense as (D) is a minimization program and hence, we restrict their values to $\{0,1\}$.

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

$$d_{ij} \geq p_i - p_j \implies d_{ij} = 1$$

For arc (i,j) with $i \in X$ and $j \in X$, $i \in \bar{X}$ and $j \in \bar{X}$ or $i \in \bar{X}$ and $j \in X$, $d_{ij} \geq 0$, thus can be set to 1 or 0. To minimize the objective we set it to 0.

Thus, objective of dual = min $s-t$ cut. That is, partition the vertex set V into $(X, \bar{X})$ such that the weight of arcs going from X to $\bar{X}$ is minimized (The weight of arc (i,j) given by c_{ij}).

Another observation for (D) is that any path from $s-t$ in G contains at least one edge from a cut C by definition. This inspires the interpretation of dual as a fractional $s-t$ cut (Note that now the decision variables need not be integral). The distance labels assigned to arcs by the dual satisfy the property that distance labels on any $s-t$ path $(s = v_0, v_1, ..., v_k = t)$ sum to at least 1. This is because

$$\sum_{i=0}^{i=k-1} d_{v_i v_{i+1}} \geq \sum_{i=0}^{i=k-1} (p_{v_i} - p_{v_{i+1}}) = p_s - p_t \geq 1$$

The first and second inequalities follow from the first and the second constraints in (D), respectively.

3.4 Relating Max-Flow Min-Cut

Due to capacity constraints, capacity of any $s-t$ cut is an upper bound on any feasible flow that can be sent from s to t. Hence, Max-Flow is at most the value of the Min-Cut. This is also evident from Weak LP-Duality.

$$\text{Max-Flow} \underset{\text{Weak LP-Duality}}{\leq} \text{Dual-LP} \underset{\text{LP relaxation}}{\leq} \text{Integer program for Min-Cut}$$

Surprisingly, we have a stronger property

$$\text{Max-Flow} \underset{\text{Strong LP-Duality}}{=} \text{Dual-LP} \underset{\text{Integral LP polyhedron}}{=} \text{Integer program for Min-Cut}$$

This implies Max-Flow = Min-Cut. The second equality is due to the fact that the Dual LP has an integral optimal solution which we will give a few pointers to. To show integrality of an LP, either show that the constraint matrix is unimodular or by total dual integrality. A totally unimodular matrix is one where every square non-singular submatrix is unimodular. Equivalently every square submatrix has determinant $0, -1, 1$. It follows then that any totally unimodular matrix has either $0, -1, 1$ as its entries.

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

$$\max c^T x$$
$$Ax \leq b,$$

there is an optimal dual solution.

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

3.5 Understanding Complementary Slackness

We will alternatively show that Max-Flow = Min-Cut through complementary slackness. Let f^* be optimal solution to (P), that is, it is the max flow and (d^*, p^*) be an optimal solution to (D) (Min-Cut defined by $(X, \bar{X})$).

- Say arc (i, j) has $i \in X$ and $j \in \bar{X}$. Then $d^*_{ij} = 1$, i.e., $d^*_{ij} \neq 0$ by virtue of complementary slackness implies that $f^*_{ij} = c_{ij}$.
- Say arc (k, l) has $k \in \bar{X}$ and $l \in X$. Then $p^*_k - p^*_l = -1$ and since $d^*_{kl} \in \{0, 1\}$ implies that $d^*_{kl} - p^*_k + p^*_l \geq 0$ must be strict inequality. Hence, by complementary slackness we get that $f^*_{ij} = 0$.

This implies that arcs going from X to $\bar{X}$ are saturated with flow and arcs going from $\bar{X}$ to X have no flow which in turn implies that Max-Flow = Min-Cut.

3.6 Alternate formulation of Dual LP for Max-Flow

For Max-Flow, consider an alternate LP based on path decomposition which is shown below. The advantage here is that there is no need for explicitly stating the conservation constraints. Let $\mathcal{P}$ denote the set of all $s - t$ paths. Then one can show that the following LP is equivalent to the originally stated Max-Flow LP in Subsection 3.1.

$$\max_{P \in \mathcal{P}} f_P$$

subject to

$$\sum_{P \in \mathcal{P} : e \in P} f_P \leq c_e$$

$$f_P \geq 0 \quad \forall P \in \mathcal{P}$$

We will define a corresponding dual variable l_e for the 1st constraint in the primal for each edge e. The dual program is as follows:

$$\min \sum_{e \in E} c_e l_e$$

subject to

$$\sum_{e \in P} l_e \geq 1 \quad \forall P \in \mathcal{P}$$

$$l_e \geq 0 \quad \forall e \in E$$

Again we show that the dual corresponds to the Min-Cut. For a fixed cut $(X, \bar{X})$ with $s \in X$ and $t \in \bar{X}$, set

$$\begin{aligned} l_{ij} &= 1 \text{ if } i \in X \text{ and } j \in \bar{X} \\ &= 0 \text{ else} \end{aligned}$$

Every $s - t$ path has at least one edge in $(X, \bar{X})$, hence if that edge is say (i, j), then $l_{ij} = 1$ and the constraint inequality is satisfied.

The objective value $\sum_{e \in E} c_e l_e = \sum_{i \in X, j \in \bar{X}} c_{ij} = Cut[X, \bar{X}]$. Hence, again

$$\text{Max-Flow} \underset{\text{Strong LP-Duality}}{=} \text{Dual-LP} \underset{\text{Integral LP polyhedron}}{=} \text{Integer program for Min-Cut}$$

Many other interesting theorems/algorithms can be viewed as a consequence of LP-Duality. For eg. Minimax theorem and Hungarian algorithms.

4 Algorithms for LP

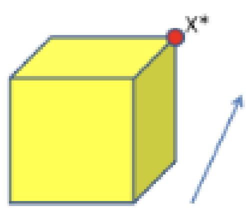
<u>How to solve LPs?</u>

A General Algorithm design paradigm:

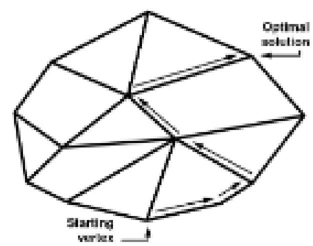
1. x is feasible for (P).
2. y is feasible for (D).
3. x, y satisfy complementary slackness conditions.

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

4.1 Simplex method



(a) Illustration of a feasible set and its 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.



(b) 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.

We briefly describe the Simplex method [Dantzig '47].

- Start from a "pivot" vertex.
- Local search: If there is any better neighbour vertex, move there.

Maintain 1, 3 and work toward 2 according to the paradigm. The advantage of this method is that it works very well in practice but the disadvantage is that, in worst case it can take exponential time. For most practical instances, it works quite well.

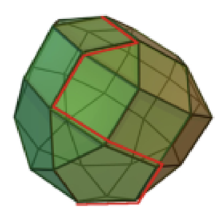


Figure 4: Polyhedron of simplex algorithm in 3D.

4.2 Ellipsoid method

Ellipsoid method [Khachiyan '79] gave a breakthrough here, in solving LPs in polynomial time.

The advantages of this method are: the first polynomial time algorithm for LP, and it can even solve LPs with exponentially many constraints.

The con of this method is that it can be very slow in practice.

Here, we maintain 1, 2 and work towards 3 according to the paradigm.

A famous result by [GLS12], which is used in the Ellipsoid method is

$$\text{Optimization} = \text{Feasibility} = \text{Separation}$$

Which says that optimization over a polytope is equivalent to finding whether a system of inequalities is feasible and these two are equivalent to answering the question of whether a point lies inside a polytope (If not, then output a separating hyperplane between the point and the polytope).

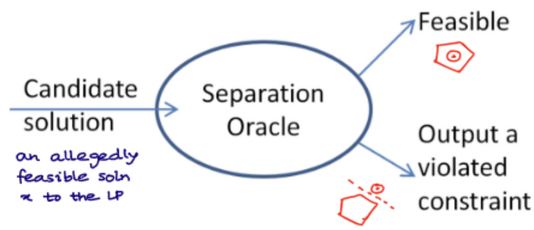


Figure 5: The responsibility of a separation oracle.

We will an example of Separation oracle: Consider the Min-Cut LP discussed in Section 3.6 where the number of non-trivial constraints is equal to the number of $s-t$ paths, which can be exponential. We will demonstrate a polynomial time separation oracle for the LP which does the following:

Given l'_e, either returns that it is a feasible solution to the LP or returns some path P such that $\sum_{e \in P} l'_e < 1$ or some $l'_e < 0$.

The solution is as follows:

Given l'_e first check whether $l'_e \geq 0$ for all $e \in E$. If for some $e \in E$, $l'_e < 0$ return it. If that is not the case and $l'_e \geq 0$ for all $e \in E$, then run Dijkstra's algorithm to compute $s-t$ shortest path, using l'_e as lengths on the edges. Now, if shortest path has length < 1, return violated constraint. Else all $s-t$ paths have length ≥ 1 which implies l'_e is a feasible solution.

Important to mention here that separation oracles are heavily used in the design of approximation algorithms and a lot of examples of the same can be found in [WS11].

To Solve an LP:

Step 1: OPT $\rightarrow$ FEASIBILITY

To solve an LP, we first convert the optimization problem to a feasibility problem. Replace the objective (lets say $c^T x$) by a linear constraint $c^T x \geq M$ where M is some target objective function value. Thus, if we can efficiently check feasibility we can do a binary search w.r.t the objective value.

Step 2: FEASIBILITY $\rightarrow$ SEPARATION

This is where we use the Ellipsoid method as mentioned below.

Elementary but tedious calculations show that 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).

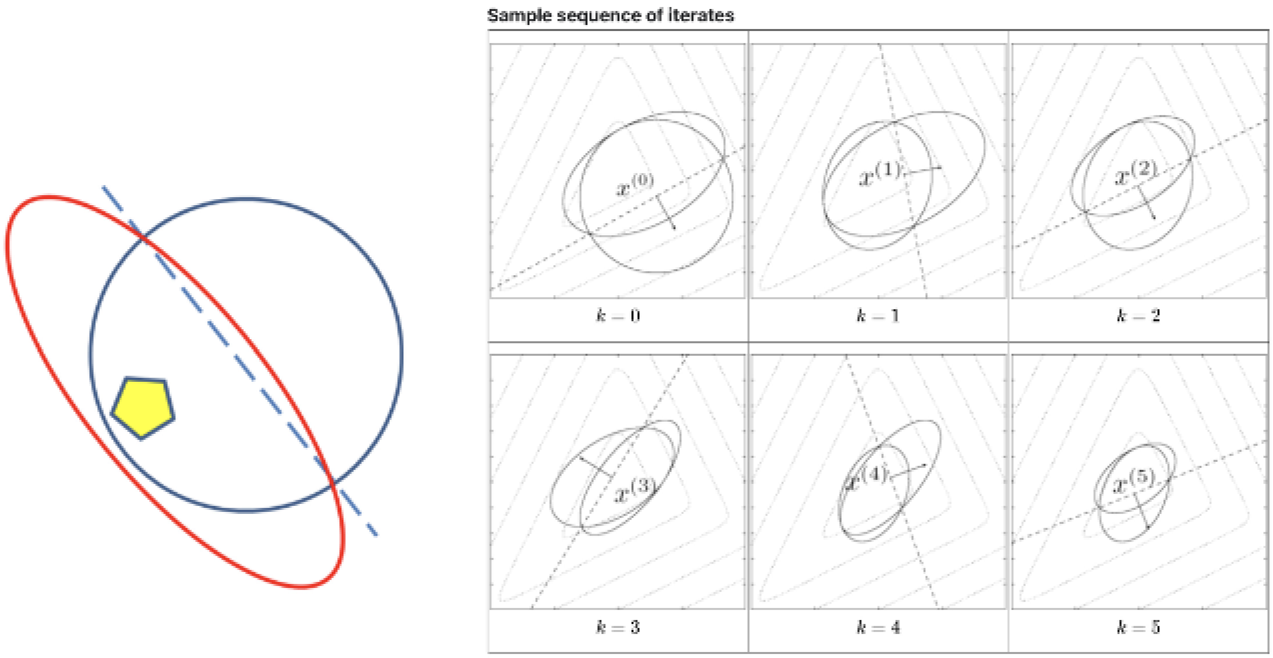


Figure 6: The ellipsoid method initializes a huge sphere (blue circle) that encompasses the feasible region (yellow pentagon). If the ellipsoid center is not feasible, then the separation oracle produces a violated constraint (dashed line) that splits the ellipsoid into two regions, one containing the feasible region and one that does not. A new ellipsoid is drawn (red oval) and the method continues recursively.

4.3 Interior-point methods

Interior point methods [Karmarkar '84], again one of the landmark results. This method runs in polynomial time and also works well in practice. Many LP solvers in fact, use this method.

Here, you have an objective function:

$$\max \quad c^T x - \lambda \cdot \underbrace{f(\text{distance between } x \text{ and boundary})}_{\text{barrier function}}$$

where $\lambda \geq 0$ is a parameter and f is a "barrier function" that blows up (to ∞) as its argument goes to 0. 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 eg. Newton's method). Then one gradually decreases 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.

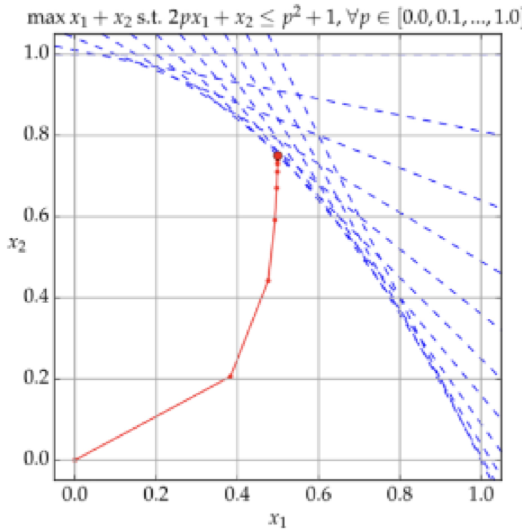
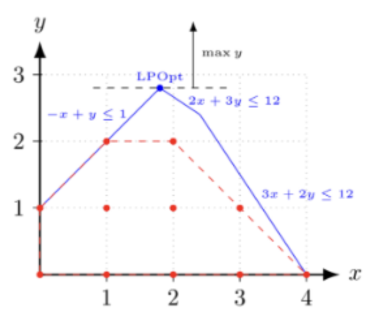


Figure 7: Example search for solutions. Blue lines show constraints and red lines show iterated solutions.

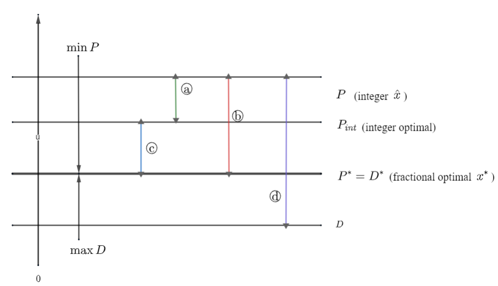
4.4 Integrality gap

Recall that many combinatorial problems of interest are NP-Hard and hence, can be encoded as integer linear programs. Since, solving an integer LP is hard by the same token, we nearly always *relax* the integrality into a linear non-negativity constraint during our analysis.

In LP rounding, we will directly round the LP solution to generate an integer combinatorial solution. In



(a) A (general) integer program and its LP relaxation.



(b) 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^* (ⓑ)).

Figure 8b, the approximation factor is ⓐ. It is often difficult to analyze this, so we use the upper bound provided by ⓑ (LP rounding) or ⓓ (dual fitting and primal-dual). Here ⓒ is called the integrality gap which is the difference between the integer and fractional optimal solutions. It is a structural property of the LP and cannot be avoided if that particular LP is used, in the approximation factor.

Definition 4.1 (Integrality gap)**.** *For an integer minimization problem, let OPT_{int} be the optimal solution and OPT^* be the optimal solution to its fractional relaxation. Let the possible set of instances to this problem* *be I. Then the integrality gap of this relaxation of the problem is:*

$$\max_{I} \frac{OPT_{int}}{OPT^*}$$

A similar form exists for maximization problems.

In other words, any integer approximation which relies on a bound against the fractional optimal will incur this loss in the approximation ratio. Some problems, however exist without any integrality gap. Another point to note here is that integrality gaps are unconditional. They do not rely on any assumption of $\mathsf{P} \neq \mathsf{NP}$. However, different LP relaxations can have different integrality gaps. One example of such a problem is BIN PACKING. The two commonly used LPs are:

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

Also SDPs (semi-definite programs) are used to generalize LPs and sometimes, used to obtain approximation guarantees that are not possible to be obtained via LPs. SDPs have decision variables as vectors instead of scalars. Finding the right LP/SDP relaxation is critical. Hierarchies help here (which we will see later in the course).

Instead of finding the heuristic approach of finding inequalities that may be helpful for an LP or an 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 SDP strengthening LS_+); Sherali, Adams [SA90] or Lassere [Las01a; Las01b]. On the t-th level, they all use $n^{O(t)}$ additional variables to strengthen an initial relaxation $K = \{x \in \mathbb{R}^n | Ax \geq b\}$ (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.

5 Extreme points

For this section, the reference is [LRS11].

Definition 5.1. *Let $P = \{x : Ax = b, x \geq 0\} \in \mathbb{R}^n$. 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 $x + y, x - y \in P$. This is also known as a vertex solution or a basic feasible solution (to be covered later).*

Definition 5.2. *Let P be a polytope and x be an extreme point solution of P then x is integral if each coordinate of x is an integer. Then P is called integral if every extreme point of P is integral.*

Lemma 5.3. *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 $x' \in P$ such that $Ax' = b$ and $c^T x' \leq c^T x$, i.e., there is always an extreme point optimal solution.*

5.1 Basic feasible solution

Consider the linear program:

$$\min \quad c^T x$$

such that

$$Ax \geq b$$

$$x \geq 0$$

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

$$\min \; c^T x$$

such that

$$Ax + s \geq b$$

$$x \geq 0$$

$$s \geq 0$$

where s is a vector. Henceforth, we study linear programs in standard from: $\{\min c^T x : Ax = b, x \geq 0\}$. Without loss of generality, we can assume that A is full row-rank. Otherwise, we can remove constraints which are linear combinations of other constraints and still have an equivalent linear program. Now, we define a basic feasible solution.

A subset of columns B of the constraint matrix A is called a *basis* if the matrix of the 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., $A_B^{-1}B \geq 0$, then it is called a *basic feasible solution*. The correspondence between bases and basic feasible solution is not one to one. There can be many bases which correspond to the same basic feasible solution. The next theorem shows equivalence between extreme point solutions and basic feasible solutions.

Theorem 5.4 ([LRS11]). *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.*

Hence, a basic solution implies A_B is invertible and $rank(A_B) = rank(A) = m$. So, put $n - m$ variables to be 0, other m variables are called basic variables. Resulting system is $A_B X_B = b$ and if A_B is invertible, then the solution is a basic solution (BS). If all variables in BS are ≥ 0, then it is a basic feasible solution (BFS). Otherwise, it is called an infeasible solution.

If some basic variables are 0 in BFS, then it is called a degenerate solution. On the other hand, if all of them are positive, it called a non-degenerate solution. Because of Lemma 5.3, we know that there exists an extreme point as a optimal solution (assuming feasibility of the program) and Theorem 5.4 then implies that there exists an optimal solution which is a basic feasible solution.

$$\text{No. of extreme points} \leq \text{BFS} \leq \text{BS} \leq \binom{n}{m}$$

To explain each inequality, the last inequality follows from the fact that some bases might be non-invertible. Second inequality is due to some of the solutions in BS being infeasible and the first inequality is because some solutions in BFS can be degenerate and multiple such ones can map to the same extreme point.

We will now look at some examples.

Example 5.1.

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

subject to

$$2x_1 + x_2 \leq 4$$

$$x_1 + 2x_2 \leq 5$$

$$x_1 \geq 0$$
$$x_2 \geq 0$$

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

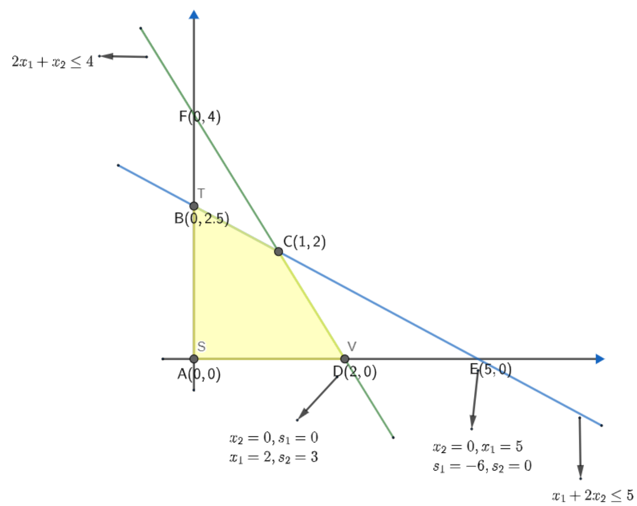


Figure 9: Example demonstrating the number of BS, BFS and extreme points.

The equivalent LP in standard form with slack variables s_1, s_2 is

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

subject to

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

In this example, $n = 4$ and $m = 2$. Hence, we show what happens for all $\binom{4}{2} = 6$ choices for the bases in the constraint matrix A which is $\begin{bmatrix} 2 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \end{bmatrix}$ There are 4 extreme points of relevance here, 4 BFS, 6 BS. Note that all the solutions here are non-degenerate.

Another more involved example which demonstrates this.

Example 5.2.

$$\max \;\; 2x_1 + x_2$$

subject to

$$\begin{aligned} x_1 + x_2 &\leq 3 \\ x_1 - x_2 &\leq 0 \\ x_1 &\geq 0 \\ x_2 &\geq 0 \end{aligned}$$

The equivalent LP in standard form with slack variables x_3, x_4 is

$$\max \;\; 2x_1 + x_2$$

subject to

$$\begin{aligned} x_1 + x_2 + x_3 &= 3 \\ x_1 - x_2 + x_4 &= 0 \\ x_1 &\geq 0 \\ x_2 &\geq 0 \\ x_3 &\geq 0 \\ x_4 &\geq 0 \end{aligned}$$

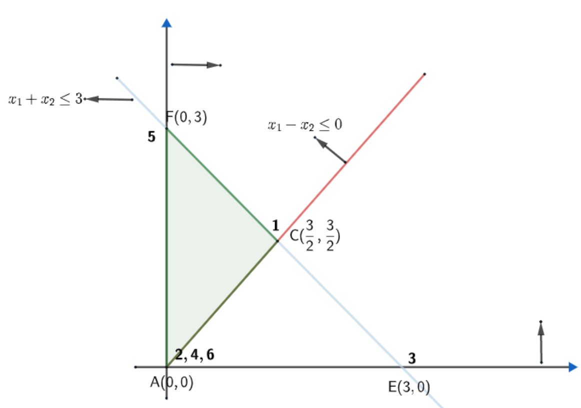


Figure 10: Figure showing geometric view of the above example. The numbers in bold indicate the corresponding point in the table below.

	Basic variable	Non-basic variable	Solution x_1, x_2, x_3, x_4	
1	x_1, x_2	$x_3 = x_4 = 0$	$(\frac{3}{2}, \frac{3}{2}, 0, 0)$	BS, BFS non-degen
2	x_1, x_3	$x_2 = x_4 = 0$	$(0, 0, 3, 0)$	BS BFS degen
3	x_1, x_4	$x_2 = x_3 = 0$	$(3, 0, 0, -3)$	BS, infeasible
4	x_2, x_3	$x_1 = x_4 = 0$	$(0, 0, 3, 0)$	BS, BFS degen
5	x_2, x_4	$x_1 = x_3 = 0$	$(0, 3, 0, 3)$	BS, BFS non-degen
6	x_3, x_4	$x_1 = x_2 = 0$	$(0, 0, 3, 0)$	BS, BFS degen

Hence, we have 3 corners (extreme-points), 5 BFS and 6 BS. For non-degenerate BFS there is a one-to-one correspondence between BFS and vertex but not for degenerate BFS. At any BFS, there are n linearly independent tight constraints.

An example to demonstrate that not all A_B need be basic solutions, i.e., A_B might be non-invertible.

Example 5.3.

$$\max \;\; 2x_1 - 4x_2 + 5x_3 - 6x_4$$

subject to

$$\begin{aligned} x_1 + 4x_2 - 2x_3 + 8x_4 &\leq 2 \\ -x_1 + 2x_2 + 3x_3 + 4x_4 &\leq 1 \\ x_1 &\geq 0 \\ x_2 &\geq 0 \\ x_3 &\geq 0 \\ x_4 &\geq 0 \end{aligned}$$

An equivalent LP in standard form with slack variables is:

$$\begin{aligned} x_1 + 4x_2 - 2x_3 + 8x_4 + s_1 &= 2 \\ -x_1 + 2x_2 + 3x_3 + 4x_4 + s_2 &= 1 \\ x_1 &\geq 0 \\ x_2 &\geq 0 \\ x_3 &\geq 0 \\ x_4 &\geq 0 \\ s_1 &\geq 0 \\ s_2 &\geq 0 \end{aligned}$$

Cases	Basic variable	Non-basic variable	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
2	x_1, x_3	$x_2 = x_4 = s_1 = s_2 = 0$	$(8, 0, 3, 0, 0, 0)$	31 (Optimal)
3	x_1, x_4	$x_2 = x_3 = s_1 = s_2 = 0$	$(0, 0, 0, 1/4, 0, 0)$	-1.5
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
6	x_2, x_3	$x_1 = x_2 = s_1 = s_2 = 0$	$(0, 1/2, 0, 0, 0, 0)$	-2
7	x_2, x_4	$x_1 = x_3 = s_1 = s_2 = 0$	Not a part of BS	Linearly dependent columns
8	x_2, s_1	$x_1 = x_3 = x_4 = s_2 = 0$	$(0, 1/2, 0, 0, 0, 0)$	-2
9	x_2, s_2	$x_1 = x_3 = x_4 = s_1 = 0$	$(0, 1/2, 0, 0, 0, 0)$	-2
10	x_3, x_4	$x_1 = x_2 = s_1 = s_2 = 0$	$(0, 0, 0, 1/4, 0, 0)$	-1.5
11	x_3, s_1	$x_1 = x_2 = x_4 = s_2 = 0$	$(0, 0, 1/3, 0, 8/3, 0)$	1.66
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	$x_1 = x_2 = x_4 = s_2 = 0$	$(0, 0, 0, 1/4, 0, 0)$	-1.5
14	x_4, s_2	$x_1 = x_2 = x_3 = s_1 = 0$	$(0, 0, 0, 1/4, 0, 0)$	-1.5
15	s_1, s_2	$x_1 = x_2 = x_3 = x_4 = 0$	$(0, 0, 0, 0, 2, 1)$	0

Here row 7 in the table below has linearly dependent columns. And the number of corners is 6, 12 BFS and 14 BS and $\binom{n}{m} = \binom{6}{2} = 15$.

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

Lemma 5.5. *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 of A corresponding to the nonzeros in x. Then x is an extreme point if and only if $A^{=}_x$ has linearly independent columns (i.e., $A^{=}_x$ has full column rank).*

We will look at an example now in the context of the above lemma.

Example 5.4.

$$\min \ \ 2x_1 + 3x_2$$

such that

$$\begin{aligned} 2x_1 + x_2 &\geq 4 \\ x_1 + 2x_2 &\geq 5 \\ x_1 + x_2 &\geq 3 \\ x_1, x_2 &\geq 0 \end{aligned}$$

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

Now,

<u>At C:</u>, $A^{=} \begin{bmatrix} 2 & 1 \\ 1 & 2 \\ 1 & 1 \end{bmatrix}$, $A^{=}_x \begin{bmatrix} 2 & 1 \\ 1 & 2 \\ 1 & 1 \end{bmatrix}$ Columns 1 and 2 are linearly independent for $A^{=}_x$ and in this case and it is an extreme point.

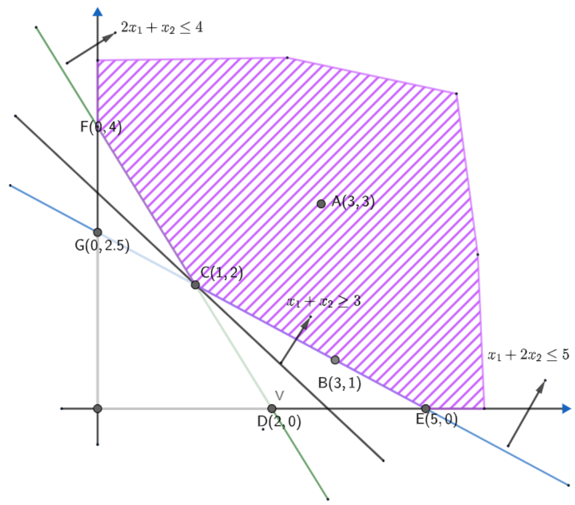


Figure 11: Figure showing geometric view of the above example. The purple region is unbounded.

<u>at F:</u> $A^{=}\begin{bmatrix}2 & 1\end{bmatrix}$, $A^{=}_{x}\begin{bmatrix}1\end{bmatrix}$ Hence, it is an extreme point.

<u>at A:</u> $A^{=}0 \times 0$ matrix, In this case, it is not an extreme point.

<u> at B:</u> $A^{=}\begin{bmatrix}1 & 2\end{bmatrix}$, $A^{=}_{x}\begin{bmatrix}1 & 2\end{bmatrix}$ Columns 1 and 2 are not linearly independent for $A^{=}_{x}$ and in this case, it is not an extreme point.

5.2 Rank Lemma

Lemma 5.6 (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 number of maximal linearly independent 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 5.5 it follows that $A^{=}$ has full column rank. Since the number of rows 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 $A^{=}$ the number of variables. Then any maximal number of linearly independent tight constraints is exactly the number of linearly independent rows of $A^{=}$ which is exactly the row rank of $A^{=}$ and hence the claim follows. ∎

<u>At C:</u>, $x_i > 0\ \forall i$, $A^{=}_{x}\begin{bmatrix}2 & 1\\1 & 2\\1 & 1\end{bmatrix}$, $Rank(A^{=}_{x}) = 2$. So maximal number of linearly independent constraints is 2.

Rank lemma basically says that if no. of variables is n and no. of constraints is $n+m$, we have n constraints that get tight at an extreme point.

If all $x_i > 0$, then all these constraints come from non-trivial constraints. That is, they come from the matrix A. And the n constraints that get tight are linearly independent. Rank lemma is one of the key ingredients in iterative rounding algorithms [LRS11].

References

[SA90] Hanif D Sherali and Warren P Adams. "A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems". In: *SIAM Journal on Discrete Mathematics* 3.3 (1990), pp. 411–430.

[LS91] László Lovász and Alexander Schrijver. "Cones of matrices and set-functions and 0–1 optimization". In: *SIAM journal on optimization* 1.2 (1991), pp. 166–190.

[BCC93] Egon Balas, Sebastián Ceria, and Gérard Cornuéjols. "A lift-and-project cutting plane algorithm for mixed 0–1 programs". In: *Mathematical programming* 58.1 (1993), pp. 295–324.

[Las01a] Jean B Lasserre. "An explicit exact SDP relaxation for nonlinear 0-1 programs". In: *International Conference on Integer Programming and Combinatorial Optimization*. Springer. 2001, pp. 293–303.

[Las01b] Jean B Lasserre. "Global optimization with polynomials and the problem of moments". In: *SIAM Journal on optimization* 11.3 (2001), pp. 796–817.

[LRS11] Lap Chi Lau, Ramamoorthi Ravi, and Mohit Singh. *Iterative methods in combinatorial optimization*. Vol. 46. Cambridge University Press, 2011.

[WS11] David P Williamson and David B Shmoys. *The design of approximation algorithms*. Cambridge university press, 2011.

[GLS12] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization*. Vol. 2. Springer Science & Business Media, 2012.

[Rou12a] Tim Roughgarden. *Tim Roughgarden Lecture notes- A second course in Algorithms: Lecture 10*. 2012. URL: `http://timroughgarden.org/w16/l/l10.pdf`.

[Rou12b] Tim Roughgarden. *Tim Roughgarden Lecture notes- A second course in Algorithms: Lecture 7*. 2012. URL: `http://timroughgarden.org/w16/l/l7.pdf`.

[Rou12c] Tim Roughgarden. *Tim Roughgarden Lecture notes- A second course in Algorithms: Lecture 8*. 2012. URL: `http://timroughgarden.org/w16/l/l8.pdf`.

[Rou12d] Tim Roughgarden. *Tim Roughgarden Lecture notes- A second course in Algorithms: Lecture 9*. 2012. URL: `http://timroughgarden.org/w16/l/l9.pdf`.