**Linear Programming**

**Learning Goals.**
- Introduce Linear Programming Problems.
- Widget Example, Graphical Solution.
- Basic Theory:
  - Feasible Set, Vertices, Existence of Solutions.
  - Equivalent formulations.
- Outline of Simplex Method.
- Runtimes for Linear Program Solvers.

**Readings:** Read text section 11.6, and sections 1 and 2 of Tom Ferguson’s notes (see course homepage).

---

**Widget Factory Example**

A factory makes $x_1$ (thousand) widgets of type 1 and $x_2$ of type 2. Total profit for making $x = (x_1, x_2)^T$ is:

$\text{profit} = x_1 + 2x_2$

Due to a limited resource (e.g. time) we require:

$x_1 + x_2 \leq 4$

Two waste products from making widget 1 are required for widget 2. So we need to make enough of widget 1 to supply the construction of widget 2. These constraints are:

$-x_1 + x_2 \leq 1$

$-3x_1 + 10x_2 \leq 15$

Finally, both $x_1$ and $x_2$ must be non-negative.

How many widgets of each type should be made to maximize profit?

---

**Linear Programming: Standard Form**

Consider a real-valued, unknown, $n$-vector $x = (x_1, x_2, \ldots, x_n)^T$.

A linear programming problem in standard form $(A, b, c)$ has the three components:

- **Objective Function:** We wish to choose $x$ to maximize:
  
  $c^T x = c_1 x_1 + c_2 x_2 + \ldots + c_n x_n$

  with $x$ subject to the following constraints:

- **Problem Constraints:** For an $m \times n$ matrix $A$, and an $m \times 1$ vector $b$:
  
  $A x \leq b$

- **Non-negativity Constraints:**
  
  $x \geq 0$

**Notation:** For two $K$-vectors $x$ and $y$, $x \leq y$ if $x_k \leq y_k$ for each $k = 1, 2, \ldots, K$. Other inequalities ($\geq$, etc.) defined similarly.

---

**Widget Factory Example: Continued.**

Pose Widget problem as a linear program in Standard Form. Need to specify constants, $(A, b, c)$.

**Unknowns:**

$x = (x_1, x_2)^T$ number (in thousands) of the two widget types.

**Objective function (profit):**

$c^T x = c_1 x_1 + c_2 x_2 = x_1 + 2x_2$, so $c^T = (c_1, c_2) = (1, 2)$.

**Problem Constraints:** $A x \leq b$

$\begin{align*}
-x_1 + x_2 & \leq 1, \\
-3x_1 + 10x_2 & \leq 15
\end{align*}$

So $A = \begin{pmatrix} -1 & 1 \\ -3 & 10 \end{pmatrix}$, $b = \begin{pmatrix} 4 \\ 15 \end{pmatrix}$

**Non-negativity Constraints:**

$x \geq 0$
Graphing the Widget Factory Example

Example: \( x = (x_1, x_2) \). Linear Program specified by \((A, b, c)\).

Objective Function: \( c^T x \),
\[ c = (1, 2)^T \]

Non-negativity:
\[ x \geq 0. \]

Problem Constraints:
\[ A \leq b, \]
\[
\begin{pmatrix}
1 & 1 \\
-1 & 1 \\
-3 & 10 \\
\end{pmatrix}
\leq
\begin{pmatrix}
4 \\
1 \\
15 \\
\end{pmatrix}
\]

Direction of increasing profit:
\( c^T x = 4 \)
\( c^T x = 2 \)

Mixtures of widgets with the same profit:
\( c^T x = 4 \)
\( c^T x = 2 \)

Feasible Set:
Max profit: \( c^T x \)
Graphing the Widget Factory Example

Example: \( x = (x_1, x_2)^T \). Linear Program specified by \((A, b, c)\).

Objective Function: \( c^T x \), where \( c = (1, 2)^T \).

Problem Constraints:

\[
A x \leq b,
\]
\[
A = \begin{pmatrix}
1 & 1 \\
-1 & 1 \\
-3 & 10
\end{pmatrix},
\]
\[
b = \begin{pmatrix}
4 \\
1 \\
15
\end{pmatrix}.
\]

Non-negativity: \( x \geq 0 \).

Optimal Solution: \( x^* = (25/13, 27/13) \).

Problem Constraints:

\[
A x \leq b,
\]
\[
A = \begin{pmatrix}
1 & 1 \\
-1 & 1 \\
-3 & 10
\end{pmatrix},
\]
\[
b = \begin{pmatrix}
4 \\
1 \\
15
\end{pmatrix}.
\]

Linear Programming: Example Applications

Linear programming is quite a general framework. For example:

Network flow problems can be written as linear programs (LPs) for the unknown flow \( f(e) \). The constraints are:

\* 0 \leq f(e) \leq c(e)
\* flow conservation at each \( v \in V \setminus \{s, t\} \).

\[
\sum_{e \in \partial^+ v} f(e) - \sum_{e \in \partial^- v} f(e) = 0
\]

which can be rewritten as \( \text{LHS} \geq 0 \) and \( \text{LHS} \leq 0 \).

The objective function is max sum \( f(e) \mid e = (s, v), v \in V \) \).

Integer Linear Programming. Weighted scheduling problems, the Knapsack problem, etc. can also be written as LPs, although for these we seek integer valued solutions.

Linear Programming Theory: Feasible Set

Given the constants \((A, b, c)\), consider the linear program:

**Objective Function:** Maximize \( c^T x \), where \( x = (x_1, x_2, \ldots, x_n)^T \).

**Problem Constraints:** \( A x \leq b \)

**Non-negativity Constraints:** \( x \geq 0 \)

Define the feasible set \( F = \{ x : A x \leq b \text{ and } x \geq 0 \} \).

\( F \) could be empty (no solutions to all the constraints).

\( F \) is a convex polytope. (A region of \( \mathbb{R}^n \) defined by the intersection of finitely many half-spaces, e.g., \( a_i^T x \leq b_i \) and \( x \geq 0 \).)

**Convexity of \( F \):** Let \( u, v \in F \), and let \( s \in [0, 1] \). Then \( su + (1-s)v \in F \).

**PF:** \( u, v \in F \) implies \( Au \leq b \) and \( Av \leq b \). So \( A [su + (1-s)v] = s Au + (1-s)Av \leq sb + (1-s)b = b \), for \( s \in [0, 1] \). A similar argument shows \( su + (1-s)v \geq 0 \). Therefore \( su + (1-s)v \in F \).
Linear Programming Theory: Characterization of a Vertex

What's a vertex of the feasible set?

Let $P$ be the $(m+n) \times n$ matrix and $p$ the $(m+n)$-vector which represents both the problem and non-negativity constraints:

$$Px \equiv \begin{pmatrix} A & -1 \end{pmatrix} x \leq \begin{pmatrix} b \\ 0 \end{pmatrix}.$$

Let $s = \{s_1, s_2, \ldots, s_n\}$ be a selection of $n$ row numbers, $1 \leq s_i \leq m+n$. Define $Q(s)$ to be the $n \times n$ matrix formed from the $s$-rows of $P$, and $q(s)$ the $n$-vector formed from the same rows of $p$.

**Defn:** A point $v \in \mathbb{R}^n$ is a vertex of the feasible set $F$ iff there exists an $s$ such that:

- $Q(s)$ is nonsingular,
- $v = [Q(s)]^{-1} q(s)$, i.e., $v$ satisfies the $n$ equalities selected by $s$, and
- $v \in F$, i.e., $v$ is feasible.

See 2D examples above, and 3D example next.
**Linear Programming Theory: Characterization of a Solution**

Given the constants \((A, b, c)\), consider the linear program:

**Objective Function:** Maximize \(c^T x\), where \(x = (x_1, x_2, ..., x_n)^T\), \(c \neq 0\).

**Problem Constraints:** \(A x \leq b\)

**Non-negativity Constraints:** \(x \geq 0\)

**Define:** Feasible set \(F = \{ x \mid A x \leq b \text{ and } x \geq 0 \}\)

**Theorem:** The linear program above either:

1. has no solution, in which case either
   - the feasible set \(F\) is empty, or
   - the objective \(c^T x\) unbounded (and \(F\) is unbounded),
2. has a solution \(x^*\).

In case 2, \(x^*\) can be taken to be a vertex of the polytope \(F\) (there may also be non-vertex solutions \(x \in F\) with \(c^T x = c^T x^*\))

See examples.
### Three Dimensional Example

Colours indicate value of $c^T x$. Larger $c^T x$ → hotter colours.

Feasible set $F$ is this closed polytope.

Maximize $c^T x$, $c^T = (2, 3, 4)$.

Multiple solns iff $c$ is perpendicular to an edge or face of $F$.

### Variations in the Formulation of Linear Programs

A given LP can be expressed in many equivalent forms.

Given an LP in standard form, with constants $(A, b, c)$.

Some alternatives:
- minimize $-c^T x$ equivalent to maximizing $c^T x$.
- Constraint $a^T x \leq b$ equivalent to $-a^T x \geq -b$.
- Constraint $a^T x = b$ iff $a^T x \leq b$ and $-a^T x \leq -b$.
- $Ax \leq b$ iff $Ax + z = b$ and $z \geq 0$, with "slack variables" $z = (z_1, \ldots, z_m)^T$.
- $\min |e|$ can be rewritten as $\min (e^+ - e^-)$ with the linear constraint $e = e^+ - e^-$ and the non-negativity constraint $e^+, e^- \geq 0$.

These alternatives are useful for posing problems as LPs in standard form, or reposing LPs in alternative forms useful for computation.

### A Sketch of the Simplex Method

**Simplex method:** Given an LP in standard form $(A, b, c)$. Let $P$ and $p$ be:

$$P x \equiv \begin{pmatrix} A \\ -I \end{pmatrix} x \leq p \equiv \begin{pmatrix} b \\ 0 \end{pmatrix}.$$ 

Let $v$ be a feasible vertex. So $v = v(s)$, where $s = (s_1, \ldots, s_n)$ denotes a set of $n$ selected rows of $P \cup p$, such that $Q(s) v = q(s)$ and $Q(s)$ is nonsingular (see the defn. of a vertex, above).

while true

* Consider each neighbour $s'$ of $s$ (i.e., $s'$ and $s$ differ only in one element).
* Choose an edge $v(s)$ to $v(s')$ s.t. the objective increases.
* If there is no such edge, $v(s)$ is a solution. Stop.
* If there is an edge leaving $v(s)$ on which the objective is unbounded, then there is no solution to this LP. Stop.
* Set $s \leftarrow s'$, $v \leftarrow v(s')$

end

* Modulo non-cycling conditions

### Three Dimensional Example: Revisited

Problem Constraints:
- $x_1 + 3x_3 \leq 600$
- $x_2 + x_3 \leq 300$
- $x_1 + x_2 + x_3 \leq 400$
- $x_2 \leq 250$

Maximize $c^T x$, $c^T = (2, 3, 4)$.

Feasible set $F$ is this closed polytope.

Multiple solns iff $c$ is perpendicular to an edge or face of $F$. 

- $c^T x$ gives shading.
Pivoting

The step from \( v(s) \) to \( v(s') \) is called pivoting.

One row of \( P v \leq p \) is dropped from \( s \), and it is replaced by another row

to form \( s' \).

The selection of a pivot is guaranteed not to decrease the objective
function.

If some care is taken to avoid cycling, the Simplex Algorithm is

guaranteed to converge to a solution after finitely many pivot steps.

In an efficient implementation, each pivot step costs \( O((m+n)n) \) real
number operations.

Unfortunately, simplex may visit exponentially many vertices in
contrived cases. E.g., number of choices for \( s \), \( (n+m) \) choose \( n \).

Obtaining an Initial Vertex

We need an initial feasible vertex to start the Simplex Algorithm.

Given the LP constants \((A, b, c)\), consider the start-up LP:

Objective Function: Maximize \(-z_1 - \ldots - z_m\), where \( z = (z_1, z_2, \ldots, z_m)\)

Equivalent to minimizing \( z_1 + \ldots + z_m \)

Problem Constraints: \( A x - z \leq b \),
Non-negativity Constraints: \( x, z \geq 0 \)

For this start-up LP we have the initial guess, \( x = 0, z = b^- \) where \( b^- = b_k \)
if \( b_k < 0 \) and \( 0 \) otherwise.

This start-up LP has a solution \((x_0, 0)\) (i.e., with \( z = 0 \) and the objective
function equal to \( 0 \)) iff \( x_0 \) is a feasible solution of the original LP.

Simplex will return a feasible vertex \( x_0 \) on this start-up LP, so long as
the original feasible set \( F \) is not empty.

Runtime for Simplex Algorithm

Worst case runtime is exponential. The Simplex Algorithm might visit
exponentially many vertices as \( m \) and \( n \) grow.

In practice:

- the method is highly efficient,
- typically requires a number of steps which is just a small multiple of
  the number of variables,
- LPs with thousands or even millions of variables are routinely solved
  using the simplex method on modern computers.
- efficient, highly sophisticated implementations are available.

Runtime for Linear Programming Solvers

Interior point methods provided the first polynomial time algorithms
known for LP.

These iterate through the interior of the feasible set \( F \).

- Interior point projective method, Karmarkar, 1984.

Interior point methods are now generally considered competitive with the
simplex method in most, though not all, applications.

Sophisticated software packages are available.

Integer Linear Programming (ILP): An LP problem but with the added
constraint that the solution vector \( x \) must be integer valued.
ILP is NP-hard.
Linear Programming

Learning Goals.
- Basic Linear Programming Theory:
  - Feasible Set, Vertices, Existence of Solutions.
  - Equivalent formulations.
- Outline of the Simplex Method.
- Runtimes

Readings: Read text section 11.6, and sections 1 and 2 of Tom Ferguson's notes (see course homepage).

Next Lectures: Duality in Linear Programs (sec. 2, Ferguson's notes).
Begin approximation algorithms, Chapter 11.