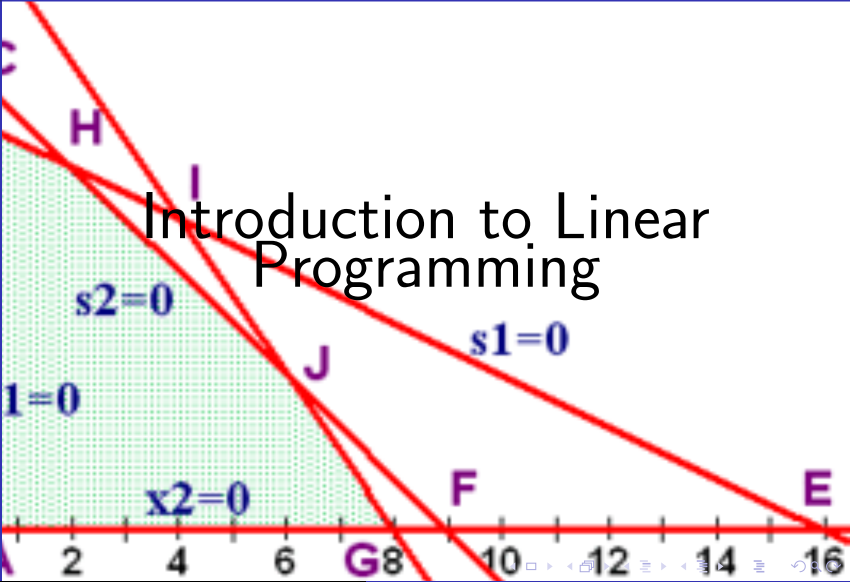
Introduction to Linear Programming



Linear Programming.

In a linear programming problem we are given a set of variables, an objective function a set of linear constrains and want to assign real values to the variables as to:

- satisfy the set of linear equations,
- maximize or minimize the objective function.

LP is of special interest because many combinatorial optimization problems can be reduced to LP: Max-Flow; Assignment problems; Matchings; Shortest paths; MinST; ...

Example.

A company produces 2 products P1, and P2, and wishes to maximize the profits.

Each day, the company *can produce* x_1 units of P1 and x_2 units of P2.
The company *makes a profit* of 1 for each unit of P1; and a profit of 6 for each unit of P2.

Due to supply limitations and labor constrains we have the following additional constrains: $x_1 \leq 200, x_2 \leq 300$ and $x_1 + x_2 \leq 400$.

What are the best levels of production?

We format this problem as a linear program:

$$
\begin{aligned}
\text{Objective function:} \quad & \max(x_1 + 6x_2) \\
\text{subject to the constraints:} \quad & x_1 \leq 200 \\
& x_2 \leq 300 \\
& x_1 + x_2 \leq 400 \\
& x_1, x_2 \geq 0.
\end{aligned}
$$

We format this problem as a linear program:

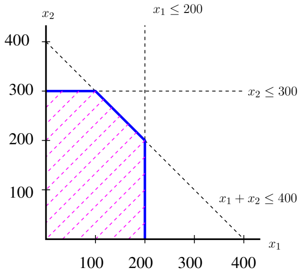
$$\begin{aligned} \text{Objective function:} \quad & \max(x_1 + 6x_2) \\ \text{subject to the constraints:} \quad & x_1 \leq 200 \\ & x_2 \leq 300 \\ & x_1 + x_2 \leq 400 \\ & x_1, x_2 \geq 0. \end{aligned}$$

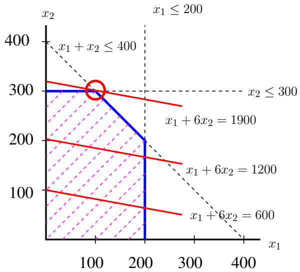
Recall a linear equation in x_1 and x_2 defines a line in $\mathbb{R}^2$. A linear inequality define a half-space.

We format this problem as a linear program:

$$\begin{aligned}
\text{Objective function:} \quad & \max(x_1 + 6x_2) \\
\text{subject to the constraints:} \quad & x_1 \leq 200 \\
& x_2 \leq 300 \\
& x_1 + x_2 \leq 400 \\
& x_1, x_2 \geq 0.
\end{aligned}$$

Recall a linear equation in x_1 and x_2 defines a line in $\mathbb{R}^2$. A linear inequality define a half-space.
The feasible region of this LP are the (x_1, x_2) in the convex polygon defined by the linear constrains.


x_2
$x_1 \leq 200$
400
300
200
100
$x_2 \leq 300$
$x_1 + x_2 \leq 400$
x_1
100
200
300
400


x_2
$x_1 \leq 200$
400
$x_1 + x_2 \leq 400$
300
$x_2 \leq 300$
$x_1 + 6x_2 = 1900$
200
$x_1 + 6x_2 = 1200$
100
$x_1 + 6x_2 = 600$
x_1
100
200
300
400

In a linear program *the optimum is achieved at a vertex of the feasible region*.

A LP is infeasible if

- The constrains are so tight that there are impossible to satisfy all of them. For ex. $x \geq 2$ and $x \leq 1$,
- The constrains are so loose that the feasible region is unbounded. For ex. $\max(x_1 + x_2)$ with $x_1, x_2 \geq 0$

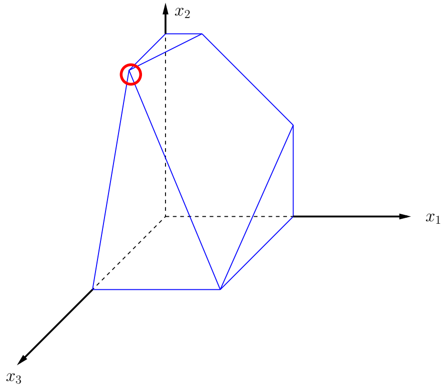
Higher dimensions.

The company produces products P1, P2 and P3, where each day it produces x_1 units of P1, x_2 units of P2 and x_3 units of P3. and makes a profit of 1 for each unit of P1, a profit of 6 for each unit of P2 and a profit of 13 for each unit of P3. Due to supply limitations and labor constrains we have the following additional constrains: $x_1 \leq 200, x_2 \leq 300$, $x_1 + x_2 + x_3 \leq 400$ and $x_2 + 3x_3 \leq 600$.

Higher dimensions.

The company produces products P1, P2 and P3, where each day it produces x_1 units of P1, x_2 units of P2 and x_3 units of P3. and makes a profit of 1 for each unit of P1, a profit of 6 for each unit of P2 and a profit of 13 for each unit of P3. Due to supply limitations and labor constrains we have the following additional constrains: $x_1 \leq 200, x_2 \leq 300$, $x_1 + x_2 + x_3 \leq 400$ and $x_2 + 3x_3 \leq 600$.

$$\begin{aligned} \max(x_1 + 6x_2 + 13x_3) \\ x_1 &\leq 200 \\ x_2 &\leq 300 \\ x_1 + x_2 + x_3 &\leq 400 \\ x_2 + 3x_3 &\leq 600 \\ x_1, x_2, x_3 &\geq 0. \end{aligned}$$


x_2
x_1
x_3

Standard form of a Linear Program.

INPUT: Given real numbers $(c_i)_{i=1}^n, (a_{ji})_{1\leq j\leq m \& 1\leq i\leq n}(b_i)_{i=1}^n$
OUTPUT: real values for variables $(x_i)_{i=1}^n$
A linear programming problem is the problem or maximizing (minimizing) a linear function the objective function $P = \sum_{i=1}^n c_i x_j$ subject to finite set of linear constraints

Standard form of a Linear Program.

INPUT: Given real numbers $(c_i)_{i=1}^n, (a_{ji})_{1\leq j\leq m \& 1\leq i\leq n}(b_i)_{i=1}^n$
OUTPUT: real values for variables $(x_i)_{i=1}^n$
A linear programming problem is the problem or maximizing (minimizing) a linear function the objective function $P = \sum_{i=1}^n c_i x_j$ subject to finite set of linear constraints

max $\sum_{i=1}^n c_i x_j$,
subject to:
$\sum_{i=1}^n a_{ji} x_i = b_j$, $1 \leq j \leq m$
$x_i \geq 0$, $1 \leq i \leq n$

A LP is in standard form if the following are true:

- Non-negative constraints for all variables.
- All remaining constraints are expressed as $=$ constraints.
- All $b_i \geq 0$.

Equivalent formulations of LP.

A LP has many degrees of freedom:

1. It can be a maximization or a minimization problem.
2. Its constrains could be equalities or inequalities.
3. The variables are often restricted to be non-negative, but they also could be unrestricted.

Most of the *"real life"* constrains are given as inequalities.
The main reason to convert a LP into standard form is because the simplex algorithm starts with a LP in standard form.
But it could be useful the flexibility to be able to change the formulation of the original LP.

Transformations among LP forms

- *To convert inequality* $\sum_{i=1}^{n} a_i x_i \leq b_i$ *into equality:*

Transformations among LP forms

- *To convert inequality $\sum_{i=1}^{n} a_i x_i \leq b_i$ into equality:* introduce a slack variable s_i to get $\sum_{i=1}^{n} a_i x_i + s_i = b_i$ with $s \geq 0$.
  The slack variable s_i measures the amount of "non-used resource."
  Ex: $x_1 + x_2 + x_3 \leq 40 \Rightarrow x_1 + x_2 + x_3 + s_1 = 40$
  So that $s_1 = 40 - (x_1 + x_2 + x_3)$

Transformations among LP forms

- *To convert inequality $\sum_{i=1}^{n} a_i x_i \leq b_i$ into equality:* introduce a slack variable s_i to get $\sum_{i=1}^{n} a_i x_i + s_i = b_i$ with $s \geq 0$.
  The slack variable s_i measures the amount of "non-used resource."
  Ex: $x_1 + x_2 + x_3 \leq 40 \Rightarrow x_1 + x_2 + x_3 + s_1 = 40$
  So that $s_1 = 40 - (x_1 + x_2 + x_3)$
- *To convert inequality $\sum_{i=1}^{n} a_i x_i \geq -b$ into equality:*

Transformations among LP forms

- *To convert inequality $\sum_{i=1}^{n} a_i x_i \leq b_i$ into equality:* introduce a slack variable s_i to get $\sum_{i=1}^{n} a_i x_i + s_i = b_i$ with $s \geq 0$.
  The slack variable s_i measures the amount of "non-used resource."
  Ex: $x_1 + x_2 + x_3 \leq 40 \Rightarrow x_1 + x_2 + x_3 + s_1 = 40$
  So that $s_1 = 40 - (x_1 + x_2 + x_3)$
- *To convert inequality $\sum_{i=1}^{n} a_i x_i \geq -b$ into equality:* introduce a surplus variable and get $\sum_{i=1}^{n} a_i x_i - s_i = b_i$ with $s \geq 0$.
  The surplus variable s_i measures the extra amount of used resource.
  Ex: $-x_1 + x_2 - x_3 \geq 4 \Rightarrow -x_1 + x_2 - x_3 - s_1 = 4$

Transformations among LP forms (cont.)

- *To to deal with an unrestricted variable x (i.e. x can be positive or negative):*

Transformations among LP forms (cont.)

- *To to deal with an unrestricted variable x (i.e. x can be positive or negative):* introduce $x^+, x^- \geq 0$, and replace all occurrences of x by $x^+ - x^-$.
  Ex: x unconstrained $\Rightarrow x = x^+ - x^-$ with $x^+ \geq 0$ and $x^- \geq 0$.

Transformations among LP forms (cont.)

- *To to deal with an unrestricted variable x (i.e. x can be positive or negative):* introduce $x^+, x^- \geq 0$, and replace all occurrences of x by $x^+ - x^-$.
  Ex: x unconstrained $\Rightarrow x = x^+ - x^-$ with $x^+ \geq 0$ and $x^- \geq 0$.
- *To turn max. problem into min. problem:*

Transformations among LP forms (cont.)

- *To to deal with an unrestricted variable x (i.e. x can be positive or negative):* introduce $x^+, x^- \geq 0$, and replace all occurrences of x by $x^+ - x^-$.
  Ex: x unconstrained $\Rightarrow x = x^+ - x^-$ with $x^+ \geq 0$ and $x^- \geq 0$.

- *To turn max. problem into min. problem:* multiply the coefficients of the objective function by -1.
  Ex: $\max(10x_1 + 60x_2 + 140x_3) \Rightarrow$ $\min(-10x_1 - 60x_2 - 140x_3)$.

Transformations among LP forms (cont.)

- *To to deal with an unrestricted variable x (i.e. x can be positive or negative):* introduce $x^+, x^- \geq 0$, and replace all occurrences of x by $x^+ - x^-$.
  Ex: x unconstrained $\Rightarrow x = x^+ - x^-$ with $x^+ \geq 0$ and $x^- \geq 0$.
- *To turn max. problem into min. problem:* multiply the coefficients of the objective function by -1.
  Ex: $\max(10x_1 + 60x_2 + 140x_3) \Rightarrow$
  $\min(-10x_1 - 60x_2 - 140x_3)$.

Applying these transformations, we can rewrite any LP into standard form, in which variables are all non-negative, the constrains are equalities, and the objective function is to be minimized.

Example:

$$\begin{aligned}
\max(10x_1 + 60x_2 + 140x_3)& \\
x_1 &\le 20 \\
x_2 &\le 30 \\
x_1 + x_2 + x_3 &\le 40 \\
x_2 + 3x_3 &\le 60 \\
x_1, x_2, x_3 &\ge 0.
\end{aligned}$$

$$\begin{aligned}
\min(-10x_1 - 60x_2 - 140x_3)& \\
x_1 + s_1 &= 20 \\
x_2 + s_2 &= 30 \\
x_1 + x_2 + x_3 + s_3 &= 40 \\
x_2 + 3x_3 + s_5 &= 60 \\
x_1, x_2, x_3, s_1, s_2, s_3, s_4, s_5 &\ge 0.
\end{aligned}$$

Algebraic representation of LP

Let $c = (c_1, \ldots, c_n)$ $x = (x_1, \ldots, x_n)$, $b = (b_1, \ldots, b_m)$ and let A be the $m \times n$ matrix of the coefficients involved in the constrains.

A LP can be represented using matrix and vectors:

$$
\begin{array}{lcl}
\max \sum_{i=1}^{n} c_i x_j & & \max \sum_{i=1}^{n} c^T x \\
\text{subject to} & \Rightarrow & \text{subject to} \\
\sum_{i=1}^{n} c_i x_j \leq b_j, \; 1 \leq j \leq m & & Ax \leq b \\
x_i \geq 0, \; 1 \leq i \leq n & & x \geq 0
\end{array}
$$

Given a LP

$$\begin{aligned} &\min\ c^T x \\ &\text{subject to} \\ &\quad Ax \geq b \\ &\quad x \geq 0 \end{aligned}$$

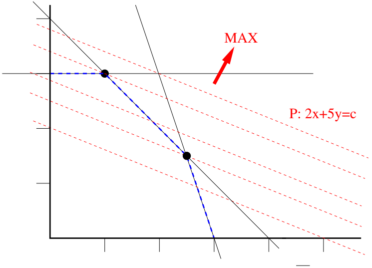
Any x that satisfies the constraints is a *feasible solution*.
A LP is *feasible* if there exists a feasible solution. Otherwise is said to be *infeasible*.
A feasible solution x^* is an optimal solution if

$$c^T x^* = \min\{c^T x \,|\, Ax \geq b, x \geq 0\}$$

The Geometry of LP

Consider P:

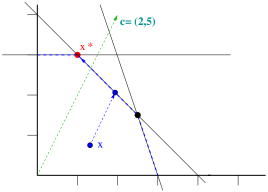
$$\begin{aligned} \max 2x{+}5y \\ 3x + y &\leq 9 \\ y &\leq 3 \\ x + y &\leq 4 \\ x, y &\geq 0 \end{aligned}$$



Theorem

If there exists an optimal solution to P, x, then there exists one that is a vertex of the polytope.

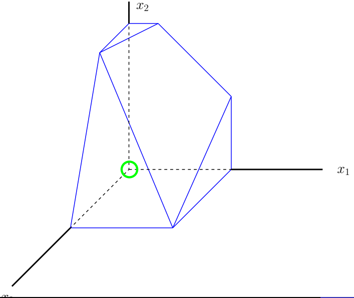
Intuition of proof If x is not a vertex, move in a non-decreasing direction until reach a boundary. Repeat, following the boundary.

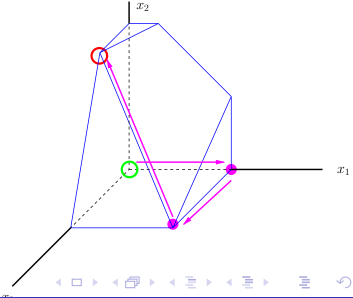


The Simplex algorithm

LP can be solved efficiently: George Dantzing (1947)

It uses hill-climbing: Start in a vertex of the feasible polytope and look for an adjacent vertex of better objective value. Until reaching a vertex that has no neighbor with better objective function.





Complexity of LP:

Input to LP: The number n of variables in the LP.

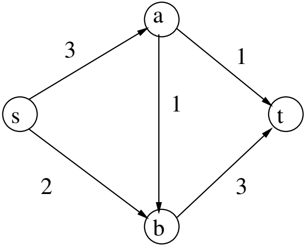
Simplex could be exponential on n: there exists specific input (the Klee-Minty cube) where the usual versions of the simplex algorithm may actually "cycle" in the path to the optimal. (see Ch.6 in Papadimitriou-Steiglitz, *Comb. Optimization: Algorithms and Complexity*)

In practice, the simplex algorithm is quite efficient and can find the global optimum (if certain precautions against cycling are taken).

It is known that simplex solves "typical" (random) problems in $O(n^3)$ steps.
Simplex is the main choice to solve LP, among engineers.

But some software packages use interior-points algorithms, which guarantee poly-time termination.

Example: The Max-Flow problem



$$
\begin{aligned}
\max\ f_{sa} + f_{sb}& \\
f_{sa} &\leq 3 \\
f_{sb} &\leq 2 \\
f_{ab} &\leq 1 \\
f_{at} &\leq 1 \\
f_{bt} &\leq 3 \\
f_{sa} - f_{ab} - f_{at} &= 0 \\
f_{sb} + f_{ab} - f_{bt} &= 0 \\
f_{sa}, f_{sb}, f_{ab}, f_{at}, f_{bt} &\geq 0.
\end{aligned}
$$

The Min Cut problem

$$\begin{aligned}
\min\ \ 3y_{sa} + 2y_{sb} + y_{ab} + y_{at} + y_{bt} & \\
y_{sa} + u_a &\geq 1 \\
y_{sb} + u_b &\geq 1 \\
y_{ab} - u_a + u_b &\geq 0 \\
y_{at} - ua &\leq 1 \\
y_{bt} - u_b &\leq 3 \\
y_{sa}, y_{sb}, y_{ab}, y_{at}, y_{bt}, u_a, u_b &\geq 0.
\end{aligned}$$

This D - LP defines the min-cut problem where for $x \in \{a, b\}$, $u_x = 1$ iff vertex $x \in S$, and $y_{xz} = 1$ iff $(x, z) \in$ cut (S, T).

Min cost maximum flow

Min Cost MaxFlow: Given a flow network and a valuation of the cost of transporting a unit of flow along each edge. Find a maximum flow with minimum cost.

Min cost maximum flow

Min Cost MaxFlow: Given a flow network and a valuation of the cost of transporting a unit of flow along each edge. Find a maximum flow with minimum cost.

- Compute the value F of a maximum flow.
- Adapt the LP for MaxFlow to ensure that the flow value is F and incorporate the cost in the objective function.
  Add the equation $f(s, V) = F$
  Objective function: minimize $\sum_{e \in E} c_e f_e$
- This approach provides a polynomial time algorithm for the problem.

Integer Linear Programming (ILP)

Consider the Min Vertex Cover problem: Given an undirected $G = (V, E)$ with $|V| = n$ and $|E| = m$, want to find $S \subseteq V$ with minimal cardinality s.t.. it covers all edges $e \in E$.

Integer Linear Programming (ILP)

Consider the Min Vertex Cover problem: Given an undirected $G = (V, E)$ with $|V| = n$ and $|E| = m$, want to find $S \subseteq V$ with minimal cardinality s.t.. it covers all edges $e \in E$.

- This problem can be expressed as a linear program on $\{0, 1\}$ variables, interpreting a solution as
  Let $x \in \{0, 1\}^n$ be seen as a set S, in the usual way, for $i \in V$:

$$x_i = \begin{cases} 1 & \text{if } i \in S \\ 0 & \text{otherwise} \end{cases}$$

Integer Linear Programming (ILP)

Consider the Min Vertex Cover problem: Given an undirected $G = (V, E)$ with $|V| = n$ and $|E| = m$, want to find $S \subseteq V$ with minimal cardinality s.t.. it covers all edges $e \in E$.

- This problem can be expressed as a linear program on $\{0, 1\}$ variables, interpreting a solution as
  Let $x \in \{0, 1\}^n$ be seen as a set S, in the usual way, for $i \in V$:
  $$x_i = \begin{cases} 1 & \text{if } i \in S \\ 0 & \text{otherwise} \end{cases}$$
- Under this interpretation we the constraints $\forall (i, j) \in E$ $x_i + x_j \geq 1$ are equivalent to say that S is a vertex cover. The constraints give $Ax \geq 1$.

Integer Linear Programming

We can express the min VC problem as:

min $\sum_{i \in V} x_i$
subject to
$x_i + x_j \geq, (i, j) \in E$
$x_i \in \{0, 1\}), i \in V$

Integer Linear Programming

We can express the min VC problem as:

$$\begin{aligned}
&\min \textstyle\sum_{i \in V} x_i \\
&\text{subject to} \\
&\quad x_i + x_j \geq, \ (i,j) \in E \\
&\quad x_i \in \{0,1\}), \ i \in V
\end{aligned}$$

where we have a new constrain, we require the solution to be 0,1. This can be replaced by requiring the variables to be positive integers (as we are minimizing).

Asking for the best possible integral solution for a LP is known as the Integer Linear Programming:

Integer Linear Programming

The ILP problem is defined:
Given $A \in \mathbb{Z}^{n \times m}$ together with $b \in \mathbb{Z}^n$ and $c \in \mathbb{Z}^m$, find a x that max (min) c^T subject to:

min $c^T x$
subject to
$Ax \geq 1$
$x \in \mathbb{Z}^m$,

Integer Linear Programming

The ILP problem is defined:
Given $A \in \mathbb{Z}^{n \times m}$ together with $b \in \mathbb{Z}^n$ and $c \in \mathbb{Z}^m$, find a x that max (min) c^T subject to:

$$\min\ c^T x$$
subject to
$$Ax \geq 1$$
$$x \in \mathbb{Z}^m,$$

Big difference between LP and ILP:
Ellipsoidal/Interior point methods solve LP in polynomial time but ILP is NP-hard.

Solvers for LP

Due to the importance of LP and ILP as models to solve optimization problem, there is a very active research going on to design new algorithms and heuristics to improve the running time for solving LP (algorithms) IPL (heuristics).

There are a myriad of solvers packages:

- CPLEX: http://ampl.com/products/solvers/solvers-we-sell/cplex/
- GUROBI Optimizer: http://www.gurobi.com/products/gurobi-optimizer