Linear Programming (LP) — Chap. 29 — 29.1

Many problems can be formulated as max/min an objective function subject to a set of constraints.

**Linear Programming Problem:**

- Objective: linear function
- Constraints: linear equalities/inequalities
- If d variables are involved — d-dimensional problem

- Linear function: \( f(x_1, x_2, \ldots, x_d) = a_1 x_1 + a_2 x_2 + \ldots + a_d x_d \)
- Linear equality: \( f(x_1, x_2, \ldots, x_d) = b \)
- Inequality: \( f(x_1, x_2, \ldots, x_d) < b \) \( \geq b \)

*Note: strict inequalities not allowed!!*

**General LP form:** maximize/minimize linear function of d variables subject to m constraints (linear inequalities)

**General Dimension LP:** d variables subject to m constraints

- If \( x = (x_1, x_2, \ldots, x_d) \) satisfies all constraints, \( x \) is a feasible solution.
- The set of all feasible solutions is a convex region — feasible domain/region
Example:

\[
\begin{align*}
\text{max} & \quad x_1 + x_2 \\
\text{at} & \quad 4x_1 - x_2 \leq 8 \\
& \quad 2x_1 + x_2 \leq 10 \\
& \quad 5x_1 - 2x_2 \geq -2 \\
& \quad x_1, x_2 \geq 0
\end{align*}
\]

* Note: optimal solution at a vertex!

LP input:

- set of constraints: linear inequalities
- linear objective function
- \( \text{min} / \text{max} \) \( n \times \) constraints

\( d \Rightarrow \text{Max} / \text{Min} \) \( c_1 x_1 + c_2 x_2 + \ldots + c_n x_n \)

\[\begin{align*}
\text{a.t.:} \quad a_{11} x_1 + a_{12} x_2 + \ldots + a_{1n} x_n & \leq b_1 \\
& \vdots \\
& a_{m1} x_1 + a_{m2} x_2 + \ldots + a_{mn} x_n & \leq b_m
\end{align*}\]

Can think of objective function as a vector:

\[C = (c_1, c_2, \ldots, c_n)\]

* find vertex of feasible region for linear function in direction \( C \) (-C)
\[ l \in \mathbb{R} \setminus \mathbb{Z} \]

- If \( F \) is empty \( \Rightarrow \) \( L \) is infeasible (no solution)
- If \( F \) unbounded on \( \mathbb{R} \) \( \Rightarrow \) \( L \) is unbounded (no solution)
- Possible \( \infty \) number of optimal solutions

Methods

- for high dimensional \( L \) \( \Rightarrow \) simplex & interior point methods
  - Simplex: find a vertex on \( F \), then walk edge by edge on boundary until reaching a local
    maximum \( \Rightarrow \) by convexity in global maximum.
    - exponential in worst case but fast in practice
  - Interior point: polynomial time algorithm
    - move through interior of \( F \)
    - polynomial in \( n, d \), \( 2 \)-etkin numbers
    - \( C(n) \) : strong poly. algo : do not depend on \( \# \) bits

In 2-3:

- \( O(n \cdot \log(n)) \) : find \( F \) then identify vertex
  - hope to do better by avoiding to compute \( F \).
- \( O(n) \) possible for any 1-d \( L \), constant
  \(-\sqrt{\Omega(\# \text{ in } F)}\)
"Unbounded LP" can find if unbounded in O(n) time or find a pair of bounding halfplanes in O(n).

Extends to higher dimensions.

For 2-D, consider outward normal vectors to halfplanes.

- Partition unit circle of directions in a angular sector.
- Consider sector containing C.

\[
\begin{align*}
\text{if } u < 180^\circ \text{ then bounded (if not empty)} \\
\text{(the 2 halfplanes converge at a point)}
\end{align*}
\]
Linear Time LP $\Rightarrow$ 2-D

- $\text{min: } ax + by$
- $\text{s.t.: } a_i x + b_i y \leq c_i, i = 1, 2, \ldots, m$

Change of variables:

\[
\begin{align*}
\begin{aligned}
x & = x \\
y & = c_i - \frac{a_i}{b_i} x
\end{aligned}
\end{align*}
\]

$\Rightarrow$ find feasible point with min $y$-coord.

$\Rightarrow$ want min feasible point on $F_i$.
Observations:
- Solution of max value of FR - lin. integer problem
- Consider only x values when 2 constraints intersect

For \( F_+ \rightarrow 0(n) \) time to find min intersection point \( x \)
- Same for \( F_- \) (max intersection)

In \( O(n) \) time can decide:
1. \( x \) not feasible and all feasible points to right/left of \( x \)
2. Need slope of \( F_+ \), \( F_- \) at \( x \)
   - at most 4 identify constraints of \( F_+ \), \( F_- \) to left/right of \( x \)
   - slope: \( \frac{d_2}{c} \)

(and the symmetric cases)
2. If feasible and optimal $x^*$ is to the right (left) of $x$, 
   (and the symmetric case)

3. No solution to $LP$
   $F_+$ below $F_-$

4. $x$ is in the min of $F_-$ => done!

At each step, make a smart choice of $x$ and decide which side
optimal $x^*$ is.

Decision allows to discard (prune) a "large" number of
constraints.
- \( F_+ \rightarrow S_1 \\
- \ F_- \rightarrow S_2 \\
- \) pair constraints in each set, \( S_1 \) and \( S_2 \) and try to eliminate
one constraint in each pair by comparing \( x_y \) with \( x \)

\[ x_y = \text{abscissa of intersection point of pair } h_i, h_j \]

- \( \rightarrow \text{optimum} \)

\[ a \]

\[ b \]

\[ c \]

\[ d \]

\[ e \]

\[ f \]

\[ g \]

\[ h \]

\[ i \]

\[ j \]

\[ k \]

\[ l \]

\[ m \]

\[ n \]

\[ o \]

\[ p \]

\[ q \]

\[ r \]

\[ s \]

\[ t \]

\[ u \]

\[ v \]

\[ w \]

\[ x \]

\[ y \]

\[ z \]

\[ \text{Analysis:} \]
\[ T(n) = T(n^{1/2}) + O(n) \]
\[ \Rightarrow \]
\[ T(n) = O(n) \]

- \( \text{linear time to put } S_1, S_2 \)
- \( \text{linear time to pair constraints} \)
- \( \text{linear time to compute intersection at } x \) and decide
  which side to first for \( x^* \)
- \( \text{how to choose } x^*? \)
  - \( \text{take median of intersection points of pairs } h_i, h_j \)
  - \( \leq \left \lfloor \frac{n}{2} \right \rfloor \) pairs = \( \frac{n^2}{4} \) points
  - \( \text{find median = } O(1) \) time
  - \( x^* \text{-coord.} \)
  - \( \text{locate side} \)
  - \( \text{for each intersection point on "next" side, eliminate a} \)
  - \( \text{constraint } \Rightarrow O(n^{1/4}) \) eliminated