Linear Programming

The following writeup focuses on an important special case of contrained optimization where all both the objective function and constraints contain only linear operations. This is commonly referred to as a Linear Programming (LP) and is expressed in ***standard form*** as

$$\max_x \quad f = \mathbf{c}^T\mathbf{x}$$

$$\text{s.t.}$$

$$\mathbf{Ax} = \mathbf{b}$$

$$\mathbf{x} \geq \mathbf{0}$$

where $\mathbf{x}$ is a vector of dimension n, $\mathbf{c}$ is a constant n dimensional vector, $\mathbf{A}$ is a constant m by n matrix, and $\mathbf{b}$ is a constant m dimensional vector. Given the dimensionality of A, three types of feasible regions are formed:

1. $\mathbf{n} < \mathbf{m}$, the problem is overspecified with more equations n than variables m and therefore no exact solution as all the equations cannot be satisfied simultaneously.
2. $\mathbf{n} = \mathbf{m}$, the problem is exactly specified with one solution if $\mathbf{x} \geq 0$ for any objective function.
3. $\mathbf{n} > \mathbf{m}$, the problem us underspecified and if the equations are constistent (don't contradict eachother), there will be infinite many solutions where many will be feasible with $\mathbf{x} \geq 0$.

$$\min_{x_1, x_2, \ldots, x_n} \quad f = c_1x_1 + c_2x_2 + \ldots + c_kx_k + \ldots + c_nx_n$$
$$\text{s.t.}$$
$$\begin{array}{ccccccccccccc}
a_{11}x_1 & + & a_{12}x_2 & + & \ldots & + & a_{1k}x_k & + & \ldots & + & a_{1n}x_n & = & b_1 \\
a_{21}x_1 & + & a_{22}x_2 & + & \ldots & + & a_{2k}x_k & + & & + & a_{2n}x_n & = & b_2 \\
\vdots & & \vdots & & \vdots & & \vdots & & & & \vdots & \vdots & \vdots \\
a_{r1}x_1 & + & a_{r2}x_2 & + & \ldots & + & a_{rk}x_k & + & \ldots & + & a_{rn}x_n & = & b_r \\
\vdots & & \vdots & & & & \vdots & & & & \vdots & \vdots & \vdots \\
a_{m1}x_1 & + & a_{m2}x_2 & + & \ldots & + & a_{mk}x_k & + & \ldots & + & a_{mn}x_n & = & b_m
\end{array}$$
$$x_1 \geq 0, \quad x_2 \geq 0, \ \ldots, \ x_k \geq 0, \ \ldots, \ x_n \geq 0$$

Becuase a linear programs feasible regions is defined only be linear constraints and the function itself has no curvature (being linear), if an optimum exists, it cannot occur in the “interior” of the feasible region as moving in the direction of improvement would always do better until you hit a boundary. Therefore, the optimum must occur at an **extreme point** (a vertex/corner), or along a boundary/edge if the objective is parallel to that constraint, in which case multiple optimal solutions exist.

Tranforming to Standard Form

Any linear program can be rewritten into *standard form*, which requires a minimization of the objective and equality constraints with non-negative variables. Problems not already in this form can be transformed by (i) rewriting the objective as a $-f$ (if originally a maximization of f) and (ii) introducing slack variables to convert inequality constraints into equalities. An inequality conversion is shown below

$$a_1x_1 + a_2x_2 \leq b$$

$$a_1x_1 + a_2x_2 + s = b, \quad s \geq 0$$

where the slack variable x_3 measures how much 'room' is left before the inequality becomes active. Intuitively, a slack variale quantifies the distance between the current solution and the boundary where the inequality constraint is tight.

- **Slack variable = 0** → the inequality is *active* (or "tight"). The solution lies right on the boundary of the constraint, meaning you can't decrease the left-hand side any further without violating feasibility.
- **Slack variable > 0 (large)** → the inequality is *inactive*. There's "room to spare," so the constraint isn't limiting the feasible region at the current point.
- **Slack variable small but positive** → the solution is close to the boundary. The inequality is still satisfied, but only just barely.

It is important to note that standard form represents the important structure

$$\begin{bmatrix} \mathbf{c}^T & \mathbf{0} \\ \mathbf{A} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}_N \\ \mathbf{x}_B \end{bmatrix} = \begin{bmatrix} f \\ \mathbf{b} \end{bmatrix}$$

where the elements of vector $\mathbf{x}_B$ are denoted the ***basic variables*** and elements in vector x_N are the ***nonbasic variables***. This configuration allows for the immediate calculation of a initial feasible solution by setting $\mathbf{x}_N = 0$ resulting in $\mathbf{x}_B = \mathbf{b}$ through $\mathbf{A}\mathbf{x}_\mathbf{N} + \mathbf{I}\mathbf{x}_\mathbf{B} = \mathbf{b}$ and $f = 0$. This result is the origin and a vertex point in the convex feasible region.

Simplex Intuition

Instead of the brute-force approach of evaluating the objective at every possible vertex—which grows combinatorially and becomes infeasible even for moderately sized problems—the simplex method provides an efficient alternative. It relies on the fact that each basic feasible solution corresponds to a vertex of the feasible region, and that neighboring vertices differ only in which basic variables are included ($m - 1$ of those basic variables are still the same). Starting from one feasible vertex, the method identifies a neighboring vertex that yields the greatest improvement in the objective value, then moves there. This process repeats until no further improvement is possible, at which point an optimal solution has been reached (very similar to a greedy search or dikstras algorithm).

Simplex Algorithm

1. Standardize the Problem: Convert the LP into *standard form* where the objective is **maximized** and all inequality constraints are converted into equalities by introducing **slack** (for "≤"), **surplus** (for "≥"), and possibly **artificial variables** (to ensure an initial feasible solution).

2. Find an Initial Basic Feasible Solution (BFS): Identify a vertex of the feasible region which by definitoin is a basic feasible solution. Commonly done by setting the basic variables $\mathbf{x}_N$ equal to zero and slack variables $\mathbf{x}_B$ equal to $\mathbf{b}$.

3. Compute the Entering Variable (Pivot Column): Because neighboring vertices share most of the same basic variables but differ in one, we must determine which nonbasic variable should enter the basis. This is chosen by examining the **reduced costs** in the objective row which, for **maximization**, selects the nonbasic variable with the **most negative reduced cost** (since increasing the objective is the goal).

4. Compute the Leaving Variable (Pivot Row): Perform the **minimum ratio test**: divide the current right-hand side values by the corresponding positive entries in the pivot column. The smallest ratio identifies the basic variable that must leave to preserve feasibility.

5. Pivot Operation: Perform row operations (like Gaussian elimination) so that the entering variable becomes basic and the leaving variable becomes nonbasic. This corresponds to moving along an edge of the feasible polytope to a neighboring vertex.

6. Check for Optimality: If all reduced costs are **positive** (for maximization), the current BFS is optimal. Otherwise, return to Step 3 and continue.

*****Unbounded (Special Case):*** If no leaving variable can be chosen (all ratios are undefined or negative), the problem is unbounded.

***** Degeneracy (Special Case):*** If the minimum ratio test ties, cycling may occur — tie-breaking rules are applied.

*****Multiple optima (Special Case):*** If alternative optima exist, more than one optimal BFS will be found.

Simplex 2-D Example

The following provides a 3-dimensional minimization problem to illustrate the simplex algorithm methodology:

$$\max_{\mathbf{x}} \{z = 3x_1 + 2x_2 + x_3\}$$

s.t.

$$x_1 + x_2 + x_3 \le 4$$

$$2x_1 + x_2 \le 5$$

$$x_! + 3x_2 + 2x_3 \le 7$$

$$x_1, x_2, x_3 \ge 0$$

Introducing the slack variables $s_1, s_2, s_3 \ge 0$ we rewrte the inequalities as equalities

$$x_1 + x_2 + x_3 + s_1 = 4$$

$$2x_1 + x_2 + s_2 = 5$$

$$x_! + 3x_2 + 2x_3 + s_3 = 5$$

Initial Simplex Tableau: Using this standardized form, an initial simplex tableau can be formed with column order $\{x_1 \;\; x_2 \;\; x_3 \;\; s_1 \;\; s_2 \;\; s_3 \;\; | \;\; \text{RHS}\}$ and initial basic variables s_1, s_2, and s_3 with corresponding basic solution $f = 0$.

	x_1	x_2	x_3	s_1	s_2	s_3	RHS
(1)	1	1	1	1	0	0	4
(2)	2	1	0	0	1	0	5
(3)	1	3	2	0	0	1	7
z	-3	-2	-1	0	0	0	0

Pivot 1: The entering variables is choosen as the largest negative in the z row $\rightarrow x_1$ and using the ration test on the x_1 column: $4/1 = 4$, $5/2 = 2.5$, and $7/1 = 7$, row (2) is selected to leave as it is the smallest. The row operation $R_2 \leftarrow R_2/2$ is performed to make the pivot $= 1$ in the x_1 column, row operation $R_1 \leftarrow R_2 - 1$ and $R_3 \leftarrow R_3 - 1R_2$ is performed to zero the x_1 column in the equations above and below row (2), and finally row operation $R_z \leftarrow R_z + 3R_2$ is performed to zero out the x_1 coefficient in the objective row thus x_1 becomes basic (unit in row (2), zeros everywhere else), and s_2 becomes nonbasic.

$$R_2 \leftarrow R_2/2\text{: } [1,\ 0.5,\ 0,\ 0,\ 0.5,\ 0 \mid 2.5]$$
$$R_1 \leftarrow R_1 - R_2\text{: } [0,\ 0.5,\ 1,\ 1,\ -0.5,\ 0 \mid 1.5]$$
$$R_3 \leftarrow R_3 - R_2\text{: } [0,\ 2.5,\ 2,\ 0,\ -0.5,\ 1 \mid 4.5]$$
$$R_z \leftarrow R_z + 3R_2\text{: } [0,\ -0.5,\ -1,\ 0,\ 1.5,\ 0 \mid 7.5]$$

The resulting tableau is now the following with basis: $\{x_1 \;\; s_1 \;\; s_3\}$.

	x_1	x_2	x_3	s_1	s_2	s_3	RHS
(1)	0	0.5	1	1	−0.5	0	1.5
(2)	1	0.5	0	0	0.5	0	2.5
(3)	0	2.5	2	0	−0.5	1	4.5
z	0	−0.5	−1	0	1.5	0	7.5

Pivot 2: The most negative value in the z corresponds to x_3 thus it is the entering variable and from the ratio test on that column: $1.5/1 = 1.5$ and $4.5/2 = 2.25$, thus s_1 leaves the basis. Row operations

Pivot already 1 at R_1.

$R_3 \leftarrow R_3 - 2R_1$:

$[0,\ 2.5-1,\ 2-2,\ 0-2,\ -0.5-(-1),\ 1-0 \mid 4.5-3]$

$\Rightarrow [0,\ 1.5,\ 0,\ -2,\ 0.5,\ 1 \mid 1.5]$

$R_z \leftarrow R_z + R_1$:

$[0,\ -0.5+0.5,\ -1+1,\ 0+1,\ 1.5-0.5,\ 0+0 \mid 7.5+1.5]$

$\Rightarrow [0,\ 0,\ 0,\ 1,\ 1,\ 0 \mid 9]$

results in the following tableau with basis: $\{x_1\ \ x_3\ \ s_3\}$.

	x_1	x_2	x_3	s_1	s_2	s_3	RHS
(1)	0	0.5	1	1	−0.5	0	1.5
(2)	1	0.5	0	0	0.5	0	2.5
(3)	0	1.5	0	−2	0.5	1	1.5
z	0	0	0	1	1	0	9

As there are no negative reduced costs remaining in the objective row, this solution is optimal! Given the current basic variables row 1: $x_3 = 1.5$, row 2: $x_1 = 2.5$, and row 3: $s_3 = 1.5$, all nonbasic variables are set to zero resulting in the final solution

$$x_{opt} = (x_1, x_2, x_3) = (2.5, 0, 1.5) \text{ where } z_{opt} = 9$$

Geometrically, the constraints define a convex polyhedron in $\mathbf{R}_3$ with the vertex $(2.5.0, 1.5)$ where the objective is maximized. Looking at the final slack values we note that if $s_i = 0 \rightarrow$ the constraint i is said to be binding (active) and geometrically the solutoin lies on the corresponding face/hyperplane. If instead $s_i > 0 \rightarrow$ the constraint i is inactive meaning we are inside the feasible region away from that face be exactly s_i units. A negative slack would mean that the solutoin is infeasible. For this example

$$x_1 + x_2 + x_3 + s_1 = 4 \rightarrow s_1 = 0 \rightarrow \text{Binding Face}$$

$$2x_2 + x_3 + s_2 = 5 \rightarrow s_2 = 0 \rightarrow \text{Binding Face}$$

$$x_1 + 3x_2 + x2_3 + s_3 = 4 \rightarrow s_3 = 1.5 \rightarrow \text{Inactive Face}$$

Matlab LP Solver

The LP solver avaliable in MATLAB solves the general problem defined in the standard (or nonstandard) form above

```
[xopt,fopt,flag,output,lambda]
              =linprog(c,A,b,Aeq,beq,LB,UB)
```

attempts to solve the linear programming problem given by:

```
min c'*x subject to: A*x <= b,Aeq*x=beq&LB<x< UB
```

Set `A=[]` and `B=[]` if no inequalities exist. Use empty matrices for `LB` and `UB` if no bounds exist. `xopt` is the optimal solution and `fopt` the optimum value of the objective function. When `flag=1` the global solution has been found. `Lambda` is s structure with the Lagrange multipliers.