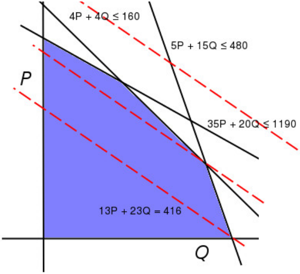
Linear Programming

Dr Timothy Kimber

March 2015



Motivating Problem

Example (Farming)

A farmer went walking with his cow and managed to swap it for 160 magic beans. The beans are magic because they can be used to grow either peas or quinao. If he grows peas he can sell them for 13 groats per bushell. If he grows quinao he can sell it for 23 groats per bushell. Each crop will produce one bushell from just 4 beans. He has 1190 kg of fertiliser and 480 gallons of pesticide. Growing a bushell of peas will use 35 kg of fertiliser and 5 gallons of pesticide. Growing a bushell of quinoa will use 20 kg of fertiliser and 15 gallons of pesticide. What is the maximum revenue the farmer can make?

- 34 bushells of peas uses: 1190 fert., 170 pest. and 136 beans
- 32 bushells of quinoa uses: 640 fert., 480 pest. and 128 beans

A Linear Programming Problem

This is another kind of optimisation problem

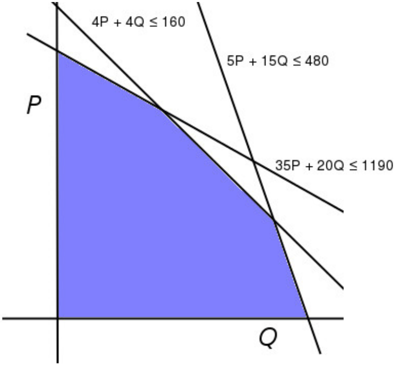
- Let bushells of peas $= P$, bushells of quinao $= Q$
- We have a series of equations, inequalities and constraints in P and Q

$$\begin{aligned} 13P + 23Q & \qquad \textit{revenue} && (1) \\ 4P + 4Q \leq 160 & \qquad \textit{beans} && (2) \\ 35P + 20Q \leq 1190 & \qquad \textit{fertiliser} && (3) \\ 5P + 15Q \leq 480 & \qquad \textit{pesticide} && (4) \\ P \geq 0, Q \geq 0 & \qquad \text{(no negative amounts)} && (5) \end{aligned}$$

- We want to maximise revenue subject to constraints 2–5

Geometric Interpretation

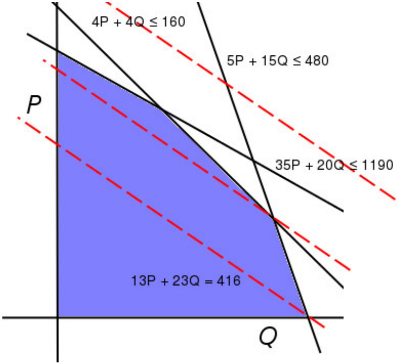
In two dimensions the problem has a geometric interpretation



- Each constraint defines a half-plane
- Values of P and Q in the shaded area satisfy all the constraints

Geometric Interpretation

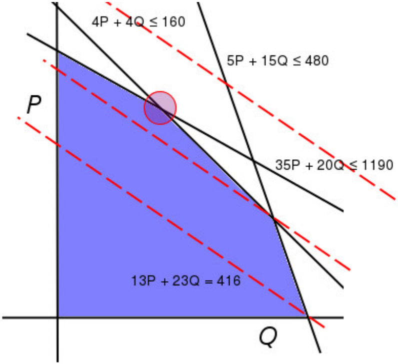
We want to maximise the value of the revenue function



- The feasible region (shaded) is convex polygon
- So, the maximum will occur at a vertex

Geometric Interpretation

We want to maximise the value of the revenue function



- The feasible region (shaded) is convex polygon
- So, the maximum will occur at a vertex

In General

As the number of dimensions (variables) scales up there is good news:

- The geometric intuition still holds
- Only extreme points (vertices) need to be considered

and bad news:

- The number of possible solutions increases exponentially

and more good news:

- There is an algorithm that usually runs in polynomial time
- There are other algorithms that always run in polynomial time

Definitions

Definition (Linear Function)

Given a set of variables $x_1, x_2, \ldots, x_n$, a linear function on these variables is a function f of the form

$$f(x_1, x_2, \ldots, x_n) = a_1x_1 + a_2x_2 + \cdots + a_nx_n$$

where $a_1, a_2, \ldots, a_n$ is a set of real numbers.

Definition (Linear Equality)

Given a linear function f and a real number b, a linear equality is an equation of the form

$$f(x_1, x_2, \ldots, x_n) = b$$

Definitions

Definition (Linear Inequality)

Given a linear function f and a real number b, a linear inequality is an inequality of the form

$$f(x_1, x_2, \ldots, x_n) \leq b, \text{ or}$$
$$f(x_1, x_2, \ldots, x_n) \geq b$$

- A linear constraint is either a linear equality or a linear inequality.

Definition (Linear Programming Problem)

A linear programming problem is the problem of minimising or maximising a linear function subject to a set of linear constraints.

- A feasible solution is an assignment $\bar{x}$ of values to $x_1, \ldots x_n$ that satisfies all the constraints
- A solution $\bar{x}$ has an objective value $f(\bar{x})$

Standard Form

Before we can apply algorithms we need a standard form

Standard Form

$$
\begin{aligned}
\text{Maximise:} \quad & c_1x_1 + \cdots + c_nx_n \\
\text{Subject to:} \quad & a_{11}x_1 + \cdots + a_{1n}x_n \leq b_1 \\
& \ldots \\
& a_{m1}x_1 + \cdots + a_{mn}x_n \leq b_m \\
& x_1 \geq 0, \ldots, x_n \geq 0
\end{aligned}
$$

- The objective function must be maximised
- All variables must be constrained to be non-negative
- All other constraints must be less-than-or-equal constraints

Converting to Standard Form

Any linear programming problem can be transformed into an equivalent problem in standard form.

Definition (Equivalent Linear Programs)

A linear program L is is equivalent to a linear program L' if and only if

- for every feasible solution to L with objective value z there is a feasible solution to L' with objective value z, and
- for every feasible solution to L' with objective value z there is a feasible solution to L with objective value z.

- Multiply minimised functions by -1
- Multiply $\geq$ constraints by -1
- Replace equalities with two inequalities
- Replace a variable x that has no ≥ 0 constraint by $x' - x''$

Converting to Standard Form

Example

$$\begin{aligned} \text{Minimise:} \quad & 3x_1 - 4x_2 \\ \text{Subject to:} \quad & x_1 + x_2 = 20 \\ & 2x_1 + 5x_2 \le 55 \\ & x_1 \ge 0, \ldots \end{aligned}$$

Becomes

$$\begin{aligned} \text{Maximise:} \quad & -3x_1 + 4x_2 - 4x_3 \\ \text{Subject to:} \quad & -x_1 - x_2 + x_3 \le -20 \\ & x_1 + x_2 - x_3 \le 20 \\ & 2x_1 + 5x_2 - 5x_3 \le 55 \\ & x_1 \ge 0, x_2 \ge 0, x_3 \ge 0 \end{aligned}$$

Slack Form

To apply the Simplex Algorithm we need to convert the LP to Slack Form

- A slack variable is added to transform inequalities to equalities
- Non-negativity constraints are not changed
- New non-negativity constraints are added for the slack variables

So, $x_1 + x_2 - x_3 \leq 20$ becomes $x_1 + x_2 - x_3 + x_4 = 20$

Example (Slack Form)

$$
\begin{array}{rcrcrcrcr}
z & = & & & -3x_1 & + & 4x_2 & - & 4x_3 \\
x_4 & = & -20 & + & x_1 & + & x_2 & - & x_3 \\
x_5 & = & 20 & - & x_1 & - & x_2 & + & x_3 \\
x_6 & = & 55 & - & 2x_1 & - & 5x_2 & + & 5x_3
\end{array}
$$

$$x_1, x_2, x_3, x_4, x_5, x_6 \geq 0$$

The Simplex Algorithm

The Simplex Algorithm (George Dantzig, 1947) moves from vertex to vertex of the simplex (the feasible region).

Example

$$
\begin{array}{rcrcrcr}
z & = & & & 13P & + & 23Q \\
s_1 & = & 160 & - & 4P & - & 4Q \\
s_2 & = & 1190 & - & 35P & - & 20Q \\
s_3 & = & 480 & - & 5P & - & 15Q \\
\end{array}
$$
$$P, Q, s_1, s_2, s_3 \geq 0$$

The variables on the LHS are called basic variables

- The basic variables are initially the slack variables
- Basic and non-basic variables will be exchanged later
- Generate a basic solution by setting all non-basic variables to 0
- If a basic solution is feasible it is a basic feasible solution

The Simplex Algorithm

Example

$$
\begin{array}{rcrcrcr}
z & = & & & 13P & + & 23Q \\
s_1 & = & 160 & - & 4P & - & 4Q \\
s_2 & = & 1190 & - & 35P & - & 20Q \\
s_3 & = & 480 & - & 5P & - & 15Q \\
\end{array}
$$

$$P, Q, s_1, s_2, s_3 \geq 0$$

The first basic solution is $\{s_1 = 160, s_2 = 1190, s_3 = 480, P = 0, Q = 0\}$

- This corresponds to the vertex at the origin
- P and Q have their lowest feasible values
- The slack variables all have their largest values
- Next increase P or Q without violating a constraint

Pivoting

The algorithm proceeds by choosing a non-basic variable to increase

- This should be a variable in the objective function
- It must have a positive coefficient
- In this case either will do, so we will pick Q

Example

$$\begin{array}{rcrcrcr} z & = & & & 13P & + & 23Q \\ s_1 & = & 160 & - & 4P & - & 4Q \\ s_2 & = & 1190 & - & 35P & - & 20Q \\ s_3 & = & 480 & - & 5P & - & 15Q \end{array}$$
$$P, Q, s_1, s_2, s_3 \geq 0$$

Pivoting

Next, a basic variable is chosen to be decreased

- The choice depends on Q
- Choose the variable that imposes the tightest limit on Q
- In the example, if $Q > 32$ then $s_3 < 0$
- This is the greatest value Q can take without breaking a constraint
- So, s_3 will be selected

Example

$$
\begin{array}{rcrcrcr}
z & = & & & 13P & + & 23Q \\
s_1 & = & 160 & - & 4P & - & 4Q \\
s_2 & = & 1190 & - & 35P & - & 20Q \\
s_3 & = & 480 & - & 5P & - & 15Q \\
\multicolumn{7}{c}{P, Q, s_1, s_2, s_3 \geq 0}
\end{array}
$$

Pivoting

Now a pivot operation is performed

- Rewrite the selected equation with Q on the left: $Q = \frac{480}{15} - \frac{P}{3} - \frac{s_3}{15}$
- Then substitute this expression into the other equations
- The result is to make Q basic and s_3 non-basic

Example

$$\begin{array}{rcrcrcr} z & = & & & 13P & + & 23Q \\ s_1 & = & 160 & - & 4P & - & 4Q \\ s_2 & = & 1190 & - & 35P & - & 20Q \\ s_3 & = & 480 & - & 5P & - & 15Q \\ & & & P, Q, s_1, s_2, s_3 \geq 0 & & & \end{array}$$

Pivoting

Now a pivot operation is performed

- Rewrite the selected equation with Q on the left: $Q = \frac{480}{15} - \frac{P}{3} - \frac{s_3}{15}$
- Then substitute this expression into the other equations
- The result is to make Q basic and s_3 non-basic

Example

$$\begin{array}{rcrcrcr} z & = & 736 & + & \frac{16P}{3} & - & \frac{23s_3}{15} \\ s_1 & = & 32 & - & \frac{8P}{3} & + & \frac{4s_3}{15} \\ s_2 & = & 550 & - & \frac{85P}{3} & + & \frac{4s_3}{3} \\ Q & = & 32 & - & \frac{P}{3} & - & \frac{s_3}{15} \\ \end{array}$$
$$P, Q, s_1, s_2, s_3 \geq 0$$

- The second basic solution is $\{s_1 = 32, s_2 = 550, Q = 32, P = 0, s_3 = 0\}$
- The objective value is now 736

Pivoting

A second pivot using P and s_1 gives

- $P = 12 - \frac{3s_1}{8} - \frac{s_3}{10}$

And

Example

$$\begin{array}{rcrcccc} z & = & 800 & - & 2s_1 & - & s_3 \\ P & = & 12 & - & \frac{3s_1}{8} & - & \frac{s_3}{10} \\ \\ s_2 & = & 210 & + & \frac{85s_1}{8} & - & \frac{3s_3}{2} \\ Q & = & 28 & + & \frac{s_1}{8} & - & \frac{s_3}{10} \\ & & & P, Q, s_1, s_2, s_3 \geq 0 & & & \end{array}$$

- The third basic solution is $\{P = 12, Q = 28, s_2 = 210, s_1 = 0, s_3 = 0\}$

Optimal Solution

There are no variables with positive coefficients left in the objective function

- The objective value is now 800 and cannot be increased further
- The algorithm terminates here

Example

$$\begin{array}{rcrcrcr} z & = & 800 & - & 2s_1 & - & s_3 \\ P & = & 12 & - & \frac{3s_1}{8} & - & \frac{s_3}{10} \\ \\ s_2 & = & 210 & + & \frac{85s_1}{8} & - & \frac{3s_3}{2} \\ Q & = & 28 & + & \frac{s_1}{8} & - & \frac{s_3}{10} \end{array}$$

$$P, Q, s_1, s_2, s_3 \geq 0$$

Other Considerations

Initialisation of the Simplex algorithm is not always this simple

- We selected P and Q to be non-basic initially
- With different constraints, this could have been an infeasible solution
- If the problem is feasible, some initial solution must be found
- An initialisation algorithm
  - determines if the problem is feasible, and if so
  - returns a basic feasible solution

The Simplex algorithm might also cycle

- The objective value never decreases but can remain constant
- So, states can be repeated
- With care, this can be avoided
- e.g. follow Bland's rule and always select lowest indexed non-basic var

Other Considerations

The problem might be unbounded

- The constraints may not limit the objective function
- Can be detected by the Simplex algorithm

Example

If x_1 has been selected to change from non-basic to basic (the entering variable), and the constraints are

$$\begin{array}{rcrcrcr} x_2 & = & 40 & + & 2x_1 & - & 5x_5 \\ x_3 & = & 556 & + & \frac{20}{9}x_1 & + & \frac{x_5}{4} \\ x_4 & = & 19 & + & \frac{19}{9}x_1 & - & \frac{28}{11}x_5 \end{array}$$

then all constraints limit x_1 to ∞ and the objective value is unbounded.

Other Considerations

And

- Floating point calculations are fraught with problems

Recommended : use an existing linear programming package

- OpenOpt
- glpk
- LpSolve
- Matlab
- etc.

Using the ellipsoid or interior points algorithms these packages are guaranteed to solve linear programs efficiently.

Other Problems as Linear Programs

Many problems can be expressed, and efficiently solved, as linear programs

Problem *(Shortest Paths)*

Input: digraph $G = (V, E)$ with weight function w

Input: source vertex s

Input: destination vertex t

Output: shortest path from s to t in G

Recall Bellman–Ford

- Have a variable $dist_v$ for every vertex in G
- Each edge (u, v) supplies a constraint: $dist_v \leq dist_u + w(u, v)$
- For the source vertex: $dist_s = 0$

Shortest Paths Linear Program

As a linear program, the problem is

Shortest Path Linear Program

$$
\begin{aligned}
\text{Maximise:} \quad & dist_t \\
\text{Subject to:} \quad & dist_v \leq dist_u + w(u,v), \text{for each } (u,v) \in E \\
& dist_s = 0
\end{aligned}
$$

- The objective function has to be maximised (or the optimal value is 0)
- We know for each v: $dist_v = dist_u + w(u,v)$ for some u
- So, each $dist_v$ is the maximum value that satisfies the constraints
- Therefore, $dist_t$ is maximised

Applications of Linear Programming

It is often easy to express an optimisation problem as a linear program

- If so it is easy and efficient to solve

For a given problem

- There might be a specialised algorithm (e.g. Bellman–Ford)
- This could be faster
- But there might not be one
- Or LP solution might be good enough

This makes linear programming an important tool

- A general problem solving model
- Widely applied in industry
- A big topic that we have briefly dipped into