Linear programming and the Simplex algorithm

by

Jens Hee
https://jenshee.dk

January 2018

Change log

20. January 2018

1. Document started.

29. January 2018

1. Quadratic programming added.

13. March 2020

1. Meta data added.

Contents

1 Linear programming **1**

1.1 Introduction . . . 1

1.2 Geometric view . . . 1

1.3 Arithmetic view . . . 2

1.3.1 Unconstrained variables . . . 3

1.4 Solving the problem . . . 3

1.4.1 Phase I . . . 3

1.4.2 Phase II . . . 4

A Quadratic programming **5**

Chapter 1

Linear programming

1.1 Introduction

Optimizations are used in a large number of situations. Examples are least square fit, mini-max approximation, the travelling salesman problem and the knapsack problem. This document concentrates on a class of problems called linear programming where a linear function is maximized or minimized subject to linear constraints. The related problem quadratic programming is briefly covered in Appendix A. An linear programming example is:

Maximize:

$$x + 2y$$

subject to the constraints:

$$\begin{array}{rcrcr} & & y & \leq & 8 \\ x & + & y & \leq & 12 \\ 2x & + & y & \leq & 20 \end{array}$$

1.2 Geometric view

The above constraints are shown in Figure 1.1 together with the object function (the function to be maximized).
Assuming $x \geq 0$ and $y \geq 0$, it is seen that the constraints form a polygon. It is also clear that the maximal solution is found by translating the object function as much as possible to the right.
If the problem is ill-formed the polygon is unbound and there is no final solution. If for example the assumption x and y being positive are removed and a minimum solution is desired, the object function can be moved infinitely to the left.
A linear Programming problem may have no solutions at all, if the constraints contradicts. An example is shown i Figure 1.2.
In general, where more variables are used, the constraints form a hyper polyhedron and the object function is a hyper plane. The solution can be found by translating the hyper plane until the optimal solution is found.
Note that the solution is at one of the corners of the hyper polyhedron. The Simplex algorithm searches through the corners in order to find a solution.

The algorithm is divided into two phases: Phase I finds a corner and Phase II then searches from one corner to the next, increasing the object function at each iteration.

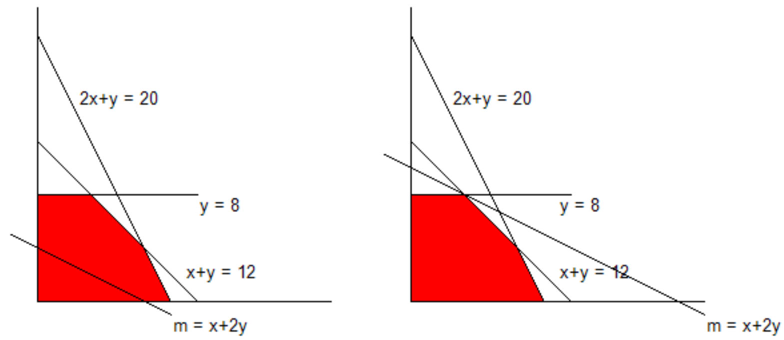

Figure 1.1: Simple example

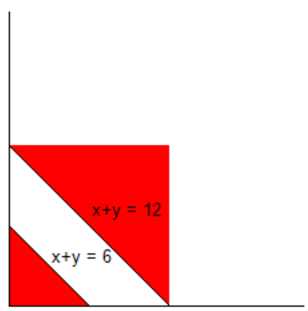

Figure 1.2: Example having no solution

1.3 Arithmetic view

In general each constraint can be an equality (=) or an inequality ($\leq$ *or* $\geq$). The $\geq$ can be changed to $\leq$ by changing the sign on each side of the inequality. By adding a positive so called slag variable on the left side of the inequality it can be changed to an equality. This means that the constraints in any linear programming problem can be written on the form:

$$\sum_j a_{ij}x_j = b_i$$

where the variables x_j may be constrained as follows:

$$\begin{array}{rcccl} a & \leq & x_j & < & \infty \\ -\infty & < & x_j & \leq & b \\ a & \leq & x_j & \leq & b \\ -\infty & < & x_j & < & \infty \end{array}$$

The upper limit of the third variable constraint can be removed by adding it to the linear programming problem. Now the variable constraints, by a shift and/or negation of the variable, can be reduced to:

$$0 \leq x_j < \infty$$

or x_j unconstrained.

1.3.1 Unconstrained variables

If an unconstrained variable x_j is replaced by the difference between two positive variables $x_j = z1_j - z2_j$ all variables will be constrained by:

$$0 \leq x_j < \infty$$

However, this approach may lead to numeric problems, when solving the linear programming problem. Since the unconstrained variables do not influence the solution a better approach is to eliminate these variables from the linear programming problem, solve the reduced problem and finally calculate the values of the unconstrained variables.

1.4 Solving the problem

As can be seen from the above, any linear programming problem can be written as:

Maximize or minimize:

$$\sum_j c_j x_j$$

subject to the constraints:

$$\sum_j a_{ij} x_j = b_i, \qquad b_i \geq 0, \qquad x_j \geq 0$$

It can be proven that among the solutions only the basic feasible solutions are necessary to consider. If the A matrix above has n columns and m rows a basic solution is a solution where $m - n$ variables are set to zero and the resultant variables are found by solving the resulting m by m matrix. The solution is found by a sequence of row operations, but only solutions where $x_j \geq 0$ are feasible. Once a basic feasible solution is known, all other basic feasible solutions can be found by appropriate row operations.

1.4.1 Phase I

In Phase I a basic feasible solution is found if possible. This is done by adding a set of so called artificial variables to the set of equations. These variables must be zero in order not to alter the problem. This is achieved by first requiring them to be positive, then minimizing their sum, keeping the other variables ≥ 0 using row operations. If the minimum is zero, then all the artificial variables are zero and a basic feasible solution has been found, otherwise the original problem has no solution.

1.4.2 Phase II

If a basic feasible solution is found in Phase I, then Phase II is used used to find the optimal solution using row operations. In the following it is assumed a minimum solution is desired. At each iteration a pivot element is chosen so that the coefficient of the object function in the pivot column k is negative. Often the leftmost negative is chosen for simplicity. The pivot element must be positive for the solution to be positive, since the right hand side is positive. The pivot row is the row where the pivot element has the least quotient b_i/a_{ik}, in order to keep the right hand side positive after the row operations.
At each iteration a new basic feasible solution is found and the object function is decreased. When no more negative coefficients are found in the object function, the minimum solution has been found.
If a pivot column is encountered where all elements are negative including the object function coefficient, the solution is at negative infinity.

Appendix A

Quadratic programming

If a quadratic term is added to the object function the problem is called quadratic programming:

Minimize:

$$\sum_j c_j x_j + \frac{1}{2} \sum_j xj \sum_k Q_{jk} x_k, \qquad Q_{ij} = Q_{ji}$$

subject to the constraints:

$$\sum_j a_{ij} x_j \leq b_i, \qquad x_j \geq 0$$

Using the Karush-Kuhn-Tucker conditions the problem can be formulated:

$$\sum_j Q_{ij} x_j + \sum_j a_{ji} \mu_j - y_i = -c_i$$

$$\sum_j a_{ij} x_j + \nu_i = b_i$$

$$x_j \geq 0$$

$$\mu_j \geq 0$$

$$y_j \geq 0$$

$$\nu_j \geq 0$$

$$\sum_j y_j x_j = 0$$

$$\sum_j \mu_j \nu_j = 0$$

y_j are called the surplus variables and ν are the slag variables. The last two equations requires y_j and x_j not being in the basis at the same iteration and correspondingly for μ and ν. If any $b_j < 0$, the equation is multiplied by -1.

The constraints are now on the same form as in section 1.4 and the problem is solved as Phase I in section 1.4.1.

Bibliography

[1] Katta G. Murty, *Linear Programming*, 1983, John Wiley & Sons.

[2] Paul A. Jensen and Jonathan F. Bard, *Operation Research Models and Methods, Quadratic Programming*, John Wiley & Sons.