

Section B6: Mathematical Programming

B6.1. Introduction

Mathematical programming seeks to optimize (maximize or minimize) an *objective function* subject to *constraints*. There are many types of mathematical programming, including the following:

- Linear programming (LP): optimizes a linear function subject to linear constraints.
- Quadratic programming (QP): optimizes a quadratic function subject to linear constraints.
- Mixed integer programming (MIP): optimizes a linear function subject to linear constraints, but with some or all variables restricted to integer values

There are many other types of mathematical programming, not discussed here, including nonlinear programming, combinatorial optimization, and dynamic programming.

Figure 1 show an example “toy” linear program, along with basic terminology used to describe a mathematical program (from Paris Q (2016) *An Economic Interpretation of Linear Programming*).

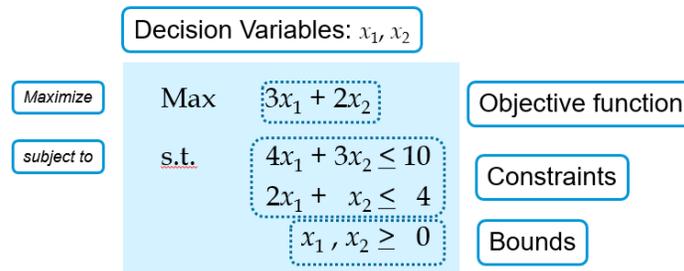


Figure 1. Example linear program

This toy example has only two decision variables (x_1 and x_2) and two constraints. LPs and QPs used in NEMS and WEPS can have many thousands of decision variables and constraints.

B6.2. Solution Approach

Mathematical programs are usually solved iteratively, starting with a set of non-optimal decision variable values. Each iteration involves adjusting the values of the decision variables and evaluating the objective function value. The iterative procedure, shown in Figure 2, stops when it is no longer possible to improve (increase or decrease) the value of the objective function.

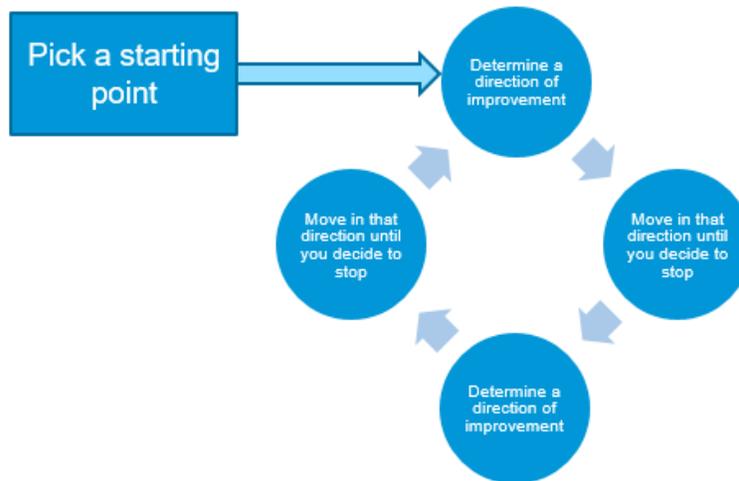


Figure 2. Iterative solution approach

Solution approaches for LPs can be broadly classified as *extreme point* methods and *interior point* methods. The two approaches can be described graphically for a small LP that comprises only two decision variables.

Figure 3 is a graphical representation of the the LP described in Figure 1. Each decision variable is represented by an axis; the constraints and bounds define a *feasible* region with a blue *interior*. The objective function is shown with an arrow representing the direction of improvement. Each vertex formed by an intersection of constraints (or bounds) is called an an *extreme point* and is shown by a red dot in the figure.

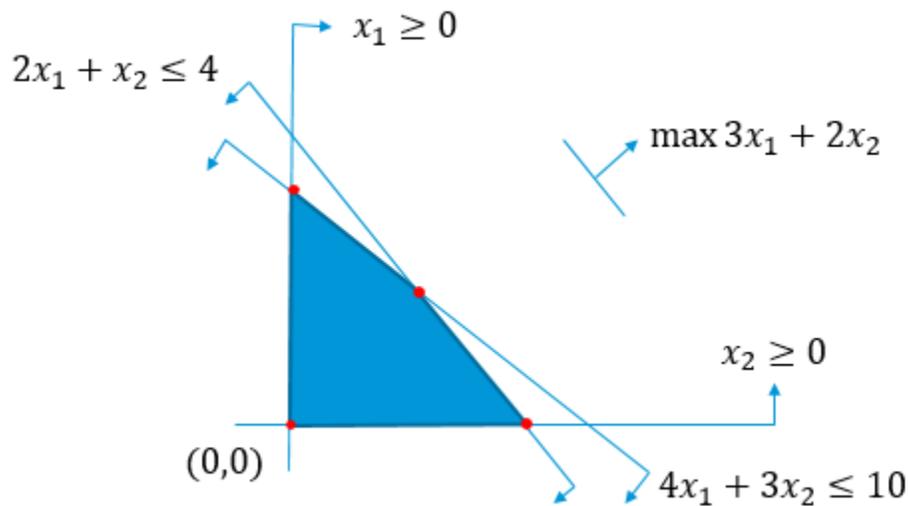


Figure 3. Graphical representation of example LP

An extreme point method, such as the *simplex method* (see the references below), uses an extreme point as the starting point. Each step of the solution algorithm requires deciding which

adjacent extreme point to visit next, then moving to that point. The algorithm ends when there is no further improvement possible.

An interior point method, such as the barrier method, starts from any feasible point, and proceeds through the interior of the feasible region until no further improvement is possible.

B6.3. Interpreting an Optimal Solution

An optimal solution is characterized by

- Objective function value
- Decision variable values
- Shadow prices (for constraints)
- Slacks (for constraints)
- Reduced costs (for decision variables)

The objective function value and decision variable values are the values of the objective function and the decision variables, respectively, when the objective function is optimized subject to the constraints.

The *shadow price*, or dual price, of a constraint is the partial derivative of the the objective function with respect to the right-hand side of the constraint, evaluated at the point specified by the optimal solution. In other words, a constraint's shadow price tells how much the value of the objective function would change if the the scalar portion of the constraint were changed by a *small* amount.

For some carefully constructed LPs and QPs, the shadow price can be interpreted as the *price* of a resource or product. For example, the shadow price of a certain constraint might represent the wholesale price of motor gasoline in the U.S. Gulf Coast region. In general, the shadow price indicates the extent to which the constraint affects the optimal value of the objective function; every constraint potentially limits the optimal value, raising the minimum or lowering the maximum.

The *slack* of a constraint is the amount by which the constraint is non-binding. For example, if $x + y$ is restricted to be less than or equal to 10, and $x + y$ is actually 8 in the optimal solution, then this constraint has a slack value of $10 - 8 = 2$. A constraint with a non-zero slack value has a shadow price of zero.

The *reduced cost* of a decision variable tells how much the variable's objective function coefficient would have to change before the variable would take a non-zero value in the optimal solution. Variables with a zero reduced cost have non-zero values in the optimal solution.

References

More information about mathematical programming is readily available on the internet, and from undergraduate-level textbooks such as the following:

- Carter MW, Price CC (2001) *Operations Research: A Practical Introduction* (CRC Press)
- Eiselt HA, Sandblom CL (2012) *Operations Research: A Model-Based Approach* (Springer)
- Hillier FS, Lieberman GJ (2015) *Introduction to Operations Research* (10th Edition) (McGraw Hill)
- Mariappan P (2013) *Operations Research: An Introduction* (Pearson)
- Rardin RL (2017) *Optimization in Operations Research* (2nd Edition) (Pearson)
- Taha, HA (2017) *Operations Research: An Introduction* (10th Edition) (Pearson)
- Winston, WL (2004) *Operations Research: Applications and Algorithms* (4th Edition) (Thomson Brooks/Cole)

The following text is more advanced but especially relevant to EIA LP models:

- Paris Q (2016) *An Economic Interpretation of Linear Programming* (Revised and Updated Edition) (Palgrave/Macmillan)