What Is Mathematical Modeling and Operations Research

Thomas R. Cameron

January 12, 2026

1 What Is Mathematical Modeling?

Mathematical modeling is the process of translating a problem into a mathematical formulation that can be be used to provide analysis, a solution, or an approximation. A model is often a simplification of reality since it retains only those features that are relevent or can be solved.

1.1 An Example from Calculus

Here is a familiar problem from calculus: An open box is to be constructed from a $16'' \times 30''$ piece of cardboard by cutting out squares of equal size from the four corners and bending up the sides. What size should the squares be to obtain a box with the largest volume. Below is a mathematical model that represents this problem, where x denotes the length and width of the square that is being cut out from each corner.

$$\text{maximize} \quad x(16-2x)(30-2x) \tag{1a}$$

$$\text{subject to} \quad 0 \leq x \leq 8 \tag{1b}$$

The model in (1a)–(1b) has an objective function of $x(16-2x)(30-2x)$ along with two constraints that bound the values of the independent variable x. We can determine an optimal value to this model using standard calculus techniques.

1.2 An Example from Linear Algebra

In linear algebra, systems of equations often arise from measurements or data fitting. Suppose a model predicts that an output y depends linearly on an input x, so that $y \approx ax+b$. Given measured data (x_i, y_i) for $i = 1, \ldots, n$, one seeks parameters a and b that best explain the data. This leads to an overdetermined linear system

$$\begin{pmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} \approx \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}.$$

A least squares solution is obtained by minimizing the squared error

$$\min_{a,b} \sum_{i=1}^{n} (ax_i + b - y_i)^2,$$

2 Operations Research and the Berlin Airlift

Operations Research (OR) is concerned with the application of mathematical models to decision-making problems involving limited resources. The field emerged prominently during World War II and its aftermath.

2.1 Historical Context: The Berlin Airlift

The Berlin Airlift of 1948–1949 involved supplying West Berlin by air after land routes were blocked. Aircraft capacity, runway availability, fuel supply, and crew scheduling all imposed constraints.

Figure 1: Map (left) illustrating the air corridors used during the Berlin Airlift. Cargo aircraft (right) used during Berlin Airlift. Source: Wikimedia Commons.

2.2 A Simplified Model

Consider a simplified version of the airlift, see (2a)–(2e). Suppose two types of aircraft are available. Let x and y denote the number of daily flights of each type, the cargo capacity of each is 3000 and 2000, respectively. No more than 44 planes could be used in a given time period; x requires 14 personnel and y requires 10 personnel, the total number of personnel cannot exceed 503; flight x costs 8000 and flight y costs 5000. the total cost cannot exceed 275000.

$$\text{maximize} \quad 30000x + 20000y \tag{2a}$$

$$\text{subject to} \quad x + y \leq 44, \tag{2b}$$

$$14x + 10y \leq 503, \tag{2c}$$

$$8000x + 5000y \leq 275000, \tag{2d}$$

$$x, y \geq 0. \tag{2e}$$

Figure 2 illustrates the feasible region of the simplifed Berlin Airlift model. As we will see, when an optimal solution to a linear program exists, it occurs at a vertex of this region. In this figure, we display each vertex and in red we display the optimal solution.

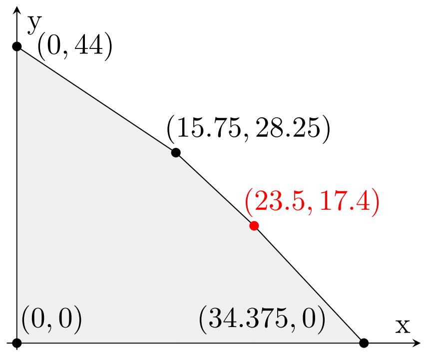


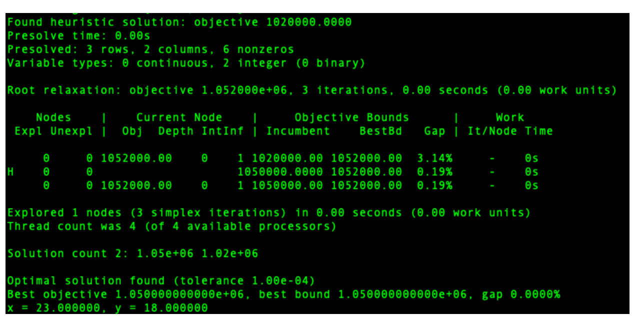
Figure 2: Feasible region for simplified Berlin Airlift model.

2.3 Solving the Model in Gurobi

Obviously we cannot have a fractional number of planes; hence, we desire an integral solution. This integral solution occurs in the feasible region but may not be one of the vertices. We will discuss how to find optimal integral solutions later in the course. Below we show how to solve this model in Gurobi. Figure 3 shows a Python snippet that uses Gurobi to solve the simplified Berlin Airlift model. Figure 4 shows a terminal snippet that outputs the Gurobi solution to the simplified Berlin Airlift model.

```python
from gurobipy import Model, GRB
model = Model("berlin_airlift")
x = model.addVar(vtype=GRB.INTEGER)
y = model.addVar(vtype=GRB.INTEGER)
model.setObjective(30000*x+20000*y,GRB.MAXIMIZE)
model.addConstr(x+y<=44)
model.addConstr(14*x+10*y<=503)
model.addConstr(8000*x+5000*y<=275000)
model.optimize()
```

Figure 3: Python snippet to solve simplified Berlin Airlift model in Gurobi.



```
Found heuristic solution: objective 1020000.0000
Presolve time: 0.00s
Presolved: 3 rows, 2 columns, 6 nonzeros
Variable types: 0 continuous, 2 integer (0 binary)

Root relaxation: objective 1.052000e+06, 3 iterations, 0.00 seconds (0.00 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0 1052000.00    0    1 1020000.00 1052000.00  3.14%     -    0s
H    0     0                    1050000.0000 1052000.00  0.19%     -    0s
     0     0 1052000.00    0    1 1050000.00 1052000.00  0.19%     -    0s

Explored 1 nodes (3 simplex iterations) in 0.00 seconds (0.00 work units)
Thread count was 4 (of 4 available processors)

Solution count 2: 1.05e+06 1.02e+06

Optimal solution found (tolerance 1.00e-04)
Best objective 1.050000000000e+06, best bound 1.050000000000e+06, gap 0.0000%
x = 23.000000, y = 18.000000
```

Figure 4: Terminal snippet showing Gurobi output when solving simplified Berlin Airlift model.