TicTech

Aster Santana, July 2020

This is the documentation of the TicTech use case, which is part of the Learning Mip project maintained by Mip Master.

Concepts

- The three steps to solve a problem
- The three components of a formulation
- Modeling with binary decision variables
- Calling a MIP solver

Problem statement

Mr. Mip has been contacted by *TicTech*, an organization that is now going through a digital transformation and needs to make an important decision. They need to decide on the right type of technology that will support them making complex decisions as their business grow.

After reading a blog about technologies for decision-making, Mr. Mip arrived at three options:

1) Consulting services
2) Off-the-shelf software
3) Purpose-built web applications (Apps)

By analyzing pos and cons of each technology in the context of TicTech, Mr. Mip derived in collaboration with them a score to each technology as follows: 12-Consulting, 17-Software, 25-Apps. What technology should be selected?

Assuming that Mr. Mip he can only recommend one technology and the goal is to maximize the score, it's obvious that TicTech should pick apps. However, Mr. Mip is a MIP fanatic! And sometimes he uses MIP just for fun. And sometimes just for teaching others how to use this fantastic technology! So, let's see how he solved this baby problem using MIP.

Mr. Mip always breaks the problem in three major steps:

1) Understanding the business problem: Identifying all the requirements of the problem, including the data.
2) Mathematical formulation: Writing a precise mathematical representation of the problem.
3) Model implementation and optimization: Coding the formulation as an optimization model and using an optimization solver to compute the best solution.

Formulation

For Mr. Mip, formulation is nothing but a very precise representation of the business problem in mathematical terms.

Typically, his formulations have three main components:

i. Decision variables
ii. Constraints

iii. Objective function

Decision variables

While there might be multiples ways to define the decision variables, choosing a good set of variables is crucial. Because, once the variables are defined, constraints and the objective are defined as functions of it.

For this problem, Mr. Mip defined three binary variables.

Decision variables:

x_1 equals 1 if consulting is chosen, 0 otherwise

x_2 equals 1 if off-the-shelf software is chosen, 0 otherwise

x_3 equals 1 if purpose-built apps are chosen, 0 otherwise

Constraints

In this use case, there is only one requirement: one, and only one, technology must be chosen.

Mr. Mip formulated this requirement using a single constraint.

Constraints – Exactly one technology:

$$x_1 + x_2 + x_3 = 1.$$

Take a moment to think why this equation does the work.

Can you think of an alternative way? There is a hint[1] in the footnote if you need.

Objective

The objective of this problem is to maximize the score, which Mr. Mip formulated as following.

Objective:

$$\max 12\, x_1 + 17\, x_2 + 25\, x_3.$$

Take another moment to think why this objective function does the work.

Final formulation

Putting all together, Mr. Mip arrived at the following formulation.

Final formulation:

$$\begin{array}{ll} \max & 12\, x_1 + 17\, x_2 + 25\, x_3 \\ s.t. & x_1 + x_2 + x_3 = 1 \\ & x_1, x_2, x_3 \in \{0,1\}. \end{array}$$

Implementation and optimization

For Mr. Mip, implementation is the translation of the mathematical formulation into a language that computers understand. While there are many programming and modeling languages out there for that

[1] How about using three constraints, involving two variables at a time?

end, one of Mr. Mip's personal favorite is *gurobipy*, the Python interface of the Gurobi solver. What he likes the most in gurobipy, apart from being in Python, is that the code looks very much like the actual formulation.

Next is an implementation of the formulation we just saw. The code is available in the Mip Master repository on GitHub [link].

```
1  import gurobipy as gp
2
3  # Define the model
4  mdl = gp.Model('Pick a Tech')
5
6  # Add variables
7  x = mdl.addVars([1, 2, 3], vtype=gp.GRB.BINARY, name='x')
8
9  # Add Constraints
10 mdl.addConstr(x[1] + x[2] + x[3] == 1, name='exactly_one_tech')
11
12 # Set the objective function
13 mdl.setObjective(12*x[1] + 17*x[2] + 25*x[3], sense=gp.GRB.MAXIMIZE)
14
15 # Optimize
16 mdl.optimize()
17
18 # Retrieve the solution
21 x_sol = {i: x[i].X for i in [1, 2, 3]}
22 print(x_sol)
```

This code is very readable, but let me explain each line:

- In Line 1, Mr. Mip imports the gurobipy package.
- In Line 4, he initializes a model instance and gives it a name.
- In Line 7, he adds the three decision variables at once from a list of indices and give it a name.
- In Line 10, he adds the only constraint of the model and give it a name too.
- In Line 13, he sets the objective function.
- In Line 16, when the model is fully populated, Mr. Mip call the optimizer.
- Finally, in Line 21 and 22, he retrieves and prints out the solution.

The output is a dictionary that look like this:

```
{1: 0.0, 2: 0.0, 3: 1.0}.
```

From where Mr. Mip concludes that the optimal solution is:

$$x_1 = 0, \qquad x_2 = 0, \qquad x_3 = 1,$$

Which means that apps is the right technology for TicTech, as we already knew.

Challenge yourself

1) Convince yourself that the following are also correct formulations for this use case.

$$(I) \quad \begin{array}{ll} \max & 12\,x_1 + 17\,x_2 + 25\,x_3 \\ s.t. & x_1 + x_2 + x_3 \le 1 \\ & x_1, x_2, x_3 \in \{0,1\}. \end{array}$$

$$(II) \quad \begin{array}{ll} \max & 12\,x_1 + 17\,x_2 + 25\,x_3 \\ s.t. & x_1 + x_2 \le 1 \\ & x_1 + x_3 \le 1 \\ & x_2 + x_3 \le 1 \\ & x_1, x_2, x_3 \in \{0,1\}. \end{array}$$

2) Implement the two new formulations in 1 to see that you obtain the same solution.
3) In your implementation, try to retrieve the solution using these alternative methods:

```
# Retrieve the solution
x_sol = {x[i].VarName: x[i].X for i in x.keys()}
print(x_sol)
```

```
# Retrieve the solution
x_sol = mdl.getAttr('X', x)
print(x_sol)
```

Takeaways

1. There are three major steps in solving problems using MIP:
    1. Understanding the problem.
    2. Writing a mathematical formulation.
    3. Implementing and solving the optimization model.
2. A typical formulation has three main components:
    1. Decision variables
    2. Constraints
    3. Objective function
3. There may be multiple ways to formulate and solve the same problem.
4. It’s easy to code and solve an optimization model using a solver!