EM384: Analytical Methods for Engineering Management

Lesson 24: Binary Decision Variables

27 March 2023

Table of contents

Lesson Objectives

Lesson Objectives

- Understand Integer Programming as a subset of optimization problems
- Describe how binary decision variables are used to represent yes-or-no decisions.
- Explain the difficulties with models requiring integer variables.
- Use binary variables and logical relationships in integer programming.

Integer Programming

Integer Programming

- So far, we have only considered continuous decision variables.
- Sometimes, the decision variables only make sense as integers, e.g. number of aircraft, location of a police station, etc.
- We can designate integer variables using *Data > Solver > Constraints > "Int"*
- Sensitivity Report and Limits Report are NOT meaningful for problems with integer decision variables.
- Solution algorithms are more complex, and take longer to run

Why can't we just round the solution of a Linear Program?

- Rounded point may not be feasible.
- Rounded point may be feasible but NOT optimal.

Integer Programming

Example integer program. Constraints are still linear.

Decision Variables:

x_1, x_2, x_3, x_4

Objective Function:

Minimize $Z = 2x_1 + 3x_2 + x_3 + x_4$

Constraints:

$2x_1 + x_2 + x_3 \leq 20$

$x_3 = 4x_4$

$x_1, x_2, x_3, x_4 \in \mathbb{Z}$ (where $\mathbb{Z}$ denotes the set of non-negative integers)

Mixed-Integer Programming

You can also combine integer and continuous decision variables in a mixed-integer program formulation. Constraints are still linear!

Decision Variables:

x_1, x_2, x_3, x_4

Objective Function:

Minimize $Z = 2x_1 + 3x_2 + x_3 + x_4$

Constraints:

$2x_1 + x_2 + x_3 \leq 20$

$x_3 = 4x_4$

$x_1, x_2 \geq 0$

$x_3, x_4 \in \mathbb{Z}$ (where $\mathbb{Z}$ denotes the set of non-negative integers)

Binary Integer Programming

Binary Integer Programming

Binary decision variables are used to represent:

- Binary Choice
- Go/No-Go
- Accept/Reject
- Yes/No
  We can design binary decision variables using *Data > Solver > Constraints > "Bin"*

Binary Integer Programming

If all decision variables must be binary (either 0 or 1), this is known as a binary integer programming. We will usually denote binary decision variables with the letter y to avoid confusion. Constraints are still linear!

Decision Variables:

y_1, y_2, y_3, y_4

Objective Function:

Minimize $Z = 10y_1 + 10y_2 + 15y_3 + 5y_4$

Constraints:

$y_1 \geq y_2$

$y_3 + y_4 \leq 1$

$y_1 + y_2 = 1$

$y_1, y_2, y_3, y_4 \in \{0, 1\}$

Mixed Binary Integer Programming

If only some of the decision variables must be binary (either 0 or 1), this is known as a mixed binary integer programming. We will usually denote binary decision variables with the letter y to avoid confusion. Constraints are still linear!

Decision Variables:

x_1, x_2, x_3, y_1, y_2

Objective Function:

Minimize $Z = 3x_1 + 4x_2 + 7x_3 + 10y_1$

Constraints:

$x_1 + x_2 \leq 5y_2$

$x_1 + x_3 \leq 10y_1$

$y_1 + y_2 = 1$

$x_1, x_2, x_3 \geq 0$

$y_1, y_2 \in \{0, 1\}$

Constraint formulation using Binary Decision Variables

- We can enforce relationships with the help of binary decision variables
- For example:
  - At least m projects must be selected
  - At most n projects must be selected
  - Exactly k projects must be selected
  - Some projects are mutually exclusive
  - Some project have contingency relationships

Constraint formulation using Binary Decision Variables

Let y_1, y_2, and y_3 be three projects that we are considering for selection. $y_i = 0$ means project i is not selected and $y_i = 1$ means that project i is selected. Let x be a continuous non-negative decision variable. Examples of constraint formulation:

- At least m projects must be selected: $y_1 + y_2 + y_3 \geq m$
- At most n projects must be selected: $y_1 + y_2 + y_3 \leq n$
- Exactly k projects must be selected: $y_1 + y_2 + y_3 = k$
- If project 1 is selected, project 2 cannot be selected: $y_1 + y_2 \leq 1$
- If project 1 is selected, project 2 must also be selected: $y_2 \geq y_1$
- Either both projects 1 and 2 are selected, or none are selected: $y_1 = y_2$
- Project 1 must be selected if any other projects are selected: $2y_1 \geq y_2 + y_3$
- decision variable x must be zero, unless project 3 is selected (in which case it can be any number between 0 and 1000): $x \leq 1000y_3$

In-class exercise

Constraint formulation using Binary Decision Variables

Let z_1, z_2 be two machines ($z_i = 0$ means machine i is not running and $z_i = 1$ means machine i is running) and y_1, y_2, and y_3 be three workers ($y_i = 0$ means worker i is not on duty and $y_i = 1$ means worker i is on duty). Design a single constraint that satisfies the following requirement:

There must be at least one worker on duty at all times

Constraint formulation using Binary Decision Variables

Let z_1, z_2 be two machines ($z_i = 0$ means machine i is not running and $z_i = 1$ means machine i is running) and y_1, y_2, and y_3 be three workers ($y_i = 0$ means worker i is not on duty and $y_i = 1$ means worker i is on duty). Design a single constraint that satisfies the following requirement:

There must be at least one worker on duty at all times

$y_1 + y_2 + y_3 \geq 1$

Constraint formulation using Binary Decision Variables

Let z_1, z_2 be two machines ($z_i = 0$ means machine i is not running and $z_i = 1$ means machine i is running) and y_1, y_2, and y_3 be three workers ($y_i = 0$ means worker i is not on duty and $y_i = 1$ means worker i is on duty). Design a single constraint that satisfies the following requirement:

If machine 2 is running, worker 3 cannot be on duty

Constraint formulation using Binary Decision Variables

Let z_1, z_2 be two machines ($z_i = 0$ means machine i is not running and $z_i = 1$ means machine i is running) and y_1, y_2, and y_3 be three workers ($y_i = 0$ means worker i is not on duty and $y_i = 1$ means worker i is on duty). Design a single constraint that satisfies the following requirement:

If machine 2 is running, worker 3 cannot be on duty

$z_2 + y_3 \leq 1$

Constraint formulation using Binary Decision Variables

Let z_1, z_2 be two machines ($z_i = 0$ means machine i is not running and $z_i = 1$ means machine i is running) and y_1, y_2, and y_3 be three workers ($y_i = 0$ means worker i is not on duty and $y_i = 1$ means worker i is on duty). Design a single constraint that satisfies the following requirement:

Less than 3 workers are on duty at the same time

Constraint formulation using Binary Decision Variables

Let z_1, z_2 be two machines ($z_i = 0$ means machine i is not running and $z_i = 1$ means machine i is running) and y_1, y_2, and y_3 be three workers ($y_i = 0$ means worker i is not on duty and $y_i = 1$ means worker i is on duty). Design a single constraint that satisfies the following requirement:

Less than 3 workers are on duty at the same time

$y_1 + y_2 + y_3 \leq 2$

Note: Cannot say $y_1 + y_2 + y_3 < 3$ since we must only use $\leq$, $\geq$ or $=$

Constraint formulation using Binary Decision Variables

Let z_1, z_2 be two machines ($z_i = 0$ means machine i is not running and $z_i = 1$ means machine i is running) and y_1, y_2, and y_3 be three workers ($y_i = 0$ means worker i is not on duty and $y_i = 1$ means worker i is on duty). Design a single constraint that satisfies the following requirement:

If machine 1 is running, then worker 1 has to be on duty

Constraint formulation using Binary Decision Variables

Let z_1, z_2 be two machines ($z_i = 0$ means machine i is not running and $z_i = 1$ means machine i is running) and y_1, y_2, and y_3 be three workers ($y_i = 0$ means worker i is not on duty and $y_i = 1$ means worker i is on duty). Design a single constraint that satisfies the following requirement:

If machine 1 is running, then worker 1 has to be on duty

$z_1 \leq y_1$

Constraint formulation using Binary Decision Variables

Let z_1, z_2 be two machines ($z_i = 0$ means machine i is not running and $z_i = 1$ means machine i is running) and y_1, y_2, and y_3 be three workers ($y_i = 0$ means worker i is not on duty and $y_i = 1$ means worker i is on duty). Design a single constraint that satisfies the following requirement:

Exactly one machine must be running

Constraint formulation using Binary Decision Variables

Let z_1, z_2 be two machines ($z_i = 0$ means machine i is not running and $z_i = 1$ means machine i is running) and y_1, y_2, and y_3 be three workers ($y_i = 0$ means worker i is not on duty and $y_i = 1$ means worker i is on duty). Design a single constraint that satisfies the following requirement:

Exactly one machine must be running

$z_1 + z_2 = 1$

Constraint formulation using Binary Decision Variables

Let z_1, z_2 be two machines ($z_i = 0$ means machine i is not running and $z_i = 1$ means machine i is running) and y_1, y_2, and y_3 be three workers ($y_i = 0$ means worker i is not on duty and $y_i = 1$ means worker i is on duty). Design a single constraint that satisfies the following requirement:

Worker 2 and worker 1 always work together

Constraint formulation using Binary Decision Variables

Let z_1, z_2 be two machines ($z_i = 0$ means machine i is not running and $z_i = 1$ means machine i is running) and y_1, y_2, and y_3 be three workers ($y_i = 0$ means worker i is not on duty and $y_i = 1$ means worker i is on duty). Design a single constraint that satisfies the following requirement:

Workers 1 and 2 don't always work, but when they do, workers 1 and 2 always work together

$y_1 = y_2$

Constraint formulation using Binary Decision Variables

Let z_1, z_2, z_3, z_4, z_5 be five machines ($z_i = 0$ means machine i is not running and $z_i = 1$ means machine i is running). Design a single constraint that satisfies the following requirement:

Machine 1 must be running if any other machines are running

Constraint formulation using Binary Decision Variables

Let z_1, z_2, z_3, z_4, z_5 be five machines ($z_i = 0$ means machine i is not running and $z_i = 1$ means machine i is running). Design a single constraint that satisfies the following requirement:

Machine 1 must be running if any other machines are running

$4z_1 \geq z_2 + z_3 + z_4 + z_5$

Note: any number greater than or equal to 4 on the LHS works for this constraint.

Conclusion

Next Class

Homework:

- Read Chapter 12.2 - 12.3

Next Lesson:

- Recognize a site selection problem given a transportation problem.
- Formulate and solve in Excel a site selection problem using binary decision variables.