Lesson 14. IP Formulations

1 Solving Integer Programs can be *Really* Hard!

The following integer (linear) program (IP) seeks an objective-maximizing integer linear combination of a big number.

$$\begin{aligned} \text{maximize} \quad & 213x_1 - 1928x_2 - 11111x_3 - 2345x_4 + 9123x_5 \\ \text{subject to} \quad & 12223x_1 + 12224x_2 + 36674x_3 + 61119x_4 + 85569x_5 = 89643482 \\ & x_1,\ x_2,\ x_3,\ x_4,\ x_5 \geq 0,\ \text{integer} \end{aligned}$$

If we implement this problem in python (using GLPK), and solve it as a **linear program** we obtain the following solution relatively fast: `(0,0,0,0,1047.62)`.

If you solve it as an integer program (using GLPK) it can take **hours** to solve!

- Only has 5 variables
- Problems like this are the ones I look at in my research :)

In general, IPs are **significantly harder** to solve than LPs.

- In the next two lessons, we will discuss why IPs are harder than LPs and why the way we model IP problems can impact solver performance.
- In lesson 16 we will learn about the **branch and bound algorithm** which can be used to solve IPs.

2 Review Linear Programming Solution Techniques

2.1 Types of LP solutions

Theorem: Every LP's solution is EXACTLY one of the following:

1. Unique optimal solution
2. Multiple optimal solutions
3. Unbounded LP
4. Infeasible LP

Problem 1. Sketch graphs which illustrate each of the types of LP solutions.

These types of solutions are also true for integer programs

LPs are solved via the simplex method. Small LPs can be solved graphically.

Problem 2. Review: Solve the following LP graphically:

$$\begin{aligned} \max \quad & x_1 + x_2 \\ \text{st} \quad & x_1 + 2x_2 \leq 10 \\ & x_1 \leq 4 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Theorem: If an LP has an optimal solution (i.e., it is not unbounded or infeasible), the optimal solution occurs at a **corner point** of the feasible region.

Question: Is the same true for an IP?

3 Integer Program Formulations

A **formulation** of an integer (linear) program is a set of linear $\boxed{}$ that capture ALL of the $\boxed{}$ integer points, and NO OTHER integer points.

The **LP relaxation** of an IP is the LP that is formed by relaxing (i.e., removing) the integer requirement on the variables.

Problem 3. Below are two integer programs, along with the diagrams of their constraints.

Integer Program A

$$\begin{aligned} \text{maximize} \quad & 8x + 7y \\ \text{subject to} \quad & -18x + 38y \le 133 \\ & 13x + 11y \le 125 \\ & 10x - 8y \le 55 \\ & x,\ y \ge 0,\ \text{integer} \end{aligned}$$

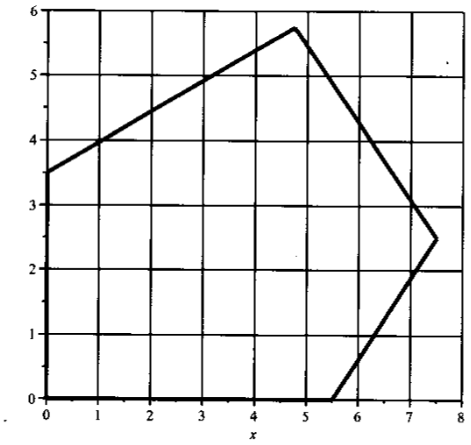

FIGURE 13.1 Feasible region for integer program (13.3).

Integer Program B

$$\begin{aligned} \text{maximize} \quad & 8x + 7y \\ \text{subject to} \quad & -x + 2y \le 6 \\ & x + y \le 10 \\ & x - y \le 5 \\ & x \le 7 \\ & y \le 5 \\ & x,\ y \ge 0,\ \text{integer} \end{aligned}$$

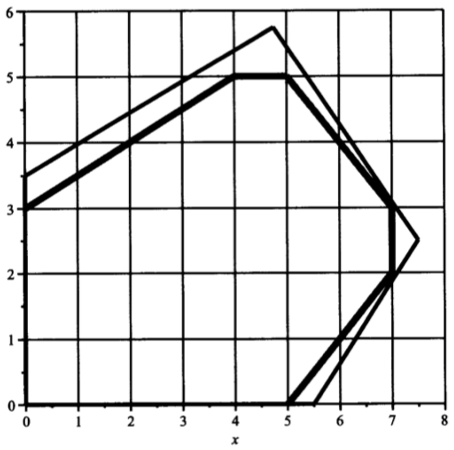

FIGURE 13.2 Feasible region for integer program (13.4).

(a) On the diagrams, identify all feasible solutions to both IPs.

(b) Are the integer feasible regions for IP A and IP B different or the same?

(c) Are the feasible regions of the LP relaxations of IP A and IP B different or the same?

(d) What does this mean about problems A and B?

(e) Based on these graphs, will the optimal solution of an IP always occur at a corner point?

(f) Which of these formulations is easier to solve? Why?

3.1 Better Formulation ⇒ Better Bound

Now let's consider the relationship between an IP and its LP relaxation.

In general:

- If we are solving a **maximization** IP, the solution of its LP relaxation provides a bound on the solution of the IP problem.
- If we are solving a **minimization** IP, the solution of its LP relaxation provides a bound on the solution of the IP problem.

The **tighter** a formulation, the bound you obtain via the LP relaxation.

This idea is key for solving IPs!

Problem 4. Sketch a problem which proves if we're maximizing, $z_{LP} \geq z_{IP}$ and vice versa if minimizing.

Often the decision of how to formulate an IP comes down to a tradeoff between the formulation quality and number of constraints.

- More constraints can lead to a better (tighter) formulation, but:

- Fewer constraints lead to a weaker formulation but: