CHAPTER 12

Discrete Optimization Methods

In Chapter 11 we illustrated the wide range of integer and combinatorial optimization models encountered in operations research practice. Some are linear programs with a few discrete side constraints; others are still linear but involve only combinatorial decision variables; still others are both nonlinear and combinatorial. Every one includes logical decisions that just cannot be modeled validly as continuous, so most lack the elegant tractability of the LP and network models studied in earlier chapters.

Diminished tractability does not imply dimished importance. Discrete optimization models such as those presented in Chapter 11 all represent critical decision problems in engineering and management that must somehow be confronted. Even partial analysis can prove enormously valuable.

It should not surprise that discrete optimization methods span a range as wide as the models they address. In contrast to, say, linear programming, where a few prominent algorithms have proved adequate for the overwhelming majority of models, success in discrete optimization often requires methods cleverly specialized to an individual application. Still, there are common themes. In this chapter we introduce the best known.

12.1 SOLVING BY TOTAL ENUMERATION

Beginning students often find counterintuitive the idea that discrete optimization problems are more difficult than their continuous analogs. The algebra of LP algorithms in Chapters 5 and 6 is rather daunting. By comparison, a discrete model, which has only a finite number of choices for decision variables, can seem refreshingly easy. Why not just try them all and keep the best feasible solution as optimal?

Although naive, this point of view contains a kernel of wisdom.

> **12.1** If model has only a few discrete decision variables, the most effective method of analysis is often the most direct: enumeration of all the possibilities.

Total Enumeration

To be more specific, total or complete enumeration requires checking all possibilities implied by discrete variable values.

> 12.2 **Total enumeration** solves a discrete optimization by trying all possible combinations of discrete variable values, computing for each the best corresponding choice of any continuous variables. Among combinations yielding a feasible solution, those with the best objective function value are optimal.

Swedish Steel All-or-Nothing Example

We can illustrate with the discrete version of our Swedish Steel example formulated in model (11.2) (Section 11.1):

$$
\begin{array}{lll}
\min & 16(75)y_1 + 10(250)y_2 + 8x_3 + 9x_4 + 48x_5 + 60x_6 + 53x_7 & \\
\text{s.t.} & 75y_1 + 250y_2 + x_3 + x_4 + x_5 + x_6 + x_7 & = 1000 \\
& 0.0080(75)y_1 + 0.0070(250)y_2 + 0.0085x_3 + 0.0040x_4 & \geq 0.0065(1000) \\
& 0.0080(75)y_1 + 0.0070(250)y_2 + 0.0085x_3 + 0.0040x_4 & \leq 0.0075(1000) \\
& 0.180(75)y_1 + 0.032(250)y_2 + 1.0x_5 & \geq 0.030(1000) \\
& 0.180(75)y_1 + 0.032(250)y_2 + 1.0x_5 & \leq 0.035(1000) \\
& 0.120(75)y_1 + 0.011(250)y_2 + 1.0x_6 & \geq 0.010(1000) \\
& 0.120(75)y_1 + 0.011(250)y_2 + 1.0x_6 & \leq 0.012(1000) \\
& 0.001(250)y_2 + 1.0x_7 & \geq 0.011(1000) \\
& 0.001(250)y_2 + 1.0x_7 & \leq 0.013(1000) \\
& x_3, \ldots, x_7 \geq 0 & \\
& y_1, y_2 = 0 \text{ or } 1 &
\end{array}
\qquad (12.1)
$$

In this version the first two sources of scrap iron have to be entered on an all-or-nothing basis modeled with discrete variables. The other five sources can be employed in any nonnegative amount.

There are 2 possible values for y_1 and 2 for y_2, or a total of $2 \cdot 2 = 4$ combinations to enumerate. Table 12.1 provides details. Third option $y_1 = 1$, $y_2 = 0$ yields the optimal solution with objective value 9540.3.

TABLE 12.1 Enumeration of the Swedish Steel All-or-Nothing Model

Discrete Combination	Corresponding Continuous Solution	Objective Value
$y_1 = 0,\ y_2 = 0$	$x_3 = 814.3,\ x_4 = 114.6,\ x_5 = 30.0,\ x_6 = 10.0,\ x_7 = 1.1$	9914.1
$y_1 = 0,\ y_2 = 1$	$x_3 = 637.9,\ x_4 = 82.0,\ x_5 = 22.0,\ x_6 = 7.3,\ x_7 = 0.9$	9877.3
$y_1 = 1,\ y_2 = 0$	$x_3 = 727.6,\ x_4 = 178.8,\ x_5 = 16.5,\ x_6 = 1.0,\ x_7 = 1.1$	9540.3
$y_1 = 1,\ y_2 = 1$	$x_3 = 552.8,\ x_4 = 112.9,\ x_5 = 8.5,\ x_6 = 0.0,\ x_7 = 0.9$	9591.1

Since this model has both discrete and continuous variables, each case enumerated requires solving a continuous optimization over variables $x_3, \ldots, x_7$ to find the best continuous values to go with the choice of discrete variables selected. For example, fixing $y_1 = y_2 = 0$ in model (12.1) leaves the linear program

$$
\begin{array}{lll}
\min & 16(75)(0) + 10(250)(0) + 8x_3 + 9x_4 + 48x_5 + 60x_6 + 53x_7 & \\
\text{s.t.} & 75(0) + 250(0) + x_3 + x_4 + x_5 + x_6 + x_7 & = 1000 \\
& 0.0080(75)(0) + 0.0070(250)(0) + 0.0085x_3 + 0.0040x_4 & \geq 0.0065(1000) \\
& 0.0080(75)(0) + 0.0070(250)(0) + 0.0085x_3 + 0.0040x_4 & \leq 0.0075(1000) \\
& 0.180(75)(0) + 0.032(250)(0) + 1.0x_5 & \geq 0.030(1000) \\
& 0.180(75)(0) + 0.032(250)(0) + 1.0x_5 & \leq 0.035(1000) \\
& 0.120(75)(0) + 0.011(250)(0) + 1.0x_6 & \geq 0.010(1000) \\
& 0.120(75)(0) + 0.011(250)(0) + 1.0x_6 & \leq 0.012(1000) \\
& 0.001(250)(0) + 1.0x_7 & \geq 0.011(1000) \\
& 0.001(250)(0) + 1.0x_7 & \leq 0.013(1000) \\
& x_3, \ldots, x_7 \geq 0 &
\end{array}
$$

Optimal solution $x_3 = 814.3$, $x_4 = 144.6$, $x_5 = 30.0$, $x_6 = 10.0$, $x_7 = 1.1$, completes the first case in Table 12.1.

> **SAMPLE EXERCISE 12.1: SOLVING BY TOTAL ENUMERATION**
>
> Solve the following discrete optimization model by total enumeration 12.2.
>
> $$
> \begin{array}{ll}
> \max & 7x_1 + 4x_2 + 19x_3 \\
> \text{s.t.} & x_1 + x_3 \leq 1 \\
> & x_2 + x_3 \leq 1 \\
> & x_1, x_2, x_3 = 0 \text{ or } 1
> \end{array}
> $$
>
> ***Analysis:*** Checking the $2^3 = 8$ combinations produces the following table:
>
> | Case | Objective | Case | Objective |
> |---|---|---|---|
> | $\mathbf{x} = (0, 0, 0)$ | 0 | $\mathbf{x} = (1, 0, 0)$ | 7 |
> | $\mathbf{x} = (0, 0, 1)$ | 19 | $\mathbf{x} = (1, 0, 1)$ | Infeasible |
> | $\mathbf{x} = (0, 1, 0)$ | 4 | $\mathbf{x} = (1, 1, 0)$ | 11 |
> | $\mathbf{x} = (0, 1, 1)$ | Infeasible | $\mathbf{x} = (1, 1, 1)$ | Infeasible |
>
> Solution $\mathbf{x} = (0, 0, 1)$ is the feasible one with best objective value 19, so it is optimal.

Exponential Growth of Cases to Enumerate

Our Swedish Steel example has two discrete decision variables, each with two possible values 0 and 1. A total of

$$2 \cdot 2 = 2^2 = 4$$

combinations result.

Similar thinking shows that a model with k binary decision variables would have

$$\underbrace{2 \cdot 2 \cdots 2}_{k \text{ times}} = 2^k$$

cases to enumerate. This is **exponential growth**, with every additional 0–1 variable doubling the number of combinations.

An analyst can easily run $2^2 = 4$ or $2^4 = 16$ cases—perhaps even $2^{10} = 1024$ with the aid of a computer. But $2^{100} \approx 10^{30}$, and we know that a discrete model with $k = 100$ binary variables is not particularly large.

The fastest current computers perform a few billion (10^9) arithmetic operations in a second. Future computers might well solve the entire linear program associated with each choice of discrete variables in that same time. Assume even more, that a trillion (10^{12}) cases could be checked in a single second. Enumeration of a 100-variable model would still require

$$\frac{2^{100}}{10^{12}} \approx 1.27 \times 10^{18} \text{ seconds} \approx 402 \text{ million centuries}$$

too long for the most patient of decision makers to wait.

12.3 Exponential growth makes total enumeration impractical with models having more than a handful of discrete decision variables.

SAMPLE EXERCISE 12.2: UNDERSTANDING EXPONENTIAL GROWTH

Suppose that your personal computer can enumerate one combination of discrete values each second of a given mixed-integer program, including solving the implied optimization for corresponding continuous variable values. Determine how long it would take to totally enumerate instances with 10, 20, 30, and 40 binary variables.

Analysis: For 10 variables, enumeration would require

$$2^{10} = 1024 \text{ seconds} \approx 17.1 \text{ minutes}$$

Each increment of 10 binary variables multiplies the number of combinations by $2^{10} = 1024$. Thus a case with 20 variables would require

$$1024 \cdot 17.1 \approx 17,500 \text{ minutes} \approx 12.1 \text{ days}$$

An instance with 30 binary variables would need about $1024 \cdot 12.1 = 12{,}390$ days, and one with 40 variables would require nearly 12.7 million days, more than 347 centuries.

Nonlinearities

The practicality of enumeration in mixed cases is also limited by the continuous problem that remains when discrete variables are enumerated. Our Swedish Steel case was an integer linear program (ILP), so that each case involved only solving an LP. If the remaining continuous model had been nonlinear, even evaluating the cases could have been a difficult task.

12.2 RELAXATIONS OF DISCRETE OPTIMIZATION MODELS AND THEIR USES

Because analysis of discrete optimization models is usually hard, it is natural to look for related but easier formulations that can aid in the analysis. **Relaxations** are auxiliary optimization models of this sort formed by weakening either the constraints or the objective function of the given discrete model.

EXAMPLE 12.1: BISON BOOSTERS

Before considering relaxation in the more realistic circumstances of models in Chapter 11, it will help to develop a more compact (albeit highly artificial) example. Consider the dilemma of the Bison Boosters club supporting the local atheletic team.

The Boosters are trying to decide what fundraising projects to undertake at the next country fair. One option is customized T-shirts, which will sell for \$20 each; the other is sweatshirts selling for \$30. History shows that everything offered for sale will be sold before the fair is over.

Materials to make the shirts are all donated by local merchants, but the Boosters must rent the equipment for customization. Different processes are involved, with the T-shirt equipment renting at \$550 for the period up to the fair, and the sweatshirt equipment for \$720. Display space presents another consideration. The Boosters have only 300 square feet of display wall area at the fair, and T-shirts will consume 1.5 square feet each, sweatshirts 4 square feet each. What plan will net the most income?

Certainly this problem centers on making shirts, so decision variables will include

$$x_1 \triangleq \text{number of T-shirts made and sold}$$
$$x_2 \triangleq \text{number of sweatshirts made and sold}$$

However, the Boosters also confront discrete decisions on whether to rent equipment:

$$y_1 \triangleq 1 \text{ if T-shirt equipment is rented and} = 0 \text{ otherwise}$$
$$y_2 \triangleq 1 \text{ if sweatshirt equipment is rented and} = 0 \text{ otherwise}$$

Using these decision variables, the Boosters' dilemma can be modeled:

$$\begin{array}{lll} \max & 20x_1 + 30x_2 - 550y_1 - 720y_2 & \text{(net income)} \\ \text{s.t.} & 1.5x_1 + 4x_2 \le 300 & \text{(display space)} \\ & x_1 \le 200y_1 & \text{(T-shirts if equipment)} \\ & x_2 \le 75y_2 & \text{(sweatshirts if equipment)} \\ & x_1, x_2 \ge 0 & \\ & y_1, y_2 = 0 \text{ or } 1 & \end{array} \qquad (12.2)$$

The objective function maximizes net income, and the first main constraint enforces the display space limit. The next two constraints provide the switching we have seen in other models. Any sufficiently large big-M could be used as the y_j coefficient in these constraints. Values in (12.2) derive from the greatest production possible within the 300 square feet display limit. Coefficients $300/1.5 = 200$ for T-shirts and $300/4 = 75$ for sweatshirts introduce no limitation if y's equal 1, yet switch off all production if y's equal 0.

Enumeration of the 4 combinations of y_1 and y_2 values easily establishes that the Boosters should make only T-shirts. The unique optimal solution is $x_1^* = 200$, $x_2^* = 0$, $y_1^* = 1$, $y_2^* = 0$, with net income \$3450.

Constraint Relaxations

Relaxations may weaken either the objective function or the constraints, but the elementary ones we explore in this book nearly all focus on constraints. A constraint relaxation produces an easier model by dropping or easing some constraints.

> 12.4 Model $(\bar{P})$ is a **constraint relaxation** of model (P) if every feasible solution to (P) is also feasible in $(\bar{P})$ and both models have the same objective function.

New feasible solutions may be allowed, but none should be lost.

Table 12.2 shows several constraint relaxations of the tiny Bison Boosters model (12.2). The first simply doubles capacities. The result is certainly a relaxation, because every solution fitting within the true capacity of 300 square feet will also fit within twice as much area. Still, this relaxation gains us little.

TABLE 12.2 Constraint Relaxations of Bison Boosters Model

Revised Constraints	Discussion
$1.5x_1 + 4x_2 \le 600$ $x_1 \le 400y_1$ $x_2 \le 150y_2$ $x_1, x_2 \ge 0$ $y_1, y_2 = 0$ or 1	Doubled capacities. Relaxation optimum: $\tilde{x}_1 = 400, \tilde{x}_2 = 0, \tilde{y}_1 = 1, \tilde{y}_2 = 0$, net income \$7450
$x_1 \le 200y_1$ $x_2 \le 75y_2$ $x_1, x_2 \ge 0$ $y_1, y_2 = 0$ or 1	Dropped first constraint. Relaxation optimum: $\tilde{x}_1 = 200, \tilde{x}_2 = 75, \tilde{y}_1 = 1, \tilde{y}_2 = 1$, net income \$4980
$1.5x_1 + 4x_2 \le 300$ $x_1 \le 200y_1$ $x_2 \le 75y_2$ $x_1, x_2 \ge 0$ $0 \le y_1 \le 1$ $0 \le y_2 \le 1$	Linear programming relaxation with discrete variables treated as continuous. Relaxation optimum: $\tilde{x}_1 = 200, \tilde{x}_2 = 0, \tilde{y}_1 = 1, \tilde{y}_2 = 0$, net income \$3450

> 12.5 Relaxations should be significantly more tractable than the models they relax, so that deeper analysis is practical.

Doubling capacities fails this requirement because the character of the model is unchanged.

The second relaxation of Table 12.2 is more on track. Dropping the first constraint delinks decisions about the two types of shirts. It then becomes much easier to compute a (relaxation) optimal solution. We need only decide one by one whether the maximum production now allowed each x_j when its $y_j = 1$ justifies the fixed cost of equipment rental. Both do.

SAMPLE EXERCISE 12.3: RECOGNIZING RELAXATIONS

Determine whether or not each of the following mixed-integer programs is a constraint relaxation of

$$\begin{aligned} \min \quad & 3x_1 + 6x_2 + 7x_3 + x_4 \\ \text{s.t.} \quad & 2x_1 + x_2 + x_3 + 10x_4 \ge 100 \\ & x_1 + x_2 + x_3 \le 1 \\ & x_1, x_2, x_3 = 0 \text{ or } 1 \\ & x_4 \ge 0 \end{aligned}$$

(a)
$$\begin{aligned} \min \quad & 3x_1 + 6x_2 + 7x_3 + x_4 \\ \text{s.t.} \quad & 2x_1 + x_2 + x_3 + 10x_4 \ge 100 \\ & x_1, x_2, x_3 = 0 \text{ or } 1 \\ & x_4 \ge 0 \end{aligned}$$

(b)
$$\begin{aligned} \min \quad & 3x_1 + 6x_2 + 7x_3 + x_4 \\ \text{s.t.} \quad & 2x_1 + x_2 + x_3 + 10x_4 \ge 200 \\ & x_1 + x_2 + x_3 \le 1 \\ & x_1, x_2, x_3 = 0 \text{ or } 1 \\ & x_4 \ge 0 \end{aligned}$$

(c)
$$\begin{aligned} \min \quad & 3x_1 + 6x_2 + 7x_3 + x_4 \\ \text{s.t.} \quad & 2x_1 + x_2 + x_3 + 10x_4 \ge 100 \\ & x_1 + x_2 + x_3 \le 1 \\ & x_1, x_2, x_3, x_4 \ge 0 \end{aligned}$$

(d)
$$\begin{aligned} \min \quad & 3x_1 + 6x_2 + 7x_3 + x_4 \\ \text{s.t.} \quad & 2x_1 + x_2 + x_3 + 10x_4 \ge 100 \\ & x_1 + x_2 + x_3 \le 1 \\ & 1 \ge x_1 \ge 0,\ 1 \ge x_2 \ge 0, \\ & 1 \ge x_3 \ge 0,\ x_4 \ge 0 \end{aligned}$$

Analysis: We apply definition 12.4.

(a) This model is a constraint relaxation because it is formed by dropping the second main constraint. Certainly, every solution feasible in the original model remains so with fewer constraints.

(b) This model is not a relaxation. The only change, which is increasing the right-hand side by 100, to 200, eliminates previously feasible solutions. One example is $\mathbf{x} = (0, 0, 0, 10)$.

(c) This model is a relaxation. Allowing x_1, x_2, and x_3 to take on any nonnegative value—rather than just 0 or 1—cannot eliminate previously feasible solutions.

(d) This model is also a relaxation. Allowing x_1, x_2, and x_3 to take on any values in the interval [0, 1] precludes none of their truly feasible values.

Linear Programming Relaxations

The third case in Table 12.2 illustrates the best known and most used of all constraint relaxation forms: linear programming, or more generally, continuous relaxations.

> 12.6 **Continuous relaxations (linear programming relaxations)** if the given model is an integer linear program) are formed by treating any discrete variables as continuous while retaining all other constraints.

In the real Bison Boosters model, each y_j must equal 0 or 1. In the continuous relaxation we also admit fractions, replacing each

$$y_j = 0 \text{ or } 1 \quad \text{by} \quad 1 \ge y_j \ge 0$$

Certainly, no feasible solutions are lost by allowing both fractional and integer choices for discrete variables, so the process does produce a valid relaxation. More important, the relaxed model usually proves significantly more tractable.

Our Bison Boosters model is an integer linear program (ILP), linear in all aspects except the discreteness of y_1 and y_2. Thus, relaxing discrete variables to continuous leaves a linear program to solve—the linear programming relaxation; we have already expended several chapters of this book showing how effectively linear programs can be analyzed.

12.7 LP relaxations of integer linear programs are by far the most used relaxation forms because they bring all the power of linear programming to bear on analysis of the given discrete models.

SAMPLE EXERCISE 12.4: FORMING LINEAR PROGRAMMING RELAXATIONS

Form the linear programming relaxation of the following mixed-integer program:

$$\begin{array}{ll} \min & 15x_1 + 2x_2 - 4x_3 + 10x_4 \\ \text{s.t.} & x_3 - x_4 \le 0 \\ & x_1 + 2x_2 + 4x_3 + 8x_4 = 20 \\ & x_2 + x_4 \le 1 \\ & x_1 \ge 0 \\ & x_2, x_3, x_4 = 0 \text{ or } 1 \end{array}$$

Analysis: Following definition **12.6**, we replace 0–1 constraints $x_j = 0$ or 1 by $x_j \in [0, 1]$ to obtain the LP relaxation

$$\begin{array}{ll} \min & 15x_1 + 2x_2 - 4x_3 + 10x_4 \\ \text{s.t.} & x_3 - x_4 \le 0 \\ & x_1 + 2x_2 + 4x_3 + 8x_4 = 20 \\ & x_2 + x_4 \le 1 \\ & x_1 \ge 0 \\ & 1 \ge x_2 \ge 0,\ 1 \ge x_3 \ge 0,\ 1 \ge x_4 \ge 0 \end{array}$$

Proving Infeasibility with Relaxations

Exactly what do relaxations add to our analysis of discrete optimization models? One thing is prove infeasibility.

Suppose that a constraint relaxation comes out infeasible. Then it has no solutions at all. Since every solution to the full model must also be feasible in the relaxation, it follows that the original model was also infeasible. By analyzing the relaxation we have learned a critical fact about the model of real interest.

12.8 If a constraint relaxation is infeasible, so is the full model it relaxes.

SAMPLE EXERCISE 12.5: PROVING INFEASIBILITY WITH RELAXATIONS

Use linear programming relaxation to establish that the following discrete optimization model is infeasible:

$$\begin{array}{ll} \min & 8x_1 + 2x_2 \\ \text{s.t.} & x_1 - x_2 \ge 2 \\ & -x_1 + x_2 \ge -1 \\ & x_1, x_2 \ge 0 \text{ and integer} \end{array}$$

Analysis: The linear programming relaxation of this model is

$$\begin{array}{ll} \min & 8x_1 + 2x_2 \\ \text{s.t.} & x_1 - x_2 \ge 2 \\ & -x_1 + x_2 \ge -1 \\ & x_1, x_2 \ge 0 \end{array}$$

It is clearly infeasible, because the two main constraints can be written

$$\begin{array}{l} x_1 - x_2 \ge 2 \\ x_1 - x_2 \le 1 \end{array}$$

Thus by principle **12.8**, the given integer program is also infeasible. Any solutions satisfying all constraints would also have to be feasible in the relaxation.

Solution Value Bounds from Relaxations

Figure 12.1 illustrates how relaxations also give us bounds on optimal solution values. Constraint relaxations expand the feasible set, allowing more candidates for relaxation optimum. The relaxation optimal value, which is the best over the expanded set of solutions, must then equal or improve on the best feasible solution value to the true model.

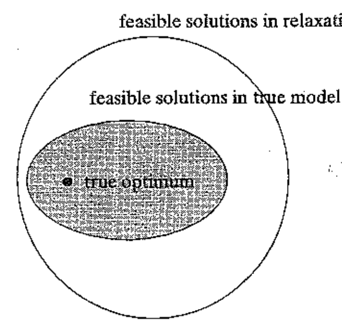

FIGURE 12.1 Relaxations and Optimality

12.9 The optimal value of any relaxation of a maximize model yields an upper bound on the optimal value of the full model. The optimal value of any relaxation of a minimize model yields a lower bound.

All three relaxations in Table 12.2 illustrate the maximize case. The optimal solution value of the Bison Boosters model (12.2) is \$3450. One of the cases in Table 12.2 yields exactly this value. The others produce higher estimates of net income. All provide the upper bound guaranteed in principle **12.9**.

With the Bison Boosters model, which is so small that it is easily solved optimally, relaxation bounds offer little new insight. A better sense of their value comes from considering the somewhat larger EMS model (11.8) of Section 11.3 that minimizes the number of stations to cover the 20 metropolitan districts:

$$\begin{array}{llll}
\min & \sum_{j=1}^{10} x_j & & \text{(number of sites)} \\
\text{s.t.} & x_2 & \geq 1 & \text{(district 1)} \\
& x_1 + x_2 & \geq 1 & \text{(district 2)} \\
& x_1 + x_3 & \geq 1 & \text{(district 3)} \\
& x_3 & \geq 1 & \text{(district 4)} \\
& x_3 & \geq 1 & \text{(district 5)} \\
& x_2 & \geq 1 & \text{(district 6)} \\
& x_2 + x_4 & \geq 1 & \text{(district 7)} \\
& x_3 + x_4 & \geq 1 & \text{(district 8)} \\
& x_8 & \geq 1 & \text{(district 9)} \\
& x_4 + x_6 & \geq 1 & \text{(district 10)} \\
& x_4 + x_5 & \geq 1 & \text{(district 11)} \\
& x_4 + x_5 + x_6 & \geq 1 & \text{(district 12)} \\
& x_4 + x_5 + x_7 & \geq 1 & \text{(district 13)} \\
& x_8 + x_9 & \geq 1 & \text{(district 14)} \\
& x_6 + x_9 & \geq 1 & \text{(district 15)} \\
& x_5 + x_6 & \geq 1 & \text{(district 16)} \\
& x_5 + x_7 + x_{10} & \geq 1 & \text{(district 17)} \\
& x_8 + x_9 & \geq 1 & \text{(district 18)} \\
& x_9 + x_{10} & \geq 1 & \text{(district 19)} \\
& x_{10} & \geq 1 & \text{(district 20)} \\
& x_1, \ldots, x_{10} = 0 \text{ or } 1 & &
\end{array} \tag{12.3}$$

How many stations does this model imply? Even with just 10 discrete variables, the answer is hardly obvious. But if we replace each $x_j = 0$ or 1 constraint by $0 \leq x_j \leq 1$, the resulting linear programming relaxation can be solved quickly with say, the simplex algorithm. An optimal solution is

$$\begin{aligned}
\tilde{x}_1 &= \tilde{x}_7 = 0 \\
\tilde{x}_2 &= \tilde{x}_3 = \tilde{x}_8 = \tilde{x}_{10} = 1 \\
\tilde{x}_4 &= \tilde{x}_5 = \tilde{x}_6 = \tilde{x}_9 = \tfrac{1}{2}
\end{aligned} \tag{12.4}$$

with optimal value 6.0. Without looking any further into the discrete model, we can conclude that at least 6 EMS sites will be required because this LP relaxation value provides a lower bound (principle 12.9).

SAMPLE EXERCISE 12.6: COMPUTING BOUNDS FROM RELAXATIONS

Compute (by inspection) the optimal solution value and the LP relaxation bound for each of the following integer programs.

(a)
$$\begin{array}{ll}
\max & x_1 + x_2 + x_3 \\
\text{s.t.} & x_1 + x_2 \leq 1 \\
& x_1 + x_3 \leq 1 \\
& x_2 + x_3 \leq 1 \\
& x_1, x_2, x_3 = 0 \text{ or } 1
\end{array}$$

(b)
$$\begin{array}{ll}
\min & 20x_1 + 9x_2 + 7x_3 \\
\text{s.t.} & 10x_1 + 4x_2 + 3x_3 \geq 7 \\
& x_1, x_2, x_3 = 0 \text{ or } 1
\end{array}$$

Analysis:

(a) Clearly, only one of the variables in this model can = 1, so the optimal solution value is 1. Corresponding linear programming relaxation

$$\begin{array}{ll}
\max & x_1 + x_2 + x_3 \\
\text{s.t.} & x_1 + x_2 \leq 1 \\
& x_1 + x_3 \leq 1 \\
& x_2 + x_3 \leq 1 \\
& 1 \geq x_1, x_2, x_3 \geq 0
\end{array}$$

yields optimal solution $\tilde{\mathbf{x}} = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ with objective value $\frac{3}{2}$. In accord with principle 12.9, relaxation value $\frac{3}{2}$ is an upper bound on the true optimal value 1 of this maximize model.

(b) Total enumeration shows that an optimal solution to this minimizing ILP is $\mathbf{x} = (0, 1, 1)$ with value 16. Its linear programming relaxation is

$$\begin{array}{ll}
\min & 20x_1 + 9x_2 + 7x3 \\
\text{s.t.} & 10x_1 + 4x_2 + 3x_3 \geq 7 \\
& 1 \geq x_1, x_2, x_3 \geq 0
\end{array}$$

with optimal solution $\tilde{\mathbf{x}} = (\frac{7}{10}, 0, 0)$ and value 14. Demonstrating principle 12.9, relaxation value 14 provides a lower bound on true optimal value 16.

Optimal Solutions from Relaxations

Sometimes relaxations not only bound the optimal value of the corresponding discrete model but produce an optimal solution.

12.10 If an optimal solution to a constraint relaxation is also feasible in the model it relaxes, the solution is optimal in that original model.

Another look at Figure 12.1 will show why. All (shaded-area) feasible solutions to the original discrete model must also belong to the larger relaxation feasible set. If the relaxation optimum happens to be one of them, it has as good an objective function value as any feasible solution to the relaxation. In particular, it has as good an objective function value as any feasible solution in the original model. It must be optimal in the full model.

The third, linear programming relaxation of the Bison Boosters model in Table 12.2 illustrates. Even though integrality was not required of y-components in the relaxation optimal solution

$$\tilde{x}_1 = 200, \quad \tilde{x}_2 = 0, \quad \tilde{y}_1 = 1, \quad \tilde{y}_2 = 0$$

it happened anyway. This relaxation optimum is feasible in the full discrete model and so optimal there.

SAMPLE EXERCISE 12.7: OBTAINING OPTIMAL SOLUTIONS FROM RELAXATIONS

Compute (by inspection) optimal solutions to each of the following relaxations, and determine whether we can conclude that the relaxation optimum is optimal in the original model.

(a) The linear programming relaxation of

$$\begin{array}{ll} \max & 20x_1 + 8x_2 + 2x_3 \\ \text{s.t.} & x_1 + x_2 + x_3 \leq 1 \\ & x_1, x_2, x_3 = 0 \text{ or } 1 \end{array}$$

(b) The linear programming relaxation of

$$\begin{array}{ll} \max & x_1 + x_2 + x_3 \\ \text{s.t.} & x_1 + x_2 \leq 1 \\ & x_1 + x_3 \leq 1 \\ & x_2 + x_3 \leq 1 \\ & x_1, x_2, x_3 = 0 \text{ or } 1 \end{array}$$

(c) The relaxation obtained by dropping the first main constraint of

$$\begin{array}{ll} \min & 2x_1 + 4x_2 + 8x_3 \\ \text{s.t.} & x_1 + x_2 + x_3 \leq 2 \\ & 10x_1 + 3x_2 + x_3 \geq 8 \\ & x_1, x_2, x_3 = 0 \text{ or } 1 \end{array}$$

Analysis:

(a) The linear programming relaxation of this model is

$$\begin{array}{ll} \max & 20x_1 + 8x_2 + 2x_3 \\ \text{s.t.} & x_1 + x_2 + x_3 \leq 1 \\ & 1 \geq x_1, x_2, x_3 \geq 0 \end{array}$$

with obvious optimal solution $\tilde{\mathbf{x}} = (1, 0, 0)$. Since this solution is also feasible in the original model, it follows from principle 12.10 that it is optimal there.

(b) The linear programming relaxation of this model is

$$\begin{array}{ll} \max & x_1 + x_2 + x_3 \\ \text{s.t.} & x_1 + x_2 \leq 1 \\ & x_2 + x_3 \leq 1 \\ & x_1 + x_3 \leq 1 \\ & 1 \geq x_1, x_2, x_3 \geq 0 \end{array}$$

with optimal solution $\tilde{\mathbf{x}} = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$. Since this solution violates integrality requirements in the original model, it is infeasible there. It could not be optimal.

(c) The indicated relaxation is

$$\begin{array}{ll} \min & 2x_1 + 4x_2 + 8x_3 \\ \text{s.t.} & 10x_1 + 3x_2 + x_3 \geq 8 \\ & x_1, x_2, x_3 = 0 \text{ or } 1 \end{array}$$

with obvious optimal solution $\tilde{\mathbf{x}} = (1, 0, 0)$. This relaxation optimum satisfies relaxed constraint

$$x_1 + x_2 + x_3 \leq 2$$

and so is feasible in the original model. It follows from principle 12.10 that it is optimal in the full integer program.

Rounded Solutions from Relaxations

When principle 12.10 applies, relaxation completely solves a hard discrete optimization model. More commonly, things are not that simple. As with the EMS solution (12.4) above, relaxation optima usually violate some constraints of the true model.

All is hardly lost. First, we have the bound of principle 12.9. We may also have a starting point for constructing a good heuristic solution to the full discrete model.

12.11 Many relaxations produce optimal solutions that are easily "rounded" to good feasible solutions for the full model.

Consider, for example, the EMS solution (12.4). The nature of model constraints (12.3), $\geq$ form with nonnegative coefficients on the left-hand side, means that feasibility of a solution is not lost if we increase some of its components. Beginning from the LP relaxation optimum and rounding up produces the approximate optimal solution

$$\begin{aligned} \hat{x}_1 &= \lceil \tilde{x}_1 \rceil = \lceil 0 \rceil = 0 \\ \hat{x}_2 &= \lceil \tilde{x}_2 \rceil = \lceil 1 \rceil = 1 \\ \hat{x}_3 &= \lceil \tilde{x}_3 \rceil = \lceil 1 \rceil = 1 \\ \hat{x}_4 &= \lceil \tilde{x}_4 \rceil = \lceil \tfrac{1}{2} \rceil = 1 \\ \hat{x}_5 &= \lceil \tilde{x}_7 \rceil = \lceil \tfrac{1}{2} \rceil = 1 \\ \hat{x}_6 &= \lceil \tilde{x}_6 \rceil = \lceil \tfrac{1}{2} \rceil = 1 \\ \hat{x}_7 &= \lceil \tilde{x}_7 \rceil = \lceil 0 \rceil = 0 \\ \hat{x}_8 &= \lceil \tilde{x}_8 \rceil = \lceil 1 \rceil = 1 \\ \hat{x}_9 &= \lceil \tilde{x}_9 \rceil = \lceil \tfrac{1}{2} \rceil = 1 \\ \hat{x}_{10} &= \lceil \tilde{x}_{10} \rceil = \lceil 1 \rceil = 1 \end{aligned} \tag{12.5}$$

with value $\sum_{j=1}^{10} \hat{x}_j = 8$. Here **ceiling** notation

$$\lceil x \rceil \triangleq \text{least integer greater than or equal to } x$$

The corresponding **floor** notation

$$\lfloor x \rfloor \triangleq \text{greatest integer less that or equal to } x$$

Heuristic optimum $\hat{\mathbf{x}}$ may not be truly optimal, but it does satisfy all constraints. Where time permits no deeper analysis, this rounded relaxation solution might well suffice. Also, feasible solutions provide bounds to complement those obtained from the optimal relaxation solution value (principle 12.9).

> 12.12 The objective function value of any (integer) feasible solution to a maximizing discrete optimization problem provides a lower bound on the integer optimal value, and any (integer) feasible solution to a minimizing discrete optimization problem provides an upper bound.

Set covering relaxation optima like (12.4) are particularly easy to round, because of the unusually simple form of the constraints. Many other forms admit similar rounding. Some round infeasible relaxation solutions up, some round down, and some do other straightforward patching. Details vary with model form.

Unfortunately, there are some discrete models that just do not round. For an example, return to our AA airline crew scheduling model (11.10) (Section 11.3). Its set partitioning form closely resembles the set covering case we just rounded easily. But set partitioning involves equality constraints. Each time we round some infeasible $\tilde{x}_j$ up to 1 or down to 0, other variables sharing constraints with that x_j will also have to be adjusted if feasibility is to be preserved. Much more complex rounding schemes are required, and success cannot be guaranteed.

Sample Exercise 12.8: Rounding Relaxation Optima

In each of the following integer linear programs, develop and apply a scheme for rounding the indicated LP relaxation optimum to an approximate solution for the full model. Also, indicate the best lower and upper bounds on the optimal integer solution value available from relaxation and rounding.

(a) $$\begin{aligned} \min\quad & 10x_1 + 8x_2 + 18x_3 \quad \text{with LP relaxation optimum } \tilde{\mathbf{x}} = (0, 1, \tfrac{1}{7}) \\ \text{s.t.}\quad & 2x_1 + 4x_2 + 7x_3 \geq 5 \\ & x_1 + x_2 + x_3 \geq 1 \\ & x_1, x_2, x_3 = 0 \text{ or } 1 \end{aligned}$$

(b) $$\begin{aligned} \max\quad & 40x_1 + 2x_2 + 18x_3 \quad \text{with LP relaxation optimum } \tilde{\mathbf{x}} = (1, 0, \tfrac{3}{7}) \\ \text{s.t.}\quad & 2x_1 + 11x_2 + 7x_3 \leq 5 \\ & x_1 + x_2 + x_3 \leq 2 \\ & x_1, x_2, x_3 = 0 \text{ or } 1 \end{aligned}$$

(c) $$\begin{aligned} \min\quad & 3x_1 + 5x_2 + 20x_3 + 14x_4 \quad \text{with LP relaxation optimum } \tilde{\mathbf{x}} = (\tfrac{16}{3}, \tfrac{17}{3}, \tfrac{16}{33}, \tfrac{17}{33}) \\ \text{s.t.}\quad & x_1 + x_2 = 11 \\ & 3x_1 + 6x_2 = 50 \\ & x_1 \leq 11x_3 \\ & x_2 \leq 11x_4 \\ & x_1, x_2 \geq 0 \\ & x_3, x_4 = 0 \text{ or } 1 \end{aligned}$$

Analysis:

(a) All main constraints of this model are $\geq$ form, and coefficients on the left-hand side are nonnegative. Thus increasing feasible variable values cannot cause a violation. We may round up to integer-feasible solution

$$\lceil \tilde{\mathbf{x}} \rceil = (\lceil 0 \rceil, \lceil 1 \rceil, \lceil \tfrac{1}{7} \rceil) = (0, 1, 1)$$

Substituting this solution in the objective function gives an upper bound (principle 12.12) of 26 on the optimal value. The corresponding lower bound, which is obtained by substituting the relaxation optimal solution (principle 12.9), is 10.57.

(b) All main constraints of this model are $\leq$ form, and coefficients on the left-hand side are nonnegative. Thus decreasing feasible variable values cannot cause a violation. We may round down to integer-feasible solution

$$\lfloor \tilde{\mathbf{x}} \rfloor = (\lfloor 1 \rfloor, \lfloor 0 \rfloor, \lfloor \tfrac{3}{7} \rfloor) = (1, 0, 0)$$

Substituting this solution in the objective function gives a lower bound (principle 12.12) of 40 on the optimal value. The corresponding upper bound, which is obtained by substituting the relaxation optimal solution (principle 12.9), is 47.71.

(c) Each of the discrete variables in this mixed-integer linear program occurs in only one $\leq$ constraint on the right-hand side. Thus increasing x_3 and x_4 from their relaxation values cannot lose feasiblity. We may round up to

$$(\tfrac{16}{3}, \tfrac{17}{3}, \lceil \tfrac{16}{33} \rceil, \lceil \tfrac{17}{33} \rceil) = (\tfrac{16}{3}, \tfrac{17}{3}, 1, 1)$$

Notice that continuous variable values were not changed.

Substituting this solution in the objective function gives an upper bound (principle 12.12) of 78.33 on the optimal value. The corresponding lower bound, which is obtained by substituting the relaxation optimal solution (principle 12.9), is 61.24.

12.3 Stronger LP Relaxations, Valid Inequalities, and Lagrangian Relaxations

It should be obvious that we will detect infeasibility quicker (principle 12.8), obtain sharper bounds (principle 12.9), have a better chance of discovering an optimal solution (principle 12.10), and find rounding much easier (principle 12.11) if the relaxations we employ closely approximate the full model of interest. Strong relaxations do just that.

12.13 A relaxation is **strong** or **sharp** if its optimal value closely bounds that of the true model, and its optimal solution closely approximates an optimum in the full model.

Analysts dealing with hard discrete models via relaxations will almost always find it worthwhile to look for means to strengthen the relaxations without losing too much tractability.

Stronger LP Relaxations

To begin, let us will focus on the standard linear programming relaxations of ILP models. How can we make such LP relaxations strong? The key insight sometimes surprises:

12.14 Equally correct integer linear programming formulations of a discrete problem may have dramatically different linear programming relaxation optima.

SAMPLE EXERCISE 12.9: UNDERSTANDING STRONGER LP RELAXATIONS

Show (by inspection) that even though the two following integer linear programming models have the same feasible solutions, the second yields a stronger linear programming relaxation.

$$\begin{array}{ll} \max & x_1+x_2+x_3 \\ \text{s.t.} & x_1+x_2 \le 1 \\ & x_1+x_3 \le 1 \\ & x_2+x_3 \le 1 \\ & x_1,x_2,x_3 = 0 \text{ or } 1 \end{array} \qquad \begin{array}{ll} \max & x_1+x_2+x_3 \\ \text{s.t.} & x_1+x_2 \le 1 \\ & x_1+x_3 \le 1 \\ & x_2+x_3 \le 1 \\ & x_1+x_2+x_3 \le 1 \\ & x_1,x_2,x_3 = 0 \text{ or } 1 \end{array}$$

Analysis: Both ILPs have the same feasible solutions,

$$\mathbf{x}^{(1)} = (1,0,0)$$
$$\mathbf{x}^{(2)} = (0,1,0)$$
$$\mathbf{x}^{(3)} = (0,0,1)$$

Thus they are both valid models of the same problem. Still, the first has LP relaxation optimum $\tilde{\mathbf{x}} = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$, and the second has relaxation optimum $\tilde{\mathbf{x}} = (1,0,0)$ (among others). The corresponding relaxation bounds are $\frac{3}{2}$ and 1, making the second relaxation stronger. In this simple case, in fact, it yields a discrete optimum (via principle 12.10).

Choosing Big-M Constants

The "sufficiently large" big-M constants needed in so many models offer one easy family where details of ILP modeling affect the LP relaxation. Return, for instance, to the tiny Bison Boosters model of (12.2) and Table 12.2. In formulating switching constraints $x_1 \le 400y_1$ and $x_2 \le 75y_2$, we constructed values 400 and 75 with a back-of-envelope computation. Any sufficiently large M would yield a correct integer linear programming model.

Suppose that we had used 10,000 for both. The new model is

$$\begin{array}{lll} \max & 20x_1+30x_2-550y_1-720y_2 & \text{(net income)} \\ \text{s.t.} & 1.5x_1+4x_2 \le 300 & \text{(display space)} \\ & x_1 \le 10,000y_1 & \text{(T-shirts if equipment)} \\ & x_2 \le 10,000y_2 & \text{(sweatshirts if equipment)} \\ & x_1, x_2 \ge 0 & \\ & y_1, y_2 = 0 \text{ or } 1 & \end{array} \quad (12.6)$$

Recall that the original model (12.2) had relaxation optimum

$$\tilde{x}_1 = 200, \quad \tilde{x}_2 = 0, \quad \tilde{y}_1 = 1, \quad \tilde{y}_2 = 0$$

matching perfectly the discrete optimal solution with value \$3450. Its LP relaxation was indeed strong.

Revision (12.6) is every bit as correct as the original (12.2) in the sense that it has exactly the same (discrete) feasible set. However, the LP relaxation of (12.6) yields optimum

$$\tilde{x}_1 = 200, \quad \tilde{x}_2 = 0, \quad \tilde{y}_1 = 0.02, \quad \tilde{y}_2 = 0 \qquad (12.7)$$

with value \$3989. The value bound \$3989 now differs significantly from the true optimal value \$3450. Also, the relaxation optimal solution has component $\tilde{y}_1$ at a tiny fractional value. With only (12.7) at hand, it would be hard to tell whether to rent or not the T-shirt equipment.

This contrast between LP relaxations of integer-equivalent models (12.2) and (12.6) highlights an important and easy-to-implement principle for strengthening relaxations:

12.15 Whenever a discrete model requires sufficiently large big-M's, the strongest relaxations will result from models employing the smallest valid choice of those constants.

SAMPLE EXERCISE 12.10: CHOOSING SMALLEST BIG-M'S

We wish to decide which combination of 2 pharmaceutical facilities should be used to produce 80 units of a needed product. One costs \$5000 to setup and has variable cost \$20 unit. The other cost \$7000 to setup and has variable cost \$15. Both have capacity of 200 units.

(a) Formulate a mixed-integer linear programming model using capacities for needed big-M's.

(b) Strengthen the linear programming relaxation of your model in part (a) by reducing big-M's to their smallest valid value.

Modeling:

(a) Using decision variables x_1 and x_2 for the amount produced in each facility, and switching variables x_3 and x_4 to track setups, a valid formulation is

$$\begin{aligned} \min \quad & 20x_1 + 15x_2 + 5000x_3 + 7000x_4 \\ \text{s.t.} \quad & x_1 + x_2 = 80 \\ & x_1 \le 200x_3 \\ & x_2 \le 200x_4 \\ & x_1, x_2 \ge 0 \\ & x_3, x_4 = 0 \text{ or } 1 \end{aligned}$$

Full capacity is available whenever setup cost is paid.

(b) Although capacities are 200, the problem calls for only 80 units to be produced. Thus neither x_1 nor x_2 will ever exceed 80 in an optimal solution. We may strengthen the model by reducing big-M constants from 200 to 80, to produce

$$\begin{aligned} \min \quad & 20x_1 + 15x_2 + 5000x_3 + 7000x_4 \\ \text{s.t.} \quad & x_1 + x_2 = 80 \\ & x_1 \le 80x_3 \\ & x_2 \le 80x_4 \\ & x_1, x_2 \ge 0 \\ & x_3, x_4 = 0 \text{ or } 1 \end{aligned}$$

The reader can verify that this new formulation has relaxation optimum $\tilde{\mathbf{x}} = (80, 0, 1, 0)$ with value \$6600, versus the original model's $\tilde{\mathbf{x}} = (80, 0, .4, 0)$ at value \$3600.

Valid Inequalities

Sharpening big-M coefficients is only one of many ways to strengthen LP relaxations. We can also add new valid inequality constraints.

> 12.16 A linear inequality is a **valid inequality** for a given discrete optimization model if it holds for all (integer) feasible solutions to the model.

Relaxations can often be strenghtened dramatically by including valid inequalities that are not needed for a correct discrete model.

Not every valid inequality strengthens a relaxation. For example, all inequality constraints of the original formulation are trivially valid because they are satisfied by every feasible solution.

> 12.17 To strengthen a relaxation, a valid inequality must cut off (render infeasible) some feasible solutions to the current LP relaxation that are not feasible in the full ILP model.

This need to cut off noninteger relaxation solutions is why valid inequalities are sometimes called **cutting planes**.

The Tmark facilities location model of Section 11.6 illustrates a classic case. The model formulated there is:

$$\begin{aligned} \min \quad & \sum_{i=1}^{8}\sum_{j=1}^{14} (d_j r_{i,j}) x_{i,j} + \sum_{i=1}^{8} f_i y_i && && \text{(total fixed cost)} \\ \text{s.t.} \quad & \sum_{i=1}^{8} x_{i,j} = 1 && \text{for all } j = 1, \ldots, 14 && \text{(carry } j \text{ load)} \\ & 1500 y_i \le \sum_{j=1}^{14} d_j x_{i,j} && \text{for all } i = 1, \ldots, 8 && \text{(minimum at } i\text{)} \\ & \sum_{j=1}^{14} d_j x_{i,j} \le 5000 y_i && \text{for all } i = 1, \ldots, 8 && \text{(maximum at } i\text{)} \\ & x_{i,j} \ge 0 && \text{for all } i = 1, \ldots, 8; \; j = 1, \ldots, 14 \\ & y_i = 0 \text{ or } 1 && \text{for all } i = 1, \ldots, 8 \end{aligned} \tag{12.8}$$

where $x_{i,j}$ if the fraction of region j's call traffic handled by center i, y_i decides whether or not center i is opened, d_j is the anticipated call demand from region j, $r_{i,j}$ is the unit cost of calls from region j to center i, and f_i is the fixed cost of opening center i.

Focus on the third, maximum capacity set of constraints. Each forces discrete variable y_i to take on a value in the relaxation satisfying

$$y_i \ge \frac{\sum_{j=1}^{14} d_j x_{i,j}}{5000} \triangleq \frac{\text{capacity used}}{\text{total available}}$$

For discrete modeling, these constraints do fine. Each y_i must equal 1 if corresponding x-variables are to use facility i at any level. In the LP relaxation, however, if x-variables use only a small part of the capacity, the corresponding y_i will take on a small fractional value.

The numerical values of Section 11.6 confirm this behavior. The LP relaxation of formulation (12.8) has

$$\begin{aligned} &\tilde{y}_1 = 0.230, \quad \tilde{y}_2 = 0.000, \quad \tilde{y}_3 = 0.000, \quad \tilde{y}_4 = 0.301 \\ &\tilde{y}_5 = 0.115, \quad \tilde{y}_6 = 0.000, \quad \tilde{y}_7 = 0.000, \quad \tilde{y}_8 = 0.650 \\ &\text{total cost} = \$8036.60 \end{aligned} \tag{12.9}$$

with many of the $\tilde{y}_j$ small.

Compare the optimal mixed-integer solution

$$\begin{aligned} &y_1^* = 0, \quad y_2^* = 0, \quad y_3^* = 0, \quad y_4^* = 1 \\ &y_5^* = 0, \quad y_6^* = 0, \quad y_7^* = 0, \quad y_8^* = 1 \\ &\text{total cost} = \$10{,}153 \end{aligned} \tag{12.10}$$

Bound \$8036 of (12.9) is only 79% of true optimal value \$10,153. Also, (12.9) suggests that 4 centers may be needed, while the optimum opens only 2.

Even when a center is used only fractionally, it may fulfill the whole demand for some single district. Such thinking suggests valid inequalities

$$x_{i,j} \le y_i \quad \text{for all } i = 1, \ldots, 8; \; j = 1, \ldots, 14 \tag{12.11}$$

which require that the fraction a center is opened be as great as the fraction of any region's demand satisfied from the center.

Certainly, these inequalities satisfy validity definition 12.16 because each is satisfied by every integer-feasible solution to model (12.8). Also, it is easy to fulfill requirement 12.17 by finding solutions to the relaxation of (12.8) that violate (12.11).

Adding these valid inequalities improves the LP relaxation dramatically. The strengthened model has optimal solution

$$\begin{array}{llll} \tilde{y}_1 = 0.000, & \tilde{y}_2 = 0.000, & \tilde{y}_3 = 0.000, & \tilde{y}_4 = 0.537 \\ \tilde{y}_5 = 0.000, & \tilde{y}_6 = 0.000, & \tilde{y}_7 = 0.000, & \tilde{y}_8 = 1.000 \\ \text{total cost} = \$10{,}033.68 & & & \end{array}$$

Its bound \$10,033 is almost 99% of optimal value \$10,153, and only one discrete variable comes out fractional. Addition of the valid inequalities (12.11) has produced a much stronger relaxation, which provides much better information about the form of a discrete optimum.

SAMPLE EXERCISE 12.11: RECOGNIZING USEFUL VALID INEQUALITIES

Consider the ILP

$$\begin{array}{ll} \max & 3x_1 + 14x_2 + 18x_3 \\ \text{s.t.} & 3x_1 + 5x_2 + 6x_3 \le 10 \\ & x_1, x_2, x_3 = 0 \text{ or } 1 \end{array}$$

with LP relaxation optimum $\tilde{\mathbf{x}} = (0, \frac{4}{5}, 1)$. Determine (by inspection) whether each of the following inequalities is valid for this model, and if so, whether adding it would strengthen the LP relaxation.

(a) $x_2 + x_3 \le 1$

(b) $x_1 + x_2 + x_3 \le 1$

(c) $3x_1 + 5x_2 \le 10$

Analysis: We apply definition 12.16 and principle 12.17.

(a) It is obvious from the main constraint that no feasible solution can have both $x_2 = 1$ and $x_3 = 1$. Thus the constraint is valid. Also, the current LP relaxation optimum is one LP-feasible solution that violates the inequality because

$$\tilde{x}_2 + \tilde{x}_3 = \tfrac{4}{5} + 1 \not\le 1$$

It follows that the constraint will strengthen the relaxation.

(b) This constraint is not valid. For example, $\mathbf{x} = (1, 0, 1)$ violates the constraint even though it is integer-feasible in the given model.

(c) This constraint is valid, because any integer-feasible solution satisfying main constraint $3x_1 + 5x_2 + 6x_3 \le 10$ certainly has $3x_1 + 5x_2 \le 10$. Still, this will also be true of all feasible solutions in the LP relaxation. Adding the inequality cannot improve the relaxation.

Lagrangian Relaxations

Even when the given model is an ILP, the strongest practical relaxation may not be the LP form obtained when integrality constraints are dropped. **Lagrangian relaxations**, which prove stronger for some model forms, adopt a completely different strategy. Instead of dropping integrality requirements, they relax some of the main linear constraints of the model. However, the relaxed constraints are not totally dropped. Instead, they are **dualized** or weighted in the objective function with suitable **Lagrange multipliers** to discourage violations.

12.18 **Lagrangian relaxations** partially relax some of the main linear constraints of an ILP by moving them to the objective function as terms

$$\cdots + v_i \left(b_i - \sum_j a_{i,j} x_j \right) \cdots$$

Here v_i is a Lagrange multiplier chosen as the relaxation is formed. If the relaxed constraint has form $\sum_j a_{i,j} x_j \ge b_i$, multiplier $v_i \le 0$ for a maximize model and $v_i \ge 0$ in a minimize. If the relaxed constraint is $\sum_j a_{i,j} x_j \le b_i$, multiplier $v_i \ge 0$ for a maximize model and $v_i \le 0$ for a minimize model. Equality constraints $\sum_j a_{i,j} x_j = b_i$ have unrestricted multipliers v_i.

Lagrangian Relaxation of the CDOT Example

We can illustrate with the CDOT generalized assignment model (11.13) of Section 11.4:

$$\begin{array}{llll} \min & 130x_{1,1} + 460x_{1,2} + 40x_{1,3} + 30x_{2,1} + 150x_{2,2} + 370x_{2,3} & & \\ & + 510x_{3,1} + 20x_{3,2} + 120x_{3,3} + 30x_{4,1} + 40x_{4,2} + 390x_{4,3} & & \\ & + 340x_{5,1} + 30x_{5,2} + 40x_{5,2} + 20x_{6,1} + 450x_{6,2} + 30x_{6,3} & & \\ \text{s.t.} & x_{1,1} + x_{1,2} + x_{1,3} = 1 & & \text{(district 1)} \\ & x_{2,1} + x_{2,2} + x_{2,3} = 1 & & \text{(district 2)} \\ & x_{3,1} + x_{3,2} + x_{3,3} = 1 & & \text{(district 3)} \\ & x_{4,1} + x_{4,2} + x_{4,3} = 1 & & \text{(district 4)} \\ & x_{5,1} + x_{5,2} + x_{5,3} = 1 & & \text{(district 5)} \\ & x_{6,1} + x_{6,2} + x_{6,3} = 1 & & \text{(district 6)} \\ & 30x_{1,1} + 50x_{2,1} + 10x_{3,1} & & \text{Estevan} \\ & \quad + 11x_{4,1} + 13x_{5,1} + 9x_{6,1} & \le 50 & \\ & 10x_{1,2} + 20x_{2,2} + 60x_{3,2} & & \text{(Mackenzie)} \\ & \quad + 10x_{4,2} + 10x_{5,2} + 17x_{6,2} & \le 50 & \\ & 70x_{1,3} + 10x_{2,3} + 10x_{3,3} & & \text{(Skidegate)} \\ & \quad + 15x_{4,3} + 8x_{5,3} + 12x_{6,3} & \le 50 & \\ & x_{i,j} = 0 \text{ or } 1 \quad i = 1, 6; j = 1, 3 & & \end{array} \quad (12.12)$$

where

$$x_{i,j} \triangleq \begin{cases} 1 & \text{if district } i \text{ is assigned to ship } j \\ 0 & \text{otherwise} \end{cases}$$

One strong Lagrangian relaxation dualizes the first 6 main constraints with weights

$$v_i \triangleq \text{the Lagrange multiplier on the constraint for district } i$$

The result is

$$\begin{aligned} \min \quad & 130x_{1,1} + 460x_{1,2} + 40x_{1,3} + 30x_{2,1} + 150x_{2,2} + 370x_{2,3} \\ & + 510x_{3,1} + 20x_{3,2} + 120x_{3,3} + 30x_{4,1} + 40x_{4,2} + 390x_{4,3} \\ & + 340x_{5,1} + 30x_{5,2} + 40x_{5,2} + 20x_{6,1} + 450x_{6,2} + 30x_{6,3} \\ & + v_1(1 - x_{1,1} - x_{1,2} - x_{1,3}) + v_2(1 - x_{2,1} - x_{2,2} - x_{2,3}) \\ & + v_3(1 - x_{3,1} - x_{3,2} - x_{3,3}) + v_4(1 - x_{4,1} - x_{4,2} - x_{4,3}) \\ & + v_5(1 - x_{5,1} - x_{5,2} - x_{5,3}) + v_6(1 - x_{6,1} - x_{6,2} - x_{6,3}) \\ \text{s.t.} \quad & 30x_{1,1} + 50x_{2,1} + 10x_{3,1} + 11x_{4,1} + 13x_{5,1} + 9x_{6,1} \le 50 \\ & 10x_{1,2} + 20x_{2,2} + 60x_{3,2} + 10x_{4,2} + 10x_{5,2} + 17x_{6,2} \le 50 \\ & 70x_{1,3} + 10x_{2,3} + 10x_{3,3} + 15x_{4,3} + 8x_{5,3} + 12x_{6,3} \le 50 \\ & x_{i,j} = 0 \text{ or } 1 \qquad i = 1, 6;\ j = 1, 3 \end{aligned} \tag{12.13}$$

Notice that the 6 equality constraints of full model (12.12) have not been completely dropped. Instead, they have been rolled into the objective function as in construction 12.18. For example, the model (12.13) objective now includes the term

$$\cdots + v_3(1 - x_{3,1} - x_{3,2} - x_{3,3}) \cdots \tag{12.14}$$

Feasible solutions in this Lagrangian relaxation may very well have

$$x_{3,1} + x_{3,2} + x_{3,3} \neq 1 \quad \text{or} \quad (1 - x_{3,1} - x_{3,2} - x_{3,3}) \neq 0$$

But if weight $v_3 \neq 0$, violations will at least affect the objective value through (12.14).

SAMPLE EXERCISE 12.12: FORMING LAGRANGIAN RELAXATIONS

Return to Bison Boosters example model (12.2):

$$\begin{aligned} \max \quad & 20x_1 + 30x_2 - 550y_1 - 720y_2 \\ \text{s.t.} \quad & 1.5x_1 + 4x_2 \le 300 \\ & x_1 - 200y_1 \le 0 \\ & x_2 - 75y_2 \le 0 \\ & x_1, x_2 \ge 0 \\ & y_1, y_2 = 0 \text{ or } 1 \end{aligned}$$

(a) Use multipliers v_1 and v_2 to form a Lagrangian relaxation dualizing the last two, switching constraints.

(b) Indicate any required sign restrictions on multipliers v_1 and v_2.

Analysis: We apply construction 12.18.

(a) The Lagrangian relaxation is formed by moving the two switching constraints to the objective function as

$$\begin{aligned} \max \quad & 20x_1 + 30x_2 - 550y_1 - 720y_2 + v_1(0 - x_1 + 200y_1) + v_2(0 - x_2 + 75y_2) \\ \text{s.t.} \quad & 1.5x_1 + 4x_2 \le 300 \\ & x_1, x_2 \ge 0 \\ & y_1, y_2 = 0 \text{ or } 1 \end{aligned}$$

(b) For $\le$ constraints in a maximize model, multipliers should satisfy $v_1, v_2 \ge 0$.

Tractable (Integer) Lagrangian Relaxations

Notice that the Lagrangian relaxation (12.13) keeps variables $x_{i,j}$ discrete. Integrality requirements of the original (12.12) have not been dropped.

Lagrangian relaxations fulfill principle 12.5 's mandate for increased tractability by dualizing enough linear constraints that the remaining discrete problem is manageable.

> 12.19 Constraints chosen for dualization in Lagrangian relaxations should leave a still-integer linear program with enough special structure to be relatively tractable.

A close look at the remaining constraints in Lagrangian relaxation (12.13) will reveal how it conforms to requirement 12.19. After dualization, each $x_{i,j}$ occurs in exactly one main constraint. Thus these relaxations can be solved as a series of single-constraint, knapsack ILPs (definition 11.4, Section 11.2), which are the simplest of integer programs. There is one for each of the 3 ships.

Lagrangian Relaxation Bounds

Dropping linear constraints in a Lagrangian relaxation 12.18 cannot eliminate any solutions. That is, Lagrangian relaxations parallel property 12.4 in having every solution feasible in the full model still feasible in the relaxation. But Lagrangian forms are more complex than constraint relaxations because they modify both constraints and objective. Fortunately, they still yield bounds.

> 12.20 The optimal value of any Lagrangian relaxation of a maximize model using multipliers conforming to 12.18 yields an upper bound on the optimal value of the full model. The optimal value of any valid Lagrangian relaxation of a minimize model yields a lower bound.

Sign rules of definition 12.18 assure that every feasible solution in the original ILP achieves no less in the objective function of the Lagrangian relaxation when we maximize and no more when we minimize. Thus the relaxation optimum, which either equals or improves on all these results for truly feasible solutions, must yield a bound.

Choosing Lagrange Multipliers

How strong the bounds of 12.20 prove to be in any Lagrangian relaxation depends on the multiplier values chosen. Some choices of v_i will produce a very weak Lagrangian relaxation, and others can make it quite strong.

> 12.21 A search is usually required to identify Lagrange multiplier values v_i defining a strong Lagrangian relaxation.

Methods for determining good Lagrange multipliers are beyond the scope of this book, but we can illustrate their potential power with

$$v_1 = 300, \quad v_2 = 200, \quad v_3 = 200, \quad v_4 = 45, \quad v_5 = 45, \quad v_6 = 30$$

in our CDOT example. The corresponding optimal value in Lagrangian relaxation (12.13) is \$470,000, which is much closer to the integer optimal value of \$480,000 than is the LP relaxation bound \$326,100.

SAMPLE EXERCISE 12.13: UNDERSTANDING LAGRANGE MULTIPLIER IMPACTS

Return to the Bison Boosters Lagrangian relaxation of Sample Exercise 12.12. Solve (by inspection) the relaxation for each of the following choices of Lagrange multipliers, and comment on the strength of the results.

(a) $v_1 = 0, v_2 = 1$

(b) $v_1 = 3, v_2 = 3$

Analysis:

(a) For $v_1 = 0, v_2 = 1$, the Lagrangian relaxation reduces to

$$\begin{aligned} \max \quad & 20x_1 + 29x_2 - 550y_1 - 645y_2 \\ \text{s.t.} \quad & 1.5x_1 + 4x_2 \le 300 \\ & x_1, x_2 \ge 0 \\ & y_1, y_2 = 0 \text{ or } 1 \end{aligned}$$

With only 0–1 constraints enforced on the y_j, setting $\tilde{y}_1 = \tilde{y}_2 = 0$ produces a relaxation optimum. Corresponding optimal choices for the continuous variables are $\tilde{x}_1 = 200$ and $\tilde{x}_2 = 0$. The resulting relaxation bound of \$4000 is weak because the mixed-integer optimal value is \$3450.

(b) For $v_1 = v_2 = 3$, the Lagrangian relaxation is

$$\begin{aligned} \max \quad & 17x_1 + 27x_2 + 50y_1 - 495y_2 \\ \text{s.t.} \quad & 1.5x_1 + 4x_2 \le 300 \\ & x_1, x_2 \ge 0 \\ & y_1, y_2 = 0 \text{ or } 1 \end{aligned}$$

With only 0–1 constraints enforced on the y_j, setting $\tilde{y}_1 = 1, \tilde{y}_2 = 0$ produces a relaxation optimum. Corresponding optimal choices for the continuous variables are again $\tilde{x}_1 = 200, \tilde{x}_2 = 0$. This time, the resulting relaxation bound of \$3450 is very strong because it exactly matches the true mixed-integer value.

12.4 BRANCH AND BOUND SEARCH

Total enumerations of Section 12.1 are impractical for all but the simplest models because every one of an explosively growing number of discrete solutions must be considered explicitly. The process would become much more manageable if we could deal with those solutions in large classes, determining for each whole class whether it is likely to contain optimal solutions, and doing so without explicit enumeration of all its members. Only the most promising classes would have to be searched in detail.

Branch and bound algorithms combine such a partial or subset enumeration strategy with the relaxations of Sections 12.2 and 12.3. They systematically form classes of solutions and investigate whether the classes can contain optimal solutions by analyzing associated relaxations. More detailed enumeration ensues only if the relaxations fail to be definitive.

EXAMPLE 12.2: RIVER POWER

As with so many other topics, an artificially small example will aid in our development of branch and bound ideas. Here we consider an operations problems at River Power Company.

River Power has 4 generators currently available for production and wishes to decide which to put on line to meet the expected 700-megawatt peak demand over the next several hours. The following table shows the cost to operate each generator (in thousands of dollars per hour) and their outputs (in megawatts).

	Generator, *j*			
	1	**2**	**3**	**4**
Operating cost	7	12	5	14
Ouput power	300	600	500	1600

Units must be completely on or completely off.

We can formulate River Power's problem as a knapsack problem like those of Section 11.2. Decision variables

$$x_j \triangleq \begin{cases} 1 & \text{if generator } j \text{ is turned on} \\ 0 & \text{otherwise} \end{cases}$$

Then a model is

$$\begin{aligned} \min \quad & 7x_1 + 12x_2 + 5x_3 + 14x_4 && \text{(total cost)} \\ \text{s.t.} \quad & 300x_1 + 600x_2 + 500x_3 + 1600x_4 \ge 700 && \text{(demand)} \\ & x_1, x_2, x_3, x_4 = 0 \text{ or } 1 \end{aligned} \tag{12.15}$$

The objective function minimizes total operating costs, and the main constraint assures that the chosen combination of generators will fulfill demand. Total enumeration establishes that an optimal solution use generators 1 and 3 and cost \$12,000.

Partial Solutions

Much like the improving searches of most of this book, branch and bound searches iterate through a sequence of solutions until we are ready to conclude optimality or

Proof: For any u, by definition of L,

$$\begin{aligned} L(u) &\leq c^\top \overline{x} + u^\top (b - A\overline{x}) \\ &= L(\overline{u}) - \overline{u}^\top (b - A\overline{x}) + u^\top (b - A\overline{x}) \\ &= L(\overline{u}) + (u - \overline{u})^\top (b - A\overline{x}) \end{aligned}$$

Let $\overline{\gamma} = (b - A\overline{x})$ and this completes the proof. □

Corollary 12.17 *The dual objective $L(u)$ is differentiable at a point $\overline{u}$ if and only if $b_i - (a^i)^\top x$ is constant over $\Gamma(\overline{u})$ for all i. In this case, the gradient of $L(\overline{u})$ is given by $(b - A\overline{x})$ for all $\overline{x} \in \Gamma(\overline{u})$.*

The subgradient extends the concept of a gradient to nondifferentiable concave/convex functions. We have seen the importance of the dual problem and some of its properties. Now let's see how we can optimize the dual function.

12.5.1 Method 1: Subgradient Optimization

Subgradient optimization extends the method of steepest ascent to the dual function which is concave and piecewise linear (see Proposition 12.10). See a nonlinear programming text such as Bazaraa, Sherali, and [44] for a treatment of gradients and steepest ascent algorithms. Unfortunately, moving in the direction of a subgradient does not necessarily lead to an improvement in the dual function. It is left as an exercise to construct an example problem where moving in the direction of a subgradient does not necessarily lead to an improvement in the dual objective function value. However, there is a rational for moving in the direction of a subgradient.

Assume $\overline{u}$ is an optimal solution to the dual problem $\max_{u \geq 0} L(u)$. If $u^j \geq 0$ and $\gamma^j = g(x^j)$ where $x^j \in \Gamma(u^j)$ it follows from the definition of subgradient that

$$L(\overline{u}) \leq L(u^j) + (\overline{u} - u^j)^\top \gamma^j$$

which implies

$$L(\overline{u}) - L(u^j) \leq (\overline{u} - u^j)^\top \gamma^j$$

But $\overline{u}$ is an optimal dual solution so $L(\overline{u}) - L(u^j) \geq 0$ which implies $(\overline{u} - u^j)^\top \gamma^j \geq 0$. Then γ^j makes an acute angle with the ray from u^j through $\overline{u}$. Let $u^{j+1} = \max\{0, u^j + t_j \gamma^j\}$ where t_j is a positive scalar called the step size. Consequently, if t_j is sufficiently small, the point u^{j+1} will be closer to $\overline{u}$ than u^j. Thus, the sequence of dual solutions generated by the subgradient algorithm approaches an optimal solution, although the value of $L(u^{j+1})$ is not necessarily greater than $L(u^j)$.

Algorithm 12.18 (Subgradient Optimization Algorithm)

Step 1: (Initialization) *Let $j \leftarrow 0$, $u^j \in (R^m)^+$ and $\epsilon > 0$.*

Step 2: *$\gamma^j \leftarrow g(x^j)$ where $x^j \in \Gamma(u^j)$.*

Step 3: *Let $u^{j+1} \leftarrow \max\{0, u^j + t_j \gamma^j\}$ where t_j is a positive scalar called the step size.*

Step 4: *If $\|u^{j+1} - u^j\| < \epsilon$ stop else let $j \leftarrow j + 1$ and go to* **Step 2**.

The question that remains is how to select the t_j in order to guarantee convergence. The fundamental theoretical result (see Held, Wolfe,and Crowder [226] and references therein) is that the sequence $\{L(u^j)\}$ converges to $L(\overline{u})$ if the sequence $\{t_j\}$ converges to zero and $\sum_{j=0}^{\infty} t_j = \infty$. The reason for the condition $\sum_{j=0}^{\infty} t_j = \infty$ is to make the step sizes "large enough" to get from an initial u^0 to an optimal $\overline{u}$. However, we want the $t_j \rightarrow 0$ so we can't continually "overshoot" $\overline{u}$. It is common to determine the t_j by a formula such as

$$t_j = \frac{\theta_j (L^* - L(u^j))}{\| \gamma^j \|^2} \tag{12.25}$$

where θ_j is a positive scalar between 0 and 2. Often the θ_j are determined by setting $\theta_0 = 2$ and halving θ_j whenever $L(u)$ has failed to increase for some fixed number of iterations. The term L^* is an upper bound on the optimal value $L(u)$. By weak duality, one such valid bound is the value of any primal feasible solution. Getting a subgradient algorithm to work in practice requires a considerable amount of "tweaking" the parameters. The best parameter values are very problem specific.

Example 12.19 (The Generalized Assignment Problem) *The generalized assignment problem has the following generic form.*

$$\min \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} x_{ij} \tag{12.26}$$

$$(GAP) \quad \text{s.t.} \quad \sum_{j=1}^{m} x_{ij} = 1, \quad i = 1, \ldots, n \tag{12.27}$$

$$\sum_{i=1}^{n} a_{ij} x_{ij} \le b_j, \quad j = 1, \ldots, m \tag{12.28}$$

$$x_{ij} \in \{0, 1\} \tag{12.29}$$

The problem was first introduced in Subsection 1.3.3 in Chapter 1. In this formulation $x_{ij} = 1$ if "task" i is assigned to "server" j, 0, otherwise. In the objective function (12.26) c_{ij} is the cost of assigning task i to server j. Constraint set (12.27) requires that every task is assigned exactly one server. If server j performs task i, a_{ij} units of resource are consumed and server j has b_j units of this resource available. Constraint set (12.28) does not allow server j to exceed the resource available. The binary requirements (12.29) do not allow any fractional assignments. See Fisher and Jaikumar [149] for an excellent description of how the generalized assignment problem arises as a subproblem in capacitated vehicle routing problems.

In applying Lagrangian relaxation to a problem it is necessary to figure out which constraints to relax (i.e. dualize). In the generalized assignment problem this requires choosing between the assignment constraints (12.27) or the resource limitation constraints (12.28). If the resource limitation constraints are relaxed, the resulting optimization problem is

$$\begin{aligned} L(u) \;=\; \min \quad & \sum_{i=1}^{n} \sum_{j=1}^{m} (c_{ij} + u_j a_{ij}) x_{ij} - \textstyle\sum_{j=1}^{m} u_j b_j \\ \text{s.t.} \quad & \sum_{j=1}^{m} x_{ij} = 1, \quad i = 1, \ldots, n \\ & x_{ij} \in \{0, 1\} \end{aligned}$$

It is easy to show that the Lagrangian function $L(u)$ has the integrality property and for the optimal u, $L(u)$ will equal the linear programming relaxation value of (GAP). The linear programming relaxation for generalized assignment problems is usually a very poor approximation for the binary solution. It is usually much better to relax the assignment constraints so that the resulting dual function is

$$\begin{aligned} L(u) \;=\; \min \quad & \sum_{i=1}^{n} \textstyle\sum_{j=1}^{m} (c_{ij} - u_i) x_{ij} + \displaystyle\sum_{i=1}^{n} u_i \\ \text{s.t.} \quad & \sum_{i=1}^{n} a_{ij} x_{ij} \le b_j, \quad j = 1, \ldots, m \\ & x_{ij} \in \{0, 1\} \end{aligned}$$

This Lagrangian function has considerable special structure. Even though it does not have the integrality property and is an integer program, it separates into

m distinct knapsack problems, one for each server. There are numerous special purpose algorithms for solving knapsack problems. See the book by Martello and Toth [313]. Since $L(u)$ is optimized relatively easily given a u vector, this Lagrangian function is a good candidate for subgradient optimization. We illustrate the process on the specific example below.

$$\begin{aligned} \min \quad 9x_{11} + 2x_{12} + x_{21} + 2x_{22} + 3x_{31} + 8x_{32} & \\ x_{11} + x_{12} &= 1 \\ x_{21} + x_{22} &= 1 \\ x_{31} + x_{32} &= 1 \\ 6x_{11} + 7x_{21} + 9x_{31} &\le 13 \\ 8x_{12} + 5x_{22} + 6x_{32} &\le 11 \\ x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32} &\in \{0, 1\} \end{aligned}$$

The linear programming relaxation solution for this example is given below.

```
OBJECTIVE VALUE =    6.42857120

        OBJECTIVE FUNCTION VALUE

        1)       6.428571

 VARIABLE         VALUE          REDUCED COST
      X12        1.000000           .000000
      X21         .571429           .000000
      X22         .428571           .000000
      X31        1.000000         -3.714286


      ROW   SLACK OR SURPLUS     DUAL PRICES
       2)         .000000         -2.000000
       3)         .000000         -2.000000
       4)         .000000         -8.000000
       5)         .000000           .142857
```

Let u^0 be the optimal linear programming relaxation dual variable values on the assignment constraints. Then

$$\begin{aligned} L(u^0) = \min \quad & 7x_{11} + 0x_{12} - x_{21} + 0x_{22} - 5x_{31} + 0x_{32} + 12 \\ \text{s.t.} \quad & 6x_{11} + 7x_{21} + 9x_{31} \le 13 \\ & 8x_{12} + 5x_{22} + 6x_{32} \le 11 \\ & x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32} \in \{0, 1\} \end{aligned}$$