PART I

1

A Flyover Introduction to Integer Linear Programming

Integer linear programming (more commonly called *integer programming*) is a versatile computational method that is widely used in management, engineering, industry, banking, transportation, etc. However, it is almost unknown to biologists, although computer scientists and mathematicians are increasingly using integer programming in computational and systems biology. This chapter introduces integer linear programming (ILP), starting with the foundational topic of *linear programming* (LP).

1.1 LINEAR PROGRAMMING (LP) AND ITS USE

It is helpful to divide the discussion of linear programming (and integer linear programming) into four parts: the *problem to be solved*; the *concrete formulation* of a linear program (or model), given all the data required to specify a specific problem instance; the *solution* of a concrete formulation; and the *abstract formulation* of a linear program.

We will explain these four elements using a simplified version of a real problem (discussed in [42, 46]) from *conservation ecology*. We take liberties in describing that work in order to simplify the presentation.[1]

1.1.1 The Threatened Species Protection Problem

The initial work in [46] concerns conservation of "remnant patches of bush on the Eyre Peninsula, South Australia." The latter work in [42] extends the approach in [46], addressing conservation of endangered plant species in the *Cape of South Africa*, which is

[1] This problem is a bit atypical for this book, first because it comes from *ecology*, while most of the book involves problems in *genomics, genetics, phylogenetics, RNA, protein, networks, and disease*; second, because it translates almost directly into a linear program and integer linear program, while most of the problems in the book have less direct translations to LP and ILP. But, the ease of translation allows a simpler introduction to linear programming. Finally, the problem is a bit atypical because it involves a question of biological *management* rather than biological science.

> … one of the most botanically species-rich areas of the world with more than 9000 species … [42]

These species span more than 700 *genera*,[2] and at least 274 of them contain one or more species that is classified as "vulnerable, endangered, or critically endangered." And,

> [o]f the 274 threatened genera, 17 belong to the top-20 most-threatened genera of South Africa, based on the proportion of their species that are threatened. [42]

Conservation agencies have (limited) resources to preserve *some* areas of the Cape, and hence to help protect some of the threatened species. So, the general question is how to most effectively use the available resources to protect threatened species. The analogous question for threatened animal species was recently addressed in a perspective piece in *Science* magazine [76].

There are many variants, extensions, and specializations of this conservation problem. We start with a simple variant, but will return to the general problem throughout the book, extending it as we explain more about LP and ILP.

In [42], the Cape is divided into about 200 square regions, each containing about 675 square kilometers. It is assumed that in each region, we know the abundance of each of the 274 threatened species. In our telling of the story, we measure abundance of a species in a region by the area (in square kms) occupied by that species.[3] The cost to preserve *all* the land in any region is also assumed to be known, but partial preservation of a region is also possible. Then, for each threatened species, a *conservation target* is established, meaning that the species would be considered protected if the total area occupied by the species in the preserved land (over all the regions) contains more than its conservation target. With all of this data, the first general problem is

> **The Species Protection Problem** What is the *least expensive* way to protect *all* of the threatened species?

1.1.1.1 A Toy Concrete Example

To make all of this concrete, we will look at an artificially small *instance* of the species protection problem, with only five regions, and two threatened species. The regions are named *A, B, C, D*, and *E*; and the species are named α and β. Table 1.1 shows the area occupied by each species in each of the regions; the cost of preserving *all* the land in each region; and the conservation target for each species.

The Concrete Problem Instance The specific data in Table 1.1 defines a concrete instance of the species protection problem: What is the minimum cost way to preserve parts of the five regions, so that for each of the two threatened species, the total area in the preserved land is at least the conservation target for that species?

More important than the actual solution for this concrete problem instance, what is a *method* we can use to solve any concrete problem instance?

[2] According to Wikipedia: "genus, pl. *genera* is a taxonomic rank used in the biological classification of living and fossil organisms in biology. In the hierarchy of biological classification, genus comes above species and below family." [216]

[3] We assume that the species is distributed uniformly over the region, so for example, any half of the region will have half of the species in the region.

Table 1.1 Concrete Data for An Instance of the Species Protection Problem.

Region	Area occupied by α in square kilometers	Area occupied by β in square kilometers	Cost for preserving the region
A	24	83	$97K
B	36	11	$73K
C	0	29	$22K
D	15	0	$11K
E	40	18	$45K
Target	64	87	

The answer is to *formulate* a *linear program (LP) model* (also called an *LP formulation*) that describes (or expresses) the details of the specific problem instance. That formulation, with all the details of the problem *instance*, is called a *concrete* LP formulation. Then, to solve the concrete instance, we *solve* the concrete LP formulation using an LP *solver*. The solution will specify how much of each region to preserve, and what the total cost will be. We next discuss more about the LP formulation.

1.1.2 Creating a Concrete LP Formulation

To create an LP formulation (model) for a concrete problem instance, we begin by creating linear programming *variables*. The term "variable" in this context is the same as in high school mathematics. An LP variable can take on a *numerical value*. The LP variables for the species protection problem express the unknown values that we ultimately want to determine: variable X_A denotes the *fraction* of region A that will be preserved. Being a fraction, we restrict X_A so that it can only take on a value between 0 and 1 (inclusive). The variables X_B, X_C, X_D, and X_E have analogous meanings for regions B, C, D, and E, respectively.

The next step in formulating a concrete LP model for a problem instance is to develop *linear constraints*, which are either *inequalities* or *equalities*. The inequalities and equalities express the additional constraints on the values that are *permitted* to be assigned to the variables. A further explanation follows below.

Linear Functions and LP Constraints A *linear function* of a set of variables is formed by multiplying each variable by a specific *coefficient* (or constant number) and adding together the resulting terms. For example, suppose the set of variables is $\{X_A, X_B\}$. Then,

$$3X_A + 4X_B,$$

is a linear function of those two variables, with coefficients 3 and 4, respectively. A linear *equality* consists of a linear function followed by the equality sign ("=") and a constant; for example,

$$3X_A + 4X_B = 17.$$

A linear *inequality* consists of a linear function followed by a symbol for an inequality *relation*, ($\leq$, $\geq$, $<$, or $>$), followed by a constant number. For example,

$$3X_A + 4X_B \leq 13,$$

is a linear inequality.

The *constraints* in a linear program consist of linear inequalities (which could actually be linear equalities). Although the *definition* of a linear inequality includes the cases of *strict* inequality, i.e., $<$ and $>$, those are not allowed symbols in linear programming solvers I have used; but there are ways to achieve the effect. We will see that in several examples throughout the book.

In the concrete formulation for the toy instance of the species protection problem, we have the simple, initial constraints:

$$\begin{aligned} X_A &\leq 1, \\ X_B &\leq 1, \\ X_C &\leq 1, \\ X_D &\leq 1, \\ X_E &\leq 1. \end{aligned} \tag{1.1}$$

We do *not* need to explicitly include the constraints $X_A \geq 0$, etc., because it is already *assumed* in linear programming that the value of any variable will be *nonnegative*.[4]

More Interesting Constraints The more interesting, and complex constraints come from the requirement to protect both of the threatened species. The following constraint expresses what is needed to protect species α:

$$24X_A + 36X_B + 0X_C + 15X_D + 40X_E \geq 64. \tag{1.2}$$

To understand this, note that each term in the inequality specifies the area occupied by species α in one specific region, times the value of the X variable for that region. For example, the constant number 24 in the term $24X_A$, is total area in region A that is occupied by species α. So, once the value of X_A is specified (which is a number from 0 to 1), $24X_A$ is the amount of species α that will be protected in region A. Hence, the linear function in (1.2) specifies the *total preserved* area that will be occupied by species α.[5] The right end of the inequality has "$\geq$ 64." The overall result is that inequality (1.2) states that the total preserved area occupied by species α, *must be* greater or equal to 64, the conservation target for α.

The analogous constraint for species β is:

$$83X_A + 16X_B + 19X_C + 0X_D + 18X_E \geq 87. \tag{1.3}$$

Feasible Solutions Some combinations of values for the variables X_A, X_B, X_C, X_D *satisfy* (make true) all the inequalities in (1.1), (1.2), and (1.3), but some combinations of values *violate* one or more of the inequalities. A combination of values assigned to the variables that satisfies *all* of the inequalities is called a *feasible solution* to the constraints. If there are no feasible solutions, then the set of constraints is called *infeasible*.

For example, if we set all five variables to 0.66, then the inequalities in (1.1) are satisfied, and the sums in inequalities (1.2) and (1.3) are 75.9 and 89.76, respectively.

[4] The default assumption in linear programming that a variable can only have nonnegative value, is for convenience; it is not limiting. There are standard ways to get around it, but in all of the problems discussed in this book, all the variables will naturally have nonnegative values.

[5] Recall that we have assumed that in any region, species α is distributed equally (uniformly) throughout the region.

Hence, that assignment of values to variables satisfies all of the inequalities, and is a feasible solution. However, if we set all the variables to 0.6, then the first sum becomes 69, and the second sum becomes 81.6. In that case, the inequalities in (1.1) and (1.2) are satisfied, but inequality (1.3) is violated. Hence, that assignment of values is *not* a feasible solution.

The Objective Function So far, we have not used the *costs* required to preserve different regions, yet the stated problem is to protect both of the threatened species, spending the *minimum* (i.e., least) amount of money possible. How is that objective included in the model? When values are assigned to the five X variables, the total cost (in thousands of dollars) will be equal to:

$$97X_A + 73X_B + 22X_C + 11X_D + 25X_E. \tag{1.4}$$

So, the total cost is a *linear function* of the five X variables. Therefore, the *objective function*, which expresses the goal of spending the least money possible (while protecting both species) is stated as:

$$\text{Minimize } 97X_A + 73X_B + 22X_C + 11X_D + 25X_E. \tag{1.5}$$

1.1.2.1 The Concrete Linear Programming Formulation

The objective function (1.5) together with the inequalities in (1.1), (1.2), and (1.3) form the *concrete linear programming formulation* for the concrete problem instance of the species protection problem. Summarizing, the full concrete LP formulation for the toy problem instance is shown in Figure 1.1.

A *feasible solution* to a concrete LP formulation is an assignment of values to the variables that satisfies all of the constraints. However, a feasible solution does *not* need to be one that *minimizes* the objective function. A feasible solution the minimizes the objective function is called an *optimal solution*.[6]

Given the values of the variables in a feasible solution, the resulting value of the linear function in the objective is called the *objective value* or the *value of the solution*.

For example, when all the variables are given the value 0.66, the value of the solution is 150.48. As we will see later, this feasible solution is *not* an optimal solution, because there is a feasible solution with smaller objective value.

LP Solvers for Concrete LP Formulations A concrete LP formulation has all the information required to allow a solution to the specific problem instance. The formulation can then be input to an *LP solver* (in the proper format). If there is a feasible solution to the concrete LP formulation, the LP solver will determine and report an *optimal* solution. Notice that the phrasing allows for the possibility that there is more than one optimal solution, which is often the case.

[6] The terms *feasible solution* and *optimal solution* can be confusing, and keeping the distinction between them will often be crucial in this book. It is even more confusing when an assignment of values is just referred to as a *solution*. In general (and I hope I have been completely consistent in this book), when the term "solution" is used by itself, it is shorthand for a "feasible solution," which *might not* be an optimal solution. The word "optimal" should always be included when making the point that a feasible solution is an optimal solution.

$$\text{Minimize } 97X_A + 73X_B + 22X_C + 11X_D + 25X_E$$

Subject to the constraints:

$$24X_A + 36X_B + 0X_C + 15X_D + 40X_E \geq 64$$

$$83X_A + 16X_B + 19X_C + 0X_D + 18X_E \geq 87$$

$$X_A \leq 1$$
$$X_B \leq 1$$
$$X_C \leq 1$$
$$X_D \leq 1$$
$$X_E \leq 1$$

Figure 1.1 The Full Concrete LP Formulation of an Instance of the Species Protection Problem. This is called a "concrete" LP formulation because it contains all of the information in this particular problem *instance*. Usually, the phrase "subject to the constraints" is abbreviated to "st," or "such that." Note that the objective function is a linear function of a subset (possibly the whole set) of the LP variables, and that each of the constraints is a linear inequality, defined on the LP variables. The last five inequalities are also called *bounds* because each one provides a bound (upper bound in this example) on a single LP variable.

Alternatively, if there is *no* feasible solution to the concrete LP formulation, the LP solver will determine and report that; and if there is a feasible solution, but there is no bound on the value of the feasible solutions (essentially infinity for maximization problems, or negative infinity for minimization problems), the LP solver will determine and report that fact. An unbounded solution is usually an indication of an error in the general problem specification, or in the logic of the LP formulation, or in the concrete LP formulation.

The Concrete Optimal Solution In the concrete LP formulation in Figure 1.1, an optimal solution has objective value (after rounding up) of 108.7, which is achieved by setting (after rounding up) X_A to 0.831, X_D to 0.269, X_E to 1, and the other two variables to 0. So, in this solution, none of regions B and C will be preserved, all of region E will be preserved, and regions A and D will be partly preserved. Note, that there might be other optimal solutions that will assign different values to the variables, but *all* optimal solutions will have the same (rounded up) objective value, i.e., 108.7 in this example.

Exercise 1.1.1 *Use the values given to X_A, X_B, X_C, X_D, and X_E in the optimal solution detailed above, to determine the total preserved area occupied by species α, and the total preserved area occupied by species β. Do you see anything interesting?*

1.1.3 Refinements to the Model

One of the most useful features of linear programming is the ease in which *"what if"* questions can be explored, once the first LP formulation is created and solved. As an illustration, suppose there is pressure to specifically preserve some land in region B, which is not preserved at all in the optimal solution found above. But, land in region B is relatively expensive and the conservation agencies have limited resources. From the optimal solution to the concrete formulation above, we know that both of the threatened species can be protected for \$108.7K. Even with spending no more than that amount, but reducing what is spent on the other regions, it *might* still be possible to preserve *some* of region B, while protecting both species α and β. So, we can ask:

> How much of region B can be preserved, without spending more than \$108.7K, while still protecting both species α and β? And, how do we figure out the answer to this question?

The answer to the second question is to use linear programming again, modifying the concrete LP formulation in Figure 1.1. We change the objective function to:

$$\text{Maximize } X_B,$$

and add the constraint:

$$97X_A + 73X_B + 22X_C + 11X_D + 25X_E \leq 108.7.$$

Then we use the LP solver to find an optimal solution to the modified concrete LP formulation. Running the solver, we get a new optimal solution with objective value of 0.00296. That means that it is *not* possible to preserve more than a very small amount (less than one third of 1%) of region B, without increasing the total amount spent for land preservation.

Exercise 1.1.2 *In the optimal solution to the modified LP formulation, the values given to the five variables are:* $X_A = 0.830, X_B = 0.00296, X_C = 0, X_D = 0.263$ *and* $X_E = 1$. *What do you think will be the result if you plug those values into the original objective function in Figure 1.1. That is, what is the result of plugging the values of the five variables into the linear function:*

$$97X_A + 73X_B + 22X_C + 11X_D + 25X_E.$$

Try to answer this and to explain you answer without *actually plugging in the values. After that, plug in the values. What did you learn?*

Another What If Given that very little of region B can be preserved (while protecting both α and β) without increased spending, we could next ask:

> How much would we have to spend if we want to preserve at least 10% of region B? And, how do we solve this problem?

Of course, the answer to the second question is to use linear programming. But how, specifically? The answer is to start with the LP formulation in Figure 1.1, and add in the constraint:

$$X_B \geq 0.1.$$

Solving this concrete LP formulation results in an optimal solution with objective value of \$111.7K, an increase of only \$3K (a steal!).

Summary This toy example illustrates the three parts of every linear programming formulation: an objective function (either to *maximize* or *minimize*) a *linear* function of a (sub)set of the LP variables; a set of *linear* inequalities (constraints), each defined on a (sub)set of the LP variables; and a set of *bounds*, each defined on a single LP variable. Each bound is actually a constraint, and so could be considered as part of the constraints, but are historically distinguished from the other constraints.

1.1.4 Algorithms and LP Solvers

In the above example, we showed optimal solutions for concrete LP formulations, but did not say *how* they were obtained. The short answer: with algorithms and LP solvers.

Algorithms There are several *algorithms* (well-specified methods) that can take any concrete LP formulation and find an optimal solution; or determine that the formulation is infeasible; or that the solution value is unbounded. The latter case is usually not a sensible result, and usually indicates a user-created error.

The first and most famous LP algorithm is the *Simplex Algorithm*, developed by George Dantzig shortly after World War II. It is still the basis for many practical LP solvers, although additional refinements have been made to the original method. Further, other algorithms were later developed that are based on very different ideas than the simplex algorithm. Some of these later algorithms have theoretical properties that the simplex algorithm lacks. For example, some LP algorithms are *provably efficient* in a *worst-case* theoretical sense which is a property that the simplex algorithm does not have, despite its efficiency *in practice*. However, for the purposes of this book, we don't need the details of any of these algorithms, or any of their theoretical properties. What is important in this book, is the fact that highly engineered computer programs have been developed that implement LP algorithms, and these programs are very effective in practice.

LP Solvers When the details of an LP algorithm are written into an executable computer program, the program is called an *LP solver*. An LP solver takes in a concrete LP formulation (in some, usually rigid, format), and returns the value of the optimal solution, together with values assigned to the LP variables in the optimal solution.

In this book, I discuss the LP solver developed by *Gurobi Optimization* ®. In my experience, Gurobi is the fastest and most reliable of the two major LP solvers, and it has excellent documentation and support. Gurobi is a proprietary, commercially created LP solver that has been extensively tuned and engineered. Fortunately, Gurobi offers free licenses, of their full software, for academic and research users.[7]

[7] It is beyond the scope of this book to review all available LP solvers, but I should mention that the other major LP solver is *Cplex*, and it is currently owned by IBM. Cplex ®, following Gurobi's lead, currently also makes free licenses available to academic users. Two free, open-source LP solvers are COIN-LP, available from *COIN-OR*, and GLPK (*GNU linear programming kit*), available from the *GNU Project*. In my experience, *GLPK* is effective on some moderate-size LP formulations, although it generally runs much slower than Gurobi or Cplex, and for more demanding formulations, it does not find the optimal in a reasonable time, while Gurobi and Cplex do. I don't have experience using COIN-LP, although I believe it has a good reputation.

1.2 INTEGER LINEAR PROGRAMMING (ILP)

Linear programming allows the LP variables to be given *fractional*, i.e., noninteger, values, as we saw in the optimal solution to the species protection problem. *Integer* linear programming refines linear programming by *requiring* that certain variables in a formulation only be given *integer* values.[8] We will also refer to an *integer linear function* to mean that the function is linear, and that its variables are only allowed to take on integer values. Similarly, an *integer* linear inequality is an inequality whose variables are restricted to be integers. However, the *coefficients* (i.e., constants) in the integer linear functions or inequalities (in the objective function and in the constraints) are still allowed to be fractional.

For example, suppose the five variables in the LP formulation in Figure 1.1 are now required to take only integer values. That turns the LP formulation into an ILP formulation. Now, because of the inequalities in (1.1), each variable must be assigned either 0 or 1. This means that in any feasible solution, for every region, the solution must either preserve *all* of that region, or *none* of it. It is no longer possible to preserve only part of a region. Does that really affect what objective value is possible?

Yes. The integer optimal solution to the toy species preservation problem has objective value of \$122K, in contrast to the optimal LP solution, which has value of \$108.7K. Remembering that *value* in this case is actually a *cost*, the LP optimal solution is less expensive than the ILP optimal solution. In one ILP optimal solution, variables X_A and X_E are set to 1, and the other three variables are set to 0. In that solution, the total preserved area occupied by species α is 64, exactly meeting the conservation target for species α; but, the preserved area occupied by species β is 101, well above the conservation target for β.

Exercise 1.2.1 *Exercise 1.1.1 asked you to calculate the total preserved area occupied by species α and β, respectively, implied by the optimal LP solution to the toy species preservation problem. Compare those numbers to the areas of 64 and 101, stated above for the ILP optimal. Do you see a general explanation for what you observed?*

More Terminology A integer variable that is further constrained to only take on value 0 or 1 is called a *binary variable*. An ILP formulation where the variables are further constrained to only take on values of 0 or 1, is called a *binary* formulation. Binary formulations are very common. The ILP for the toy species protection problem is a binary formulation, and most of the ILP formulations in this book will be binary formulations.

When the values assigned to the variables in an ILP formulation satisfy all of the constraints, and all of the variables required to have integer values, do have integer values, the solution is called an *integer feasible solution*. An integer feasible solution with the best objective value (maximum or minimum, depending on the objective function) is called an *integer optimal* solution.

Relating LP and ILP Suppose $\mathcal{P}$ denotes an ILP formulation, and $\mathcal{P}'$ denotes the same formulation where we allow *all* of the variables to take on *fractional* values,

[8] More commonly, when some variables are required to have integer values, and some are allowed to have fractional values, the term *mixed integer linear programming (MILP, or sometimes MIP)* is used. For simplicity, in this book, I will just use the acronym "ILP," even if some of the variables are allowed to have fractional values.