

## 1 Introduction

### 1.1 Limitations of LP

We have seen how Linear Programming (LP) problems can be used to answer economics-based questions such as optimal production rates and buying strategies. However, sometimes the answers produced using LP techniques are nonsensical when translated into their economic counterparts. For instance, it does no good to us when the simplex method tells us to build 2.4 new warehouses and 3.2 new factories.

The problem here is that the variables in our LP represent items that come in units that are not divisible. For a solution to be of any use to us, we must impose an additional constraint: all the variables in a feasible solution must take on integer values. Thus, we have developed a new problem, the **Integer Linear Programing** problem (or **ILP** problem). Indeed, adding integrality constraints to any optimization problem creates a new type of problem, but we will focus on integrality constraints in Linear Programming.

### 1.2 Differences between LP and ILP

Consider an arbitrary ILP problem:

$$\begin{aligned} \text{Maximize: } & c^T x = z \\ \text{Subject to: } & Ax \leq b \\ & x \geq 0 \\ & x_j \text{ is integer } \quad j = 1, 2, \dots, n \end{aligned}$$

Notice that if we drop (or “relax”) the integrality constraint, we are left with an LP problem. This is called the *LP relaxation* of the ILP. An important result to notice is every feasible solution of an ILP is also a solution of its LP relaxation. That is,

$$F_{ILP} \subseteq F_{LP}$$

One might think that since there are fewer solutions for an ILP (finitely many, if the solution set is bounded) than for its relaxation, that it is easier to find a solution for the ILP; this is not the case. In fact, it is in general much more difficult to solve an ILP than to solve an LP of similar size. This is because LP-solving algorithms take advantage of the fact that an optimal solution of an LP is a basic solution of the LP and a corner point of the feasible region. The feasible region for an ILP is not continuous, so these methods cannot be applied.

### 1.3 Binary ILP Problems

There is a subset of ILP problems that is worth separate consideration. These problems are problems in which the variables correspond to Yes/No questions. That is, given a particular solution of the problem, question  $j$  is answered with “yes” if  $x_j = 1$ , and with “no” if  $x_j = 0$ . This is particularly helpful in many network models where problems correspond to decisions such as whether to open a certain facility or not, or whether to

assign an employee to a certain task or not. These are handled similarly to ILP problems, adding a further constraint that  $x_j$  takes on the value 0 or 1.

Any ILP problem can be transformed into a Binary ILP problem if we can find a bound on the value of each decision variable in the problem. For instance, suppose that we know that variable  $x_i$  is bounded above by  $2^n - 1$ . Then we can rewrite the variable as  $x_i = \sum_{k=1}^n y_k 2^{k-1}$  where each  $y_k$  is a binary variable.

## 2 Brute Force Methods

When trying to develop methods to solve a problem, it is helpful to consider the brute force approach to get an idea of what our limitations are.

### 2.1 Enumeration

As stated earlier, the number of feasible solutions for an ILP is finite if the feasible region is bounded, so enumeration is a possible approach. In a Binary ILP problem with  $n$  variables, there are at most  $2^n$  solutions to be considered. Many of these solutions may not be feasible due to constraints, but the important thing to note here is that the number of solutions is exponential, and therefore enumeration may not be a desirable solution for problems with many variables.

### 2.2 Graphical Approach

As with LP problems, if our ILP has only two variables, a graphical approach can be used to obtain the optimal solution to the problem.

### 2.3 Rounding

Another natural thought is to run the simplex method on the LP relaxation of an ILP and then round the solution to an integer solution. First note that the nearest integer solution may not be in the feasible region, as in Figure 1 ([1]). Another method worth considering is rounding to the nearest *feasible* solution, but Figure 2 ([1]) shows that this may not find the optimal solution for the ILP.

## 3 The Cutting Plane Algorithm

One popular method for solving ILP problems is the cutting plane algorithm, and it stems from two main observations ([2]):

- **Observation 1:** If the LP relaxation of an ILP has an optimal solution  $x^*$  that is integral, then  $x^*$  is an optimal solution of the ILP.
- **Observation 2:** There may exist linear inequalities which are satisfied by every feasible solution of the ILP but violated by some solutions of the LP relaxation.

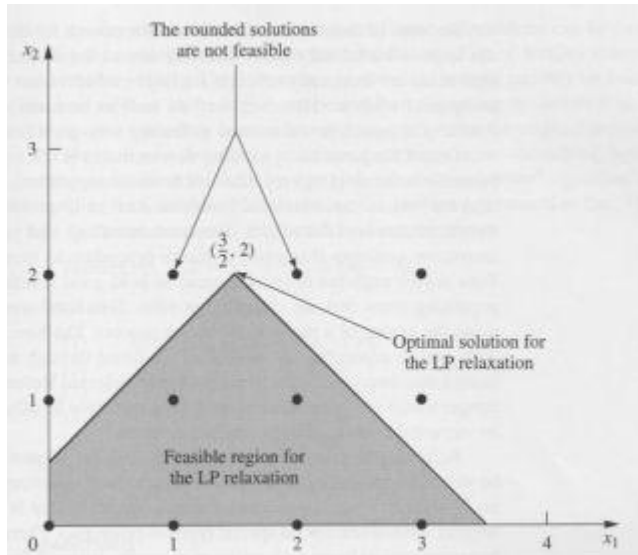


Figure 1

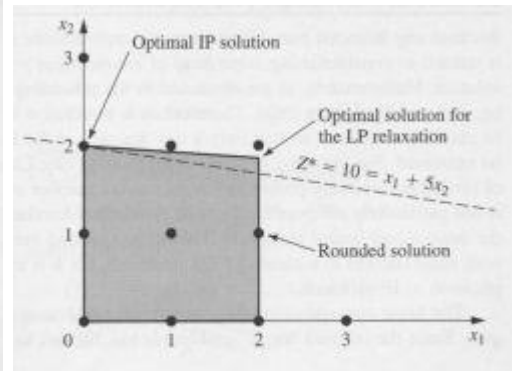


Figure 2

Both of these observations come from the fact stated earlier, that  $F_{ILP} \subseteq F_{LP}$ . To find the optimal solution for our ILP, we can add constraints to the problem, leaving the feasible region intact, until the optimal solution of the LP relaxation is integral. But how do we find these constraints?

### 3.1 Finding the Cutting Planes

To find a cutting plane for our problem, we first solve the LP relaxation using the simplex method as discussed in previous lectures. In our final tableau, there exists some row corresponding to a basic variable in which the value in the final column of this row (call it  $b_i$ ) is a non-integer value (if not, then the basic solution to the LP relaxation is integral and must be an optimal solution to the ILP). There must also exist values in this row that are non-integer (otherwise we have a sum of integral values being non-integer). We define the cut as follows:

$$(a_{i1} - \lfloor a_{i1} \rfloor)x_1 + \dots + (a_{in} - \lfloor a_{in} \rfloor)x_n \geq b_i - \lfloor b_i \rfloor$$

Putting this in terms of our original decision variables (many of these  $x_i$ s will be slack variables), we will get a new constraint. If we set aside all questions about why we are doing this and add this constraint to our original ILP, we see that the feasible region of the ILP remains the same, while the feasible region of the LP relaxation of this new problem is smaller than that of the previous problem. Continuing this process, we can eventually manipulate the LP relaxation to one where the optimal solution is integral.

### 3.2 The Cutting Plane Algorithm ([2])

(Assume the problem is not unbounded)

1. Solve the LP problem
2. If it is infeasible, stop; the ILP problem is infeasible
3. Let  $x^*$  be a basic optimal solution of the LP problem

4. If  $x^*$  is integral, stop; it is optimal for the ILP problem
5. Choose a basic variable,  $x_i$  such that  $b_i$  is not integer
6. Form the cut from the  $x_h$  row
7. Add the cut to the ILP problem and go to step 1

### 3.3 Example

Consider the following ILP:

$$\begin{array}{rcll}
 \text{Maximize:} & 2x_1 & + & 3x_2 & = & z \\
 \text{Subject to:} & -x_1 & + & 2x_2 & \leq & 6 \\
 & x_1 & + & 3x_2 & \leq & 15 \\
 & 2x_1 & + & x_2 & \leq & 12 \\
 & x_1, & & x_2 & \geq & 0 \\
 & x_1, & & x_2 & & \text{are integers.}
 \end{array}$$

Solving the LP relaxation by the simplex method yields the optimal solution  $(\frac{21}{5}, \frac{18}{5})$ , with objective value  $\frac{96}{5}$ . The final tableau of this simplex method is

$$\begin{array}{cccccc}
 0 & 0 & 1 & -\frac{3}{5} & \frac{4}{5} & \frac{33}{5} \\
 0 & 1 & 0 & \frac{2}{5} & -\frac{1}{5} & \frac{18}{5} \\
 1 & 0 & 0 & -\frac{1}{5} & \frac{3}{5} & \frac{21}{5} \\
 \hline
 0 & 0 & 0 & \frac{4}{5} & \frac{1}{5} & \frac{96}{5}
 \end{array}$$

From this tableau, we find a row corresponding to a basic variable whose value is non-integer. Choose  $x_2$ . The cut from this row is  $\frac{2}{5}s_2 + \frac{4}{5}s_3 \geq \frac{3}{5}$ . Remembering that  $s_2 = 15 - x_1 - 3x_2$  and  $s_3 = 12 - 2x_1 - x_2$  and simplifying, we get the inequality  $2x_1 + 2x_2 \leq 15$ . We add this to our original problem to get a new ILP:

$$\begin{array}{rcll}
 \text{Maximize:} & 2x_1 & + & 3x_2 & = & z \\
 \text{Subject to:} & -x_1 & + & 2x_2 & \leq & 6 \\
 & x_1 & + & 3x_2 & \leq & 15 \\
 & 2x_1 & + & x_2 & \leq & 12 \\
 & 2x_1 & + & 2x_2 & \leq & 15 \\
 & x_1, & & x_2 & \geq & 0 \\
 & x_1, & & x_2 & & \text{are integers.}
 \end{array}$$

It is helpful to see how adding this constraint changes the feasible region graphically. Solving the LP relaxation of this new problem with the simplex method gives us the optimal solution  $(\frac{15}{4}, \frac{15}{4})$ , with objective value  $\frac{75}{4}$ . The final tableau is as follows:

$$\begin{array}{cccccc}
 0 & 0 & 1 & -1 & 0 & 1 & 6 \\
 0 & 1 & 0 & \frac{1}{2} & 0 & -\frac{1}{4} & \frac{15}{4} \\
 0 & 0 & 0 & \frac{1}{2} & 1 & -\frac{5}{4} & \frac{4}{3} \\
 1 & 0 & 0 & -\frac{1}{2} & 0 & \frac{3}{4} & \frac{15}{4} \\
 \hline
 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{3}{4} & \frac{75}{4}
 \end{array}$$

Forming another cut from the  $x_1$ -row (though the cut is the same from the  $x_2$ - and  $s_3$ -rows), we get  $\frac{1}{2}s_2 + \frac{3}{4}s_4 \geq \frac{3}{4}$ , which can be arranged to get  $2x_1 + 3x_2 \leq 18$ . Adding this

to our problem and solving the new LP relaxation gives the optimal solution (3,4) with optimal value 18. Since this is an integral solution, it is an optimal solution for the ILP. Since the feasible region and objective function of this latest ILP is not changed from our original ILP, it is the optimal solution we seek for the original problem.

## 4 The Branch and Bound Algorithm

Another approach to solving ILP problems is the branch-and-bound method. It is a method for solving Binary ILP problems, but as we saw above, any ILP can be transformed into a Binary ILP if we can find a bound for each variable. The concept of this method is that of *divide and conquer*, where we break down the large difficult problem into smaller subproblems. In particular, we break the feasible region of the ILP into smaller and smaller subsets and bound how good the optimal solution can be for each subset.

### 4.1 Branching

In the *branching* step, we choose one of the variables whose value we will fix to create two new subproblems. For instance, in a problem with  $n$  variables, we may choose to fix the value of  $x_1$  to create subproblems where we replace every instance of  $x_1$  with 0 and 1 respectively, resulting in problems with one less variable. Solving the LP relaxations to one of these problems will result in a solution of the form  $(x_2, \dots, x_n)$ , which corresponds to a solution of the original problem of the form  $(0, x_2, \dots, x_n)$  or  $(1, x_2, \dots, x_n)$ . In this way, we are closer to finding an integer solution to the original problem.

### 4.2 Bounding

Solving the LP relaxation of a subproblem will yield an optimal solution with an associated value  $z^*$ . By our earlier observation, the value of every integer solution in the feasible region of this subproblem must be less than or equal to  $z^*$ . In this way, we have *bounded* the subproblem. If the associated solution is integer, then we know that every integer solution has value less than or equal to  $z^*$ . Otherwise, we can break the problem down further until doing so will not make any progress.

### 4.3 Fathoming

There are three situations that will arise that lets us know when we no longer need to break down problems. We say that a subproblem is *fathomed* if

1. Its bound is less than or equal to that of any feasible integer solution we have acquired
2. Its LP relaxation has no feasible solutions
3. Its LP relaxation's optimal solution is integral

## 4.4 The Branch and Bound Algorithm ([1])

Initialization: Set  $z^* = -\infty$ . Apply the bounding step, fathoming step, and optimality test to the whole problem. If not fathomed, classify it as a subproblem.

For each iteration:

1. **Branching:** Select an unfathomed subproblem. Branch from this problem by fixing a remaining variable as either 0 or 1.
2. **Bounding:** For each new subproblem, obtain its bound by solving the LP relaxation and rounding down the value of  $z$  for its optimal solution.
3. **Fathoming:** For each new subproblem, apply the three fathoming tests above. If its LP relaxation's solution is integer, compare its value  $z$  to  $z^*$ ; If its value is greater, than this solution is the new incumbent, and  $z^* = z$ . Reapply fathoming test 1 to all unfathomed subproblems.
4. **Optimality Test:** Stop when there are no remaining subproblems; the current incumbent is optimal to the original problem.

## 4.5 Example

Let's solve the same problem as earlier using the Branch and Bound algorithm. First, however, we must translate from an ILP problem to a binary ILP problem. To do this, we must first bound our variables  $x_1$  and  $x_2$ . From the third constraint, we can see that  $x_1$  is bounded above by 6, and from the second constraint, we can see that  $x_2$  is bounded above by 5. So, let's define binary variables  $y_1, \dots, y_6$  where  $x_1 = y_1 + 2y_2 + 4y_3$ , and  $x_2 = y_4 + 2y_5 + 4y_6$ . Now we can restate our problem as

$$\begin{array}{ll}
 \text{(BILP) Maximize:} & 2y_1 + 4y_2 + 8y_3 + 3y_4 + 6y_5 + 12y_6 = z \\
 \text{Subject to:} & -y_1 - 2y_2 - 4y_3 + 2y_4 + 2y_5 + 2y_6 \leq 6 \\
 & y_1 + 2y_2 + 4y_3 + 3y_4 + 6y_5 + 12y_6 \leq 15 \\
 & 2y_1 + 4y_2 + 8y_3 + y_4 + 2y_5 + 4y_6 \leq 12 \\
 & y_1, y_2, y_3, y_4, y_5, y_6 \leq 1 \\
 & y_1, y_2, y_3, y_4, y_5, y_6 \geq 0 \\
 & y_1, y_2, y_3, y_4, y_5, y_6 \text{ are integers.}
 \end{array}$$

Solving the LP relaxation gives the optimal solution  $(0, \frac{1}{10}, 1, 0, 0, \frac{9}{10})$  with value  $\frac{96}{5}$ . Thus, the bound on this problem is  $\lfloor \frac{96}{5} \rfloor = 19$ .  $19 \not\leq -\infty$ , the LP relaxation is feasible, and the optimal solution is not integral, so this problem is not fathomed.

Unfortunately, due to the nature of this example, no subproblems are fathomed until we have fixed at least the first three variables, which leads to the construction of up to 14 subproblems, each with 6 constraints. To save space and time, most of these subproblems are attached at the end of the lecture notes along with the optimal solutions of their LP relaxations, and we refer to them to note interesting steps of the algorithm rather than doing the problem from start to finish.

The first note of interest is that all problems where we have fixed a variable has a constant in the objective function. While this disrupts the standard form in which we

have written LP problems in the past, it is necessary since we are comparing the values of solutions from related problems.

Notice that on the first page, none of the six problems we encounter are fathomed: our value  $z^*$  is still  $-\infty$ , none of these problems are infeasible, and none of the solutions to the LP relaxations are integral (aren't you glad you didn't have to sit through that?).

When we solve the LP relaxation to (BILP(0,1,0)), something finally happens: we get an integer solution. Its value is noticeably larger than  $-\infty$ , so the solution (0,1,0,0,0,1) becomes our incumbent solution and we set  $z^* = 16$ . We can now fathom the problem (BILP(0,0,0)), since no integer solution will have value greater than 9.

Solving the LP relaxation for (BILP(0,1,1)) gives another integer solution, but its value is not greater than 16, so it does not become the incumbent solution. The problem is fathomed since it gave an integral solution.

The problems (BILP(1,0,0)) and (BILP(1,0,1)) will both be fathomed since their bounds are less than our  $z^*$ .

(BILP(1,1,0)) gives an integer solution with a better value than 16, so this solution becomes our new incumbent, and we set  $z^* = 18$ .

Notice that in (BILP(1,1,1)), the third constraint renders this problem infeasible, and thus this problem is fathomed.

At this point, (BILP(0,0,1)) remains unfathomed, and we break it down further. The algorithm will continue until there remain no unfathomed subproblems, at which point we will be left with the conclusion that (1,1,0,0,0,1) is the optimal solution to our original BILP. This solution corresponds to the solution (3,4) in our original ILP problem.

## 5 Closing Remarks

There are many other types of problems that require the use of binary variables. For instance, we may be working with a problem where we would like one of two (or in general,  $K$  of  $N$ ) constraints to be satisfied. This can be solved by defining new binary variables in our problem. This example also leads to another question: what do we do if only *some* of our variables have integrality constraints? Some work needs to be done before either of the two methods discussed today may be used to solve such a problem.

I'm sure you've also noticed that I've skipped over a lot of the mathematics behind what we are doing, particularly in how we have defined cuts and our lack of a proof that these cuts will not only be satisfied by all feasible integer solutions, but that there exist feasible non-integer solutions that do not satisfy these cuts, shrinking the feasible region for the LP relaxation, guaranteeing termination of our algorithm.

## 6 References

- [1] F. Hillier, G. Lieberman, *Introduction to Operations Research*, McGraw Hill, 2001
- [2] Course note for C&O 350, Winter 2006, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, 2005