Shu Lu *and* Quoc Tran-Dinh

Introduction to Optimization

From Linear Programming to Nonlinear Programming

November 25, 2019

STOR-UNC-Chapel Hill

Chapter 9

Introduction to Integer Programming

9.1 Modeling with integer variables

Like linear programming problems, integer programming (IP) problems are a special class of optimization problems. In an integer program, some or all of the decision variables are required to be integers. In any integer program considered in this course, the objective function and all constraints other than the integrality requirement must be linear. Accordingly, if we drop the integrality requirement in an integer program, we will obtain a linear program, which is called the **LP relaxation** of the integer program.

Integer programs can be classified into **pure** integer programs and **mixed** integer programs. An integer program with all of its variables required to be integers is a pure integer program. An integer program with some variables not subject to the integrality constraint is a **mixed integer program**.

Example 9.1. Woody's Furniture Company produces chairs and tables. Chairs are made entirely out of pine, and each chair uses 14 linear feet of pine. Tables are made of pine and mahogany, and each table uses 26 linear feet of pine and 15 linear feet of mahogany. Each chair requires 8 hours of labor to produce, and each table requires 3 hours of labor. The profit from each chair is \$35, and the profit from each tables is \$60.

Woody has 190 linear feet of pine and 60 linear feet of mahogany available each day, and has a work force of 92 labor hours each day. How should Woody use his resources to achieve the maximum daily profit, if the numbers of chairs and tables he makes daily have to be integers?

IP model: Let x_1 and x_2 denote the numbers of chairs and tables to be made each day respectively. We formulate the following IP problem:

$$\begin{cases} \max_x \; z = 35x_1 + 60x_2 & \text{profit} \\ 14x_1 + 26x_2 \le 190 & \text{pine} \\ 15x_2 \le 60 & \text{mahogany} \\ 8x_1 + 3x_2 \le 92 & \text{labor} \\ x_1, x_2 \ge 0 \text{ and are integer} \end{cases} \tag{9.1}$$

LP relaxation: By dropping the integrality constraint on x, we obtain the following LP relaxation:

$$\begin{cases} \max_x \; z = 35x_1 + 60x_2 & \text{profit} \\ 14x_1 + 26x_2 \le 190 & \text{pine} \\ 15x_2 \le 60 & \text{mahogany} \\ 8x_1 + 3x_2 \le 92 & \text{labor} \\ x_1, x_2 \ge 0 \end{cases} \tag{9.2}$$

Solving the LP problem (9.2) graphically as shown in Figure 9.1, we find its optimal solution to be $(10.976, 1.398)$. Rounding the solution to nearest integers, we obtain

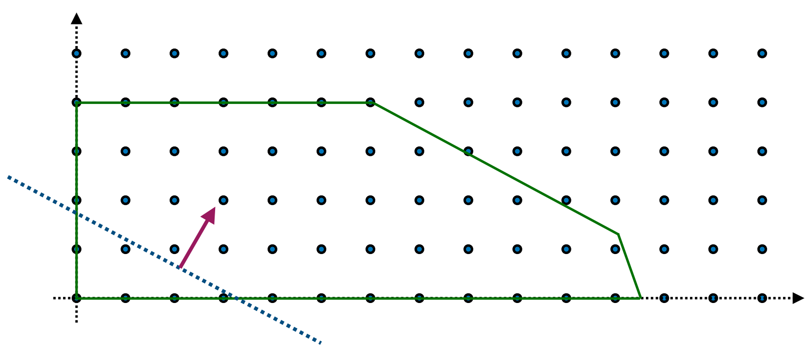


Fig. 9.1 Graphical illustration of Woody's IP and the LP relaxation

an integer solution $(11,1)$. However, the optimal solution for (9.1) is in fact $(8,3)$ instead of (11,1). In general, one cannot find an optimal solution of an integer program by rounding an optimal solution of its LP relaxation to integer. A basic method to solve integer programs is called the **branch and bound method**, the topic of next section.

GAMS model: To solve the integer program (9.1) using GAMS, we declare x_1 and x_2 as integer variables. This is done in furniture.gms by the following statement

```
integer variables x(I);
```

where `I` is the set that contains two labels "chair" and "tab." By default, GAMS sets a lower bound of zero to integer variables, so it is not necessary to include the nonnegativity constraint of `x(I)` in the model. In the solve statement,

```
solve woody using mip maximizing profit;
```

the word "mip" is the model type that stands for mixed integer programming.

```
*furniture.gms
SET
        I "Type of products" /chair, tab/
        J "Type of resource" /pine, mahogany, labor/;
PARAMETERS
        sellingPrice(I) /chair 35, tab 60/,
        resLimit(J) /pine 190, mahogany 60, labor 92/;
TABLE  a(I,J)
                pine    mahogany    labor
        chair    14      0            8
        tab      26      15           3;
FREE VARIABLE profit;
INTEGER VARIABLES x(I);
EQUATIONS obj, resLim(J);
    obj.. profit =e=   sum(I, sellingPrice(I)*x(I));
    resLim(J).. sum(I, a(I,J)*x(I)) =l= resLimit(J);
MODEL Woody /all/;
    option mip = cplex;
    $onecho > cplex.opt
    epgap=0.01
    $offecho
    woody.optfile = 1;
SOLVE Woody USING mip MAXIMIZING profit;
```

In (9.1), the integrality constraints on x_1 and x_2 arise from the explicit requirement that only integer values are acceptable as numbers of products. Integer variables are also widely used to represent yes/no decisions in optimization models. The following example is a typical capital budgeting problem. In such a problem, we use binary variables to represent decisions on whether or not to make investments.

Example 9.2. Stockco is considering four investments. Investment 1 will yield a net present value (NPV) of \$16000; investment 2, an NPV of \$22000; investment 3, an NPV of \$12000; and investment 4, an NPV of \$8000. Each investment requires a certain cash outflow at the present time: investment 1: \$5000; investment 2: \$7000; investment 3: \$4000; investment 4: \$3000. Stockco has a total of \$14000 to invest. How does Stockco maximize the total NPV?

IP model: We begin by defining a variable for each decision Stockco needs to make. This results in four variables x_j, $j = 1,2,3,4$. Each x_j is a binary variable that represents the decision on investment j, with $x_j = 1$ corresponding to the decision of making investment j, and $x_j = 0$ corresponding to the decision of not making investment j. The following IP is what Stockco needs to solve.

$$\begin{cases} \max\limits_{x} \ 16x_1 + 22x_2 + 12x_3 + 8x_4 & \\ \text{s.t.} \quad 5x_1 + 7x_2 + 4x_3 + 3x_4 & \leq 14 \\ \qquad x_j = 0 \text{ or } 1, \quad j = 1,2,3,4. & \end{cases} \tag{9.3}$$

The expression $16x_1 + 22x_2 + 12x_3 + 8x_4$ is the total NPV obtained by Stockco under the decision represented by the variables x_j. If $x_j = 1$, then the NPV of investment j is included in this expression; if $x_j = 0$ then the NPV of investment j is not included. Whatever combination of investments is undertaken, the expression gives the total NPV for that combination. Similarly, the expression $5x_1 + 7x_2 + 4x_3 + 3x_4$ is the total cash outflow to be paid by Stockco under the decision represented by variables x_j.

Next, we show how to express logical conditions on decisions by linear constraints on the corresponding binary variables. Consider the following requirements on the investment decisions of Stockco.

1. Stockco can invest in at most two investments.
2. If Stockco invests in investment 2, they must also invest in investment 1.

3. If Stockco invests in investment 2, they cannot invest in investment 4.

To account for the first requirement, add the constraint

$$x_1 + x_2 + x_3 + x_4 \leq 2$$

to (9.3). This eliminates any choice of three or four investments.

In terms of x_1 and x_2, the second requirement states that if $x_2 = 1$, then x_1 must also equal 1. If we add the constraint

$$x_2 \leq x_1$$

to (9.3), then we have eliminated the case in which $x_1 = 0$ and $x_2 = 1$. The other cases are (1) $x_1 = 0$ and $x_2 = 0$, (2) $x_1 = 1$ and $x_2 = 0$, (3) $x_1 = 1$ and $x_2 = 1$. Those cases satisfy both the requirement and the constraints. In summary, the constraint $x_2 \leq x_1$ is consistent with the second requirement in all cases regarding values of x_1 and x_2.

In terms of x_2 and x_4, the third requirement states that if $x_2 = 1$, then x_4 must equal 0. If we add the constraint

$$x_2 + x_4 \leq 1$$

to (9.3), then we have eliminated the case in which $x_2 = 1$ and $x_4 = 1$. The other cases are (1) $x_2 = 1$ and $x_4 = 0$, (2) $x_2 = 0$ and $x_4 = 0$, (3) $x_2 = 0$ and $x_4 = 1$. Those cases satisfy both the requirement and the constraint. In summary, the constraint $x_2 + x_4 \leq 1$ is consistent with the third requirement in all cases regarding values of x_2 and x_4.

It is important to note that the added constraints must be linear. For example, another constraint consistent with the third requirement would be $x_2 x_4 = 0$. However, the latter constraint is nonlinear, and therefore should not be included in an IP. Recall that an IP studied in this course cannot have nonlinear constraints except the integrality requirement.

GAMS model: Now, we provide a GAMS model to solve this problem:

```
*capitalB.gms
SET I investments /1*4/;
SCALAR cashLim /14/;
```

```
PARAMETERS
    NPV(I) in thousands /1 16, 2 22, 3 12, 4 8/,
    outflow(I) in thousands /1 5, 2 7, 3 4, 4 3/;
FREE VARIABLE profit;
BINARY VARIABLES x(I);
EQUATIONS obj, obCashLim;
    obj.. profit =e=   sum(I, NPV(I)*x(I));
    obCashLim.. sum(I, outflow(I)*x(I)) =l= cashLim;
MODEL Stockco /ALL/;
SOLVE Stockco USING mip MAXIMIZING profit;
```

9.2 The branch and bound method

The branch and bound method is the most widely used method for solving integer programs. Given an IP, the method proceeds as follows.

1. Remove the integrality constraints from the problem to obtain its LP relaxation. Find an optimal solution and the optimal value of the LP relaxation.
2. If all components of the optimal solution of the LP relaxation are integers, stop the algorithm: we already find the integer optimal solution for the original IP. Otherwise, branch on a variable with a fractional value to create two subproblems. (If more than one variable has a fractional value, arbitrarily choose any of these variables to branch on.)
3. Continue solving these subproblems, branching to further subproblems by adding appropriate constraints. Each time we create a new problem, we add a node to the branching-and-bound tree.
4. There are three situations in which we **fathom** a node, that is, stop branching from a subproblem.

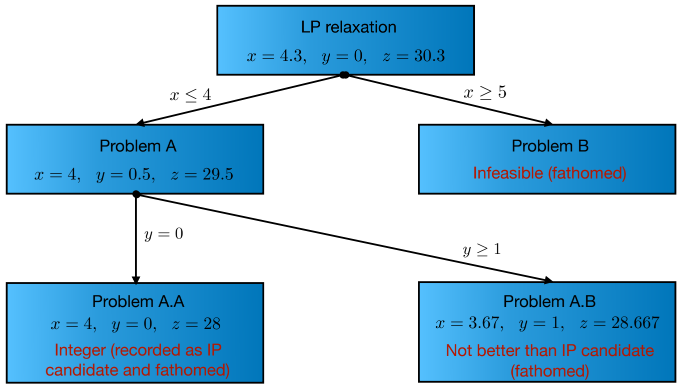
   a. This subproblem is **infeasible**.
   b. This subproblem has an **integer optimal solution**. If this integer solution achieves a better objective value than all the previously obtained integer solutions, then it becomes the **candidate IP solution** obtained so far.
   c. The optimal value of this subproblem is not better than the objective value achieved by the current candidate IP solution.

5. When there are no nodes to further branch on, stop the algorithm, with the current candidate IP solution being the IP optimal solution.

Example 9.3. Consider the following IP.

$$\begin{cases} \max\limits_{x,y} & z = 7x + 3y \\ & 2x + y \leq 9 \\ & 3x + 2y \leq 13 \\ & x, y \geq 0, \text{ and are integers.} \end{cases}$$

1. Solve the LP relaxation, to find its optimal solution is $(x,y) = (4.3, 0)$, with $z = 30.3$.
2. Since x is fractional, branch on x. Add a constraint $x \leq 4$ to the LP relaxation to obtain problem A, and another constraint $x \geq 5$ to the LP relaxation to obtain problem B. The optimal solution for problem A is $(4, 0.5)$, with $z = 29.5$. Problem B is infeasible, so it is fathomed.
3. Starting from problem A, branch on y because the y value of its optimal solution is fractional. Add a constraint $y \leq 0$ to problem A to obtain problem AA, and another constraint $y \geq 1$ to problem A to obtain problem AB. The optimal solution for problem AA is (4,0), with $z = 28$. This is the first integer solution we find so far, so we record this as the candidate IP solution, and fathom problem AA. The optimal solution for problem AB is (3.67, 1), with $z = 28.667$. Because any integer feasible solution for problem AB will have an optimal value no more than 28, and hence no better than the current candidate IP solution, we fathom problem AB as well.
4. We do not have a node to further branch on, so we stop. The current candidate IP solution, (4,0), is an optimal solution for the original IP.



9.3 Fixed charge problems

A fixed charge is an expense that does not depend on the level of goods or services produced by a business. This is in contrast to variable costs, which are volume-related (and are paid per quantity produced). To model a fixed charge in an optimization problem often requires using a binary variable.

Example 9.4. Gandhi Cloth Company makes three types of clothing: shirts, shorts, and pants. Gandhi needs to rent the appropriate machinery to make each type of clothing. The machinery can only be rented weekly. The weekly machinery rental cost, unit production cost, unit sales price, unit labor usage and unit cloth usage for each type of clothing are given below.

	Costs($)		Price($)	Unit resource usage	
Type	Rental	Production		Labor(hr)	Cloth(sq. yd.)
Shirt	200	6	12	3	4
Shorts	150	4	8	2	3
Pants	100	8	15	6	4

Each week, 150 labor hours and 160 sq. yds. of cloth are available. How should Gandhi plan weekly production to maximize his profit?

As in linear programming formulations, we define a variable for each decision that Gandhi must make. Since Gandhi needs to decide how many of each type of clothing should be manufactured each week, we define x_1, x_2, and x_3 to be the numbers of shirts, shorts and pants produced each week respectively. To account for the renting cost in the objective function, we define three binary variables y_i, $i = 1, 2, 3$, to represent whether or not to rent each type of machinery (shirts, shorts, pants). For example, $y_1 = 1$ represents the decision of renting shirt machinery, and $y_1 = 0$ represents the decision of not renting shirt machinery.

The IP model is as follows:

$$\begin{cases} \max\limits_{x} & (12x_1 + 8x_2 + 15x_3) - (6x_1 + 4x_2 + 8x_3) \\ & -(200y_1 + 150y_2 + 100y_3) \\ \text{s.t.} & 3x_1 + 2x_2 + 6x_3 \leq 150, \text{ (labor constraint)} \\ & 4x_1 + 3x_2 + 4x_3 \leq 160, \text{ (cloth constraint)} \\ & x_i \geq 0, i = 1,2,3, \\ & y_i = 0 \text{ or } 1, \ i = 1,2,3, \\ & x_i \leq M_i y_i, \ i = 1,2,3 \text{ (relation between } x_i \text{ and } y_i) \end{cases}$$

The parameter M_i that appears in the above model needs to be a large number that x_i does not exceed in any feasible solution. For example, the cloth constraint implies that $x_1 \leq \frac{160}{4} = 40$ in any feasible solution, so we can safely choose M_1 to be 40. Similarly, we can choose $M_2 = \frac{160}{3}$ and $M_3 = \frac{160}{4} = 40$. With such choices of M_i, the constraint $x_i \leq M_i y_i$ models the relation "if $x_i > 0$ then $y_i = 1$" (if any shirts are to be produced, then the shirts machinery must be rented) without imposing any unnecessary restriction on values of x_i. If M_1 were not chosen large (say $M_1 = 10$), then this constraint would unnecessarily restrict the value of x_1.

Other than the requirement that M_i needs to be an upper bound for x_i, the choice of M_i is flexible. For example the model in fixecharge.gms chooses a common M for all i. Finding an upper bound for the decision variables are usually not a problem in realistic models.

```
*fixedcharge.gms
SET
    I /shirt, shorts, pants/
    J /labor,cloth/;
SCALAR M /60/;
PARAMETERS
    sellingPrice(I) /shirt 12, shorts 8, pants 15/,
    varCost(I)  /shirt 6, shorts 4, pants 8/,
    fixCost(I) /shirt 200, shorts 150, pants 100/,
```

```
    resLimit(J) /labor 150, cloth 160/;
TABLE  a(I,J)
                labor    cloth
    shirt       3        4
    shorts      2        3
    pants       6        4;
FREE VARIABLE profit;
POSITVE VARIABLES x(I);
BINARY VARIABLES y(I);
EQUATIONS obj, resLim(J), bigM(I);
    obj.. profit =e= sum(I, sellingPrice(I)*x(I)) - sum(I, varCost(I)*x(I))
                  - sum(I,fixCost(I)*y(I));
    resLim(J).. sum(I, a(I,J)*x(I)) =l= resLimit(J);
    bigM(I).. x(I) =l= M * y(I);
MODEL Fixcharge /ALL/;
SOLVE Fixcharge USING mip MAXIMIZING profit;
```

9.4 Exercises

Exercise 9.1. For each $j = 1, 2, \cdots, 7$, let x_j be a binary variable which equals 1 if we invest in project j, and 0 if we do not. Consider the following statements:

F. We can invest in at most two projects, among projects 1, 2, and 3.

G. If we invest in project 4, we have to invest in project 5 as well.

H. We must invest in either project 6 or project 7, but not both.

Consider the following constraints:

I. $x_1 + x_2 + x_3 \geq 2$	II. $x_1 + x_2 + x_3 \leq 2$
III. $x_4 - x_5 \geq 0$	IV. $x_5 - x_4 \geq 0$
V. $x_6 + x_7 \geq 1$	VI. $x_6 + x_7 = 1$

Which of the following correctly matches the statements to their corresponding constraints?

(a) F-I, G-III, H-V	(b) F-II, G-III, H-VI
(c) F-II, G-IV, H-V	(d) F-II, G-IV, H-VI

Exercise 9.2. A manufacturer produces two types of products. The unit selling prices are \$2,000 for product 1 and \$5,000 for product 2. The production uses a

type of material. Each unit of product 1 uses 3 units of material, and each unit of product 2 uses 6 units of material. A total of 120 units of material are available. If any positive amount of product 1 needs to be produced, a setup cost of \$10,000 is incurred. If any positive amount of product 2 needs to be produced, a setup cost of \$20,000 is incurred.

Formulate a mixed integer program to maximize the total profit, by using binary variables to model the setup cost. Create a GAMS file *products.gms* to solve the problem. We assume that the manufacturer can produce fractional amounts, so it is not necessary to use integer variables to represent the amounts of products (but there are other integer variables in the formulation). The optimal value is \$80,000.

Exercise 9.3. The Lotus Point Condo Project will contain both homes and apartments. The site can accommodate up to 1000 dwelling units. (That is, the sum of numbers of homes and apartments cannot exceed 1000.) The project must also contain a recreation project: either a swimming-tennis complex or a sailboat marina, but not both. If a marina is built, then the number of homes in the project must be at least triple the number of apartments in the project.

A swimming-tennis complex will cost \$2.8 million to build, and a marina will cost \$1.2 million. The developers believe that each home will yield revenues with an NPV of \$46,000, and each apartment will yield revenues with an NPV of \$48,000. Each home (or apartment) costs \$40,000 to build.

Formulate an IP to help Lotus Point maximize profits. Create a GAMS file *condo.gms* to solve the problem. For simplicity, define the numbers of homes or apartments as nonnegative variables instead of integer variables (but there will be other integer variables in the formulation). The optimal value is \$5,300,000.

References