Section Notes 5

Integer Programming I

Applied Math / Engineering Sciences 121

Week of October 20, 2019

Goals for the week

- understand useful modeling tricks in formulating integer programs.
- practice with IP modeling.
- be able to use Big-M and logical constraints when modeling an integer program.
- be familiar with AMPL syntax for coding integer programs.
- (for those interested) understand how to model non-linear objectives by approximating piecewise linear objectives.

Contents

1 Integer Programming: A Recap

So far in this class we have been dealing with *Linear Programs*, where all the variables are continuous and can take on fractional values. As we have seen such programs are already incredibly useful. But there are many situations where fractional answers are not acceptable – and as we shall see restricting variables to integral values can provide tremendous additional modeling power.

If we restrict all of the variables to be integral, we have an *Integer Program*, or *IP*. If we allow some to be integral and others to be continuous, we have a *Mixed Integer Program*, or *MIP*. In both cases, we retain the restriction that the objective function and all of the constraints must be linear. In general, solving integer problems is significantly harder than their linear equivalence, as we shall soon see in lecture. For now let us assume that we have a means to solve the problems and concentrate on what can be done with the extra power that an IP provides.

Exercise 1

Optimal Capital Investment Solar Differential Inc. is starting up a business in manufacturing photovoltaic panels, and wants your help in determining how to best spend their startup capital. Their investors will supply an initial cash infusion of d dollars. They have five possible factory plans. Each factory $f \in F$, if built, will produce an annual profit of π_f and will cost β_f to construct. Please provide Solar Differential with an IP to determine which factories they should build.

End Exercise 1

2 Useful modeling tricks, part 1

2.1 Decision Conditions

Suppose you have binary decision variables $X_i, i \in N = \{1, ...n\}$. Let us consider how to ensure the following:

Exercise 2

Decisions are mutually exclusive (meaning no more than one of the decisions are active).

End Exercise 2

Exercise 3

No more than k of the decisions are active.

End Exercise 3

Exercise 4

At least k of the decisions must be active.

End Exercise 4

2.2 Implication Conditions

Exercise 5

Decision W is implied if at least one of the other decisions is active.

End Exercise 5

Exercise 6

Decision W is implied if all of the other decisions are active.

End Exercise 6

2.3 Binary Conditions on Continuous Variables

Exercise 7

Continuous variable Y is 0 if decision variable W is 0. You are given bounds on Y, $0 \leq Y \leq m$.

End Exercise 7

Exercise 8

Continuous variable Y must either be 0 or contained in the range $[l, h]$. (Hint: you will need a single binary variable W and two constraints.)

End Exercise 8

By completing the above exercises, you should now:

- be familiar with useful modeling tricks and how to apply them in an IP.
- understand how to model bound, implication, and binary conditions.

2.4 Preview of yet more

There are many more things you can do with integer variables, including:

- Either-Or constraint sets
- Conditional Constraints
- Elimination of Product terms

3 An IP modeling exercise: Milking the Juice

A n-th year SEAS Engineering PhD student (who shall be called N and not be named) is trying to come up with a compact circuit board that makes use of decentralized power to allow for better heat dispersion and less power linkers. N's idea is simple, but possibly effective: instead of a single power source, he will attach miniature power sources to components spread out across the board so that each power source serves the attached-to component and other neighboring power sources. For the idea to work, N will have to make sure that every component either has a power source attached, or has a neighbor with a power source attached. Furthermore, N must use as few power sources as possible to keep the size of the circuitry board and the heat dispersion low. The goal is to drive the circuit with the fewest number of power sources, so as to be able to then write his dissertation and finally graduate (in year $n + 1$).

Consider the following test circuit board:

```
aaaabbbbcccccc
ddeeeeffffffgg
ddddhhhhffffgg
ddddhhhhiiiigg
jjjkkkknnnoooop
jjjkkkknnnoooop
jjjmmmmmooooop
```

where each letter on the board denotes a particular component. Two components are neighbors if they touch in the horizontal or vertical directions.

Exercise 9

What are the variables? What is the objective?

End Exercise 9

Exercise 10

What are the constraints? (writing a specific constraint for the test circuit board is fine.)

End Exercise 10

Exercise 11

Put this together to formulate a generalize mathematical model as an integer linear program that finds the least number of power sources necessary for a circuit board to operate per the description above.

End Exercise 11

4 Modeling tricks, part 2

Integer programs gives us the ability to represent conditional and logical constraints. This is often done using the 'big-M' approach, where the M refers to a large constant that is often multiplied with a binary decision variable that allows a constraint to be trivially satisfied if the variable was 'triggered'. To be more concrete, consider the following exercises:

Exercise 12

Let X_i, $i \in N = \{1, \ldots, n\}$ be binary decision variables. Model the condition: if X_1 is true, then everything else is false.

End Exercise 12

Exercise 13

If you used a big-M in the previous exercise, explain how you would choose M.

End Exercise 13

As we have seen in lecture, we can also model more sophisticated conditional constraints, such as `either-or` constraints. Consider the following exercise:

Exercise 14

Let X_i, $i \in N = \{1, \ldots, n\}$ be binary decision variables. Model the condition: at least a third of the odd indexed variables are true, or at least half of the even indexed variables are true.

End Exercise 14

Exercise 15

Consider the above condition, but with the modification that either exactly a third of the odd indexed variables are true, or at least half of the even indexed variables are true.

End Exercise 15

5 A little bit of AMPL syntax

5.1 Integer and binary variables

Integer and binary variables are defined using the `integer` or `binary` keyword following a variable name. For example, one can define an integer variable Q in AMPL as follows:

```
var Q integer;
```

One can also define a vector of integer/binary variables as before with continuous variables. For example, given a set P, we can define binary variables $X_p, p \in P$ as follows:

```
var X{P} binary;
```

5.2 Set operations

Consider creating two sets in AMPL:

```
set N;
set M;
```

Our data file might specify these sets as:

```
set N := 1 2 4 6;
set M := 2 5 6 7;
```

We can take the intersection, union and difference of these two sets with:

```
set I := N intersect M;
set U := N union M;
set D := N diff M;
```

Which will result in the sets:

```
I:  2 6;
U:  1 2 4 5 6 7;
D:  1 4;
```

`intersect`, `union` and `diff` will work with pairs and tuples as well.

6 Piecewise Linear Objectives (Optional)

We can use boolean variables to approximate piecewise linear objectives (and thereby model non-linear objectives). Here we present a general solution. If you know your objective is convex or concave, you can use a simpler set of constraints.

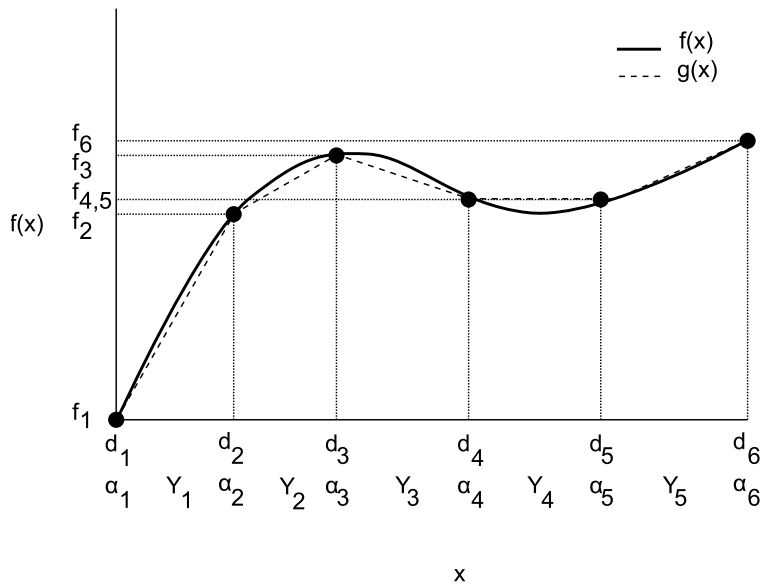

Figure 1: Piecewise Linear Approximation

Suppose we have a nonlinear objective in continuous variable X, $f(X)$. We can approximate this, by establishing a set of r control points $(d_1, f_1), ..., (d_r, f_r)$ where the d_i are in the same domain as X, and the f_i are in the same range as $f(X)$. (Think of $f_i = f(d_i)$, though in general we may not choose to have the control points live on the curve of the original function). This approximation can be seen in figure 1.

Now using r continuous mixing variables $\alpha_i \geq 0$, we define:

$$X = \sum_{i=1}^{r} \alpha_i d_i$$

and

$$f(X) \approx g(X) = \sum_{i=1}^{r} \alpha_i f_i$$

Thus we are trying to use the α_i to establish convex combinations of the mixing points, which means we need to restrict the sum of the mixing variables to 1:

$$\sum_{i=1}^{r} \alpha_i = 1$$

But this isn't enough. We also need to makes sure that no more than *two adjacent* $\alpha_i > 0$. We can establish this with a set of $r - 1$ binary variables γ_i:

$$\alpha_1 \leq \gamma_1$$

$$\alpha_i \leq \gamma_{i-1} + \gamma_i, i = 2, ...r-1$$
$$\alpha_r \leq \gamma_{r-1}$$
$$\sum_{i=1}^{r-1} \gamma_i = 1$$

The last constraint ensures that for some integer $z : 1 \leq z \leq r-1$, we have $\gamma_z = 1$ and $\gamma_i = 0 \; \forall \; i \neq z$. These γ in the former three constraints ensure that $\alpha_z \leq 1$, $\alpha_{z+1} \leq 1$, and $\alpha_i = 0 \; \forall \; i \notin \{z, z+1\}$, as required.

6.1 Solar Differential, continued

Exercise 16

Determining Production Level. Solar Differential is very pleased with your work, but they are worried that the profit constants, π_f, they told you may not have been accurate. So they have hired a market research firm to calculate a price p at which they should be able to sell their panels. They have also modeled their factories more carefully, and now have a set of non-linear functions $c_f(q)$ which indicate the cost of producing q panels at factory f (note that $c_f(0) = 0 \; \forall \; f \in F$). They have also provided bounds $l_f \leq q_f \leq u_f$ on the production levels of each possible factory. Modify your formulation to maximize profit in this setting.

End Exercise 16

Exercise 17

Now let's attempt to code some of the Solar Differential model into AMPL. Please define the necessary sets, parameters, variables and objective. Don't worry about coding the constraints.

End Exercise 17

Exercise 18

Modeling Price Note that the above model assumes sufficient demand at the quoted price. There is an ideal price for which this is true. For any price above this price, our profits will fall due to excess inventory. For any price below this point, our profits will fall due to charging less than the market will support. Solar Differential is banking that the market research firm they hired got this right.

Can you think of what might be done to explicitly capture this effect in the model? You do not need to actually update the model to handle this case.

End Exercise 18

By completing the above exercises, you should now understand how to model non-linear objectives by approximating piecewise linear objectives.

7 Solutions

Solution 1

We define a set of binary decision variables X_f for which factories to make.

$$\begin{aligned} \max \quad & \sum_{f \in F} \pi_f X_f \\ \text{s.t.} \quad & \sum_{f \in F} \beta_f X_f \leq d \\ & X_f \in \{0, 1\} \; \forall \; f \in F \end{aligned}$$

End Solution 1

Solution 2

A single constraint:

$$\sum_{i \in N} X_i \leq 1$$

End Solution 2

Solution 3

$$\sum_{i \in N} X_i \leq k$$

End Solution 3

Solution 4

$$\sum_{i \in N} X_i \geq k$$

End Solution 4

Solution 5

$$\sum_{i \in N} X_i \leq |N|W$$

End Solution 5

Solution 6

$$\sum_{i \in N} X_i \leq |N| - 1 + W$$

End Solution 6

Solution 7

$$Y \leq mW$$

End Solution 7

Solution 8

$$Y \leq hW$$
$$Y \geq lW$$

Here $W = 0$ implies $Y = 0$, and $W = 1$ implies $l \leq Y \leq h$

End Solution 8

Solution 9

Let x_z be a binary variable such that $x_z = 1$ if we place a power source on component $z \in Z$ and 0 otherwise. We aim to minimize $\sum_{z \in Z} x_z$ over all components z.

End Solution 9

Solution 10

Since every component must have a power source or border a component with a power source, we must ensure that in the neighborhood of every component, there exists at least one power source. For the component in the upper left hand corner ("a"), we require at least one of x_a, x_b, x_d, and x_e to be 1. We can write this down as follows:

$$x_a + x_b + x_d + x_e \geq 1$$

We can write such a constraint down for every component, completing our binary integer program. **Note:** Notice that per our problem statement, where on the component the power source is placed is irrelevant to satisfying the neighborhood conditions.

End Solution 10

Solution 11

Consider binary parameters a_{ij} such that $a_{ij} = 1$ if two components i and j are neighbors or $i = j$. We have the following integer program:

$$\begin{aligned} \min \quad & \sum_{z \in Z} x_z \\ \text{s.t.} \quad & \sum_{j \in Z} a_{ij} x_j \geq 1 \quad \forall\, i \in Z \\ & x_z \in \{0, 1\} \quad \forall\, z \in Z \end{aligned}$$

End Solution 11

Solution 12

$$\sum_{i \in N} X_i \leq 1 + (1 - X_1)M$$

End Solution 12

Solution 13

We can choose $M \geq n - 2$. Next week we will see that it is preferable to choose M as tight as we can, which in this case is $M = n - 2$.

End Solution 13

Solution 14

$$\begin{aligned}
&\sum_{i \in N, i\%2=0} X_i \geq \frac{n_{even}}{2} - M \cdot (1 - \alpha_A) \\
&\sum_{i \in N, i\%2=1} X_i \geq \frac{n_{odd}}{3} - M \cdot (1 - \alpha_B) \\
&\alpha_A + \alpha_B \geq 1 \\
&\alpha_A, \alpha_B \in \{0, 1\}
\end{aligned}$$

where M is a sufficiently large constant.

The constraint $\alpha_A + \alpha_B \geq 1$ makes sure that at least one of the events A and B has to be activated.

End Solution 14

Solution 15

We can modify the second constraint as follows:

$$\begin{aligned}
&\sum_{i \in N, i\%2=1} X_i \geq \frac{n_{odd}}{3} - M \cdot (1 - \alpha_B) \\
&\sum_{i \in N, i\%2=1} X_i \leq \frac{n_{odd}}{3} + M \cdot (1 - \alpha_B)
\end{aligned}$$

End Solution 15

Solution 16

First we divide our quantity space into a grid with R points, $q_1, ..., q_r$. In general we may want to perform some sort of regression to find the corresponding production costs, but for simplicity we'll just define them as $c_{fi} = c_f(q_i)\ \forall\ i \in R, f \in F$. Lets let Q be the number of panels we decide to make, and Q_f be the production level for each factory f. Note that we are using the same grid

to sample from all $|F|$ non linear cost functions. We could choose a parameterization that uses a different grid for each function.

Next, we'll introduce a set of mixing variables for each factory, α_{fi}, and a set of binary decision variables for choosing the portion of the linear function, Y_{fi}. Note the two indices, here, as we are actually modeling $|F|$ non linear functions.

Now we're in a position to construct our model:

$$
\begin{array}{lll}
\max & pQ - \sum_{f \in F, i \in R} \alpha_{fi} c_{fi} & & \text{max profit} \\
\text{s.t.} & \sum_{i \in R} \alpha_{fi} = 1 & \forall\ f \in F & \text{mix convex} \\
& \alpha_{f1} \leq Y_{f1} & \forall\ f \in F & \text{only adj.} \neq 0 \\
& \alpha_{fi} \leq Y_{f,i-1} + Y_{fi} & \forall\ i \in R - \{1, r\}, f \in F & \text{only adj.} \neq 0 \\
& \alpha_{fr} \leq Y_{f,r-1} & \forall\ f \in F & \text{only adj.} \neq 0 \\
& \sum_{i \in R - \{r\}} Y_{fi} = 1 & \forall\ f \in F & \text{only 1 segment} \\
& Q_f = \sum_{i \in R} \alpha_{fi} q_i & \forall\ f \in F & \text{quantity prod. at } f \\
& Q = \sum_{f \in F} Q_f & & \text{total quantity} \\
& Q_f \geq l_f X_f & \forall\ f \in F & \text{prod. bounds} \\
& Q_f \leq u_f X_f & \forall\ f \in F & \text{prod. bounds} \\
& \sum_{f \in F} \beta_f X_f \leq d & & \text{capital budget} \\
& & & \\
& \alpha_{fi} \geq 0 & \forall\ f \in F, i \in R & \text{mixture} \\
& Y_{fi} \in \{0, 1\} & \forall\ f \in F, i \in R - \{r\} & \text{segment} \\
& Q, Q_f \text{ integer} & \forall\ f \in F & \text{quantity} \\
& X_f \in \{0, 1\} & \forall\ f \in F & \text{construct}
\end{array}
$$

End Solution 16

Solution 17

Here we specify the ampl model file:

```
set FACTORY ordered;                      # The set of factories
param num_control >= 1;                   # The number of control points
set CONTROL = 1..num_control;             # The set of control points
set SEGMENT = CONTROL diff {num_control}; # The number of segements

param price;                          # Unit price for a solar panel
param quantity{FACTORY, CONTROL} >=0;# Quantity at control point
param cost{FACTORY, CONTROL} >= 0;   # Profit for factory at control point
param budget;                         # Captial budget
param construction{FACTORY};          # Cost of building a factory
param min_prod{FACTORY};              # Minimum production
param max_prod{FACTORY};              # Maximum production

var Alpha{FACTORY, CONTROL}>=0, <=1; # mixing factor
var Y{FACTORY, SEGMENT} binary;      # Active segment
var Qt integer;                      # total quantity
var Q{FACTORY} integer;              # quantity at factory
var X{FACTORY} binary;               # should we build factory

maximize Profit:
         price*Qt - sum{f in FACTORY, i in CONTROL} cost[f,i]*Alpha[f,i];

subject to MixConvex {f in FACTORY}: sum{i in CONTROL} Alpha[f,i] = 1;
subject to LowEndAdj {f in FACTORY}: Alpha[f,1] <= Y[f,1];
subject to   OnlyAdj {f in FACTORY, i in SEGMENT diff {1}}:
                     Alpha[f,i] <= Y[f,i-1] + Y[f,i];
subject to HighEndAdj{f in FACTORY}: Alpha[f,num_control]<=Y[f,num_control-1];
subject to OneSegment{f in FACTORY}: sum{i in SEGMENT} Y[f,i] <= 1;
subject to FactoryQuantity{f in FACTORY}:
                     Q[f] = sum{i in CONTROL}quantity[f,i]*Alpha[f,i];
subject to TotalQuantity: Qt = sum{f in FACTORY} Q[f];
subject to LowProdBound {f in FACTORY}: Q[f] >= min_prod[f]*X[f];
subject to HighProdBound{f in FACTORY}: Q[f] <= max_prod[f]*X[f];
subject to CaptialBudget: sum{f in FACTORY} construction[f]*X[f] <= budget;
```

Next we define the data file:

```
set FACTORY := 1 2 3 4 5;
param num_control := 5;

param price := 150;

param quantity:  1       2       3       4      5 :=
1                0      10     100     200    400
2                0      20     200     400    800
3                0      40     400     800   1600
4                0      80     800    1600   3200
5                0     160    1600    3200   6400;

param cost:      1       2       3       4       5 :=
1                0    2000   15000   27000   80000
```

```
2                0    3800   28500   51300  144000
3                0    7200   54000   97200  256000
4                0   13600  102000  183600  448000
5                0   25600  192000  345600  768000;

param budget := 10;                                # in millions

param: construction min_prod max_prod :=
1                 1        0      370
2                 2       20      770
3                 3       40     1500
4                 4       60     3100
5                 5       80     6100;
```

Running these files we get the following output:

```
CPLEX 12.3.0.1: optimal integer solution; objective 246000
31 MIP simplex iterations
0 branch-and-bound nodes
absmipgap = 1.92085e-09, relmipgap = 7.80834e-15
Qt = 7900

:    Q     X    :=
1    200   1
2      0   0
3      0   0
4   1600   1
5   6100   1
;

:     Y    Alpha     :=
1 1   0   0
1 2   0   0
1 3   0   0
1 4   1   1
1 5   .   0
2 1   1   1
2 2   0   0
2 3   0   0
2 4   0   0
2 5   .   0
3 1   1   1
3 2   0   0
3 3   0   0
3 4   0   0
3 5   .   0
4 1   0   0
4 2   0   0
4 3   1   0
4 4   0   1
4 5   .   0
5 1   0   0
5 2   0   0
5 3   0   0
```

```
5 4   1   0.09375
5 5   .   0.90625
;
```

In this problem as long as the production cost is greater than the sale price, producing the item will be useful. However we still want to allocate our investment dollars to the factories that will make the best use of them. In this case because the production costs at the 4th control point are generally the most favorable, the data favors running the factories at that capacity. However, due to the high volume of production at the largest factory, and the fact that its production cost remains above the sale price all the way to the 5th data point, the optimal solution is to run that factory at capacity (which is a fraction of the way to the 5th data point).

End Solution 17

Solution 18

The pQ term in the objective formulation represents the revenue generated as a function of quantity: $r(Q)$. Our previous formulation represented this a linear function, but in reality it is a non-linear concave function. If our market research firm provides us with this data approximating this non-linear function, we can modify our model to use yet one more non-linear objective – this time revenue.

Practically, what the market research firm typically thinks of is a demand curve: the quantity the market demands at a given price, or $d(Q)$. Revenue is therefore $Qd(Q)$. This function is not linear in Q, even if $d(Q)$ were to be linear. However, we can get around this by sampling from $Qd(Q)$ instead of $d(Q)$ when we formulate our piecewise linear approximation.

End Solution 18