

Section Notes 6  
**Integer Programming II**

Applied Math / Engineering Sciences 121

Week of October 28, 2019

**Goals for the week**

- understand IP relaxations
- be able to determine the relative strength of formulations
- understand the branch and bound method for solving IPs
- practice using the branch and bound method
- expand your knowledge of big ‘M’s and logical constraints.

**Contents**

<b>1 Integer Programming: Guidelines for Strong Formulations</b>	<b>2</b>
<b>2 Branch and Bound</b>	<b>4</b>
<b>3 Advanced Modeling Techniques</b>	<b>12</b>
<b>4 OPTIONAL: The Minimum Spanning Tree Problem</b>	<b>13</b>
<b>5 Solutions</b>	<b>18</b>

# 1 Integer Programming: Guidelines for Strong Formulations

## 1.1 Relaxations

Given an integer program (IP):

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b \\ & x \geq 0, \text{ integer} \end{aligned}$$

we can write it alternatively as:

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & x \in S \subseteq Z^n \end{aligned}$$

The set  $S$  defines the set of feasible solutions to the IP.

---

### Exercise 1

---

#### Relaxations.

1. Give the general form of a relaxation for the IP, i.e., the relaxed program (RP). Note that we can relax a) the cost function and b) the set of feasible solutions.
2. Denote the objective value of the solution to the IP as  $z^*$  and the objective value of the solution to the RP as  $\bar{z}$ . What is the relation between  $z^*$  and  $\bar{z}$ ?
3. What do we know about the IP if the RP is feasible?
4. What do we know about the IP if the RP is infeasible?
5. Assume  $x^*$  is an optimal solution for the RP. Give sufficient conditions on  $x^*$  such that  $x^*$  is also an optimal solution for the IP.
6. What is the linear programming relaxation (LPR) of the IP?
7. Assume  $x^*$  is an optimal solution for the LPR. Give necessary and sufficient conditions on  $x^*$  such that  $x^*$  is also an optimal solution for the IP.

---

### End Exercise 1

---

By completing the above exercises, you should now understand the relationship between an integer problem and its (linear program) relaxation.

## 1.2 Strong Formulations

Before we can talk about different formulations for an IP we need some more definitions/results:

1. Remember the set of feasible solutions of an LP is a **polytope**. We can define the polytope  $P = \{x \in R^n : Ax \leq b, x \geq 0\}$ .
2. Let  $S$  be the set of feasible integer solutions of an IP. Then any polytope  $P$  is a **valid formulation** for the IP if  $P \cap Z^n = S$  (see following figure).
3. Consider two formulations  $P_1$  and  $P_2$  for the same IP with  $P_1 \subseteq P_2$ . Then we know that the optimal objective values corresponding to the linear program relaxations of  $P_1$  and  $P_2$ , i.e.  $z_1^{LP}$  and  $z_2^{LP}$  have the following relation:  $z_1^{LP} \leq z_2^{LP}$ .
4. If  $P_1 \subset P_2$  we say that  $P_1$  is a **stronger formulation** than  $P_2$ .

---

### Exercise 2

---

**Strong formulations.** Remember figure 1 from lecture, which displays different formulations (polyhedra) for the same IP. Think of the Branch and Bound algorithm for solving IPs. Use this to give an intuitive explanation for why a stronger formulation is better for the Branch and Bound algorithm.

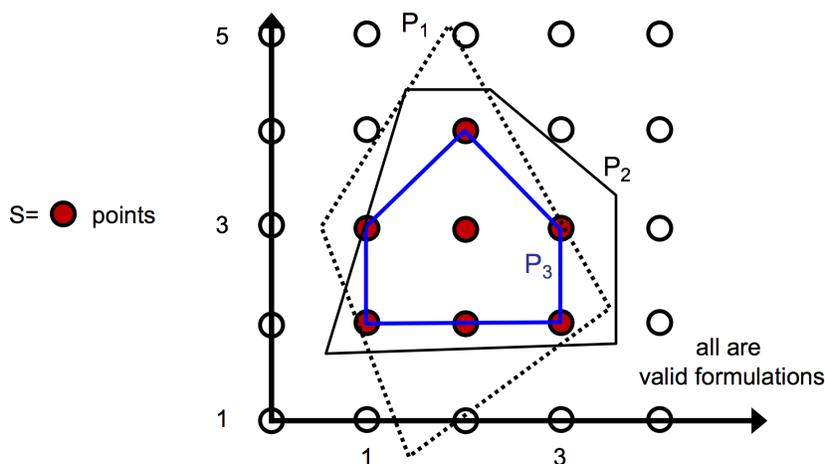


Figure 1: IP Relaxations

---

**End Exercise 2**

---

## 2 Branch and Bound

(Inspired by L.A. Wolsey, Chapter 7). A very successful algorithmic strategy is called *divide and conquer*. In computer science one of the most famous representatives of this class of algorithms is the quicksort algorithm, which divides an unsorted list of numbers again and again until the task is trivial because only two numbers are left in the smallest lists. After that the solutions of the small subproblems are assembled to form the solution of the large problem.

*Branch and bound* is an algorithm that uses this strategy to solve the (usually) computationally expensive integer programs and mixed integer programs. The feasible region of a relaxed problem is split into smaller regions, and the best solution from these subregions are compared to determine the optimal solution.

### 2.1 Using LP Relaxations in Branch and Bound

Consider the following integer LP.

$$\begin{aligned}
 \max \quad & 4x_1 - x_2 \\
 \text{s.t.} \quad & 7x_1 - 2x_2 \leq 14 \\
 & x_2 \leq 3 \\
 & 2x_1 - 2x_2 \leq 3
 \end{aligned}$$

$$x_1, x_2 \geq 0, \text{ integer}$$

---

**Exercise 3**

---

**LP Relaxation.** Assuming that we ignore the integer constraints for the moment, would the optimal objective value be larger (or equal) than for the integer program?

---

**End Exercise 3**

---

We can construct two subproblems by adding the constraint  $x_1 \leq 2$  to get the first one and  $x_1 \geq 3$  to get the second one. Then the two new subproblems have the following feasible integer regions:

$$\begin{aligned} S_1 &= S \cap \{x : x_1 \leq 2\} \\ S_2 &= S \cap \{x : x_1 \geq 3\} \end{aligned}$$

Let  $z_1$  denote the optimal solution of the first LPR and  $z_2$  the solution of the second LPR and finally  $z^*$  the optimal solution of the integer problem.

---

**Exercise 4**

---

**LP Relaxation.** Without solving the linear program, is  $z_1 \geq z^*$ ? (Answers include: Yes, No, can't tell).

---

**End Exercise 4**

---

---

**Exercise 5**

---

**LP Relaxation.** Without solving the linear program, is  $\max(z_1, z_2) \geq z^*$ ? (Yes, No, can't tell).

---

**End Exercise 5**

---

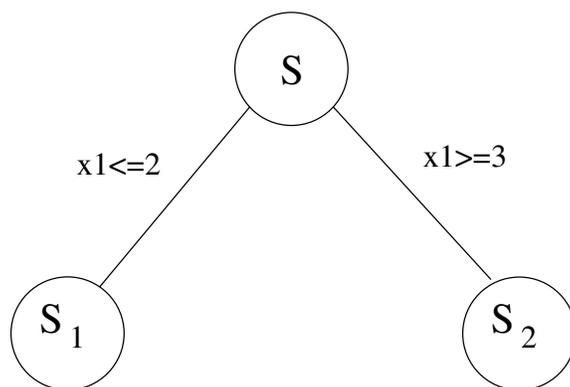
## 2.2 Branch and Bound: Idea and Algorithm

By introducing additional constraints into the relaxation of the original problem, the branch and bound algorithm splits the feasible region into smaller parts, which still contain all feasible solutions of the integer problem. These splits can be conveniently represented in the form of a binary tree, where the nodes represent different subsets of the initial feasible region of the relaxed problem.

But the splitting alone is not sufficient to get an useful algorithm. Efficient pruning of the branches of the trees is required to quickly find solutions.

Three reasons allow us to remove items from the list of nodes that have to be examined more closely (cf. L.A. Wolsey):

1. Pruning by integrality:  $z^t = \{\max cx : x \in S_t\}$  has been solved



2. Pruning by bound:  $\bar{z}^t \leq \underline{z}$ .
3. Pruning by infeasibility:  $S_t = \emptyset$ .

The variable  $\bar{z}^t$  denotes the optimum of a certain node of the tree, i.e. the optimal feasible solution over all nodes descending from that node. Furthermore,  $\underline{z}$  denotes the best lower bound that has been found so far.

Putting these things together, we get the following algorithm for solving an integer program (simplified from L.A. Wolsey, Fig.7.10). Here the index  $i$  indicates the subproblem.

1. *Initialization* Put initial problem  $S$  with formulation  $P$  into list, lower bound  $\underline{z} = -\infty$ .
2. *Termination* If no more problems in list, terminate.
3. *Choosing a node* Chose problem  $S^i$  with formulation  $P^i$ .
4. *Optimizing* Solve LP relaxation over  $P^i$ , dual bound  $\bar{z}^i = LPvalue$ .
5. *Pruning*
  - If  $P^i$  infeasible, prune
  - If  $\bar{z}^i \leq \underline{z}$ , prune by bound
  - If solution is integer, update best feasible solution  $\underline{z} := \max(\underline{z}, \bar{z}^i)$  and prune by integrality.
6. *Branching* If this subproblem is not pruned, then create two new subproblems  $S_1^i$  and  $S_2^i$  with formulations  $P_1^i$  and  $P_2^i$ .
7. Return to step 2.

### 2.3 Using Dual Simplex in Branch and Bound

In the context of the Branch and Bound algorithm, the properties of the dual simplex method are particularly useful. As we have seen in the previous sections, at each branch of the tree, the solution of two extended versions of the original linear program is required. These subproblems can be efficiently solved by adding the new constraints to the current tableau and performing several dual simplex iterations, until the new optimum is found.

Adding slack variables  $x_3, x_4, x_5$  to our problem and doing several pivot operations we obtain the following tableau:

$$\left[ \begin{array}{c|cccccc} \text{vars} & x_1 & x_2 & x_3 & x_4 & x_5 & RHS \\ \hline z & 0 & 0 & 4/7 & 1/7 & 0 & 59/7 \\ x_1 & 1 & 0 & 1/7 & 2/7 & 0 & 20/7 \\ x_2 & 0 & 1 & 0 & 1 & 0 & 3 \\ x_5 & 0 & 0 & -2/7 & 10/7 & 1 & 23/7 \end{array} \right]$$

---

#### Exercise 6

---

**Dual Simplex.** The branch and bound algorithm introduces the additional constraint  $x_1 \leq 2$ . Add this constraint to the tableau and do one dual simplex iteration.

---

#### End Exercise 6

---

## 2.4 Branch and Bound: A Complete Example

(From Wolsey, Chapter 7, Exercise 2.) Consider the two variable integer program:

$$\begin{array}{ll} \max & 9x_1 + 5x_2 \\ \text{s.t.} & 4x_1 + 9x_2 \leq 35 \\ & x_1 \leq 6 \\ & x_1 - 3x_2 \geq 1 \\ & 3x_1 + 2x_2 \leq 19 \\ & x_1, x_2 \geq 0, \text{ integer} \end{array}$$

---

### Exercise 7

---

**Graphical Solution.** Solve this integer problem graphically

---

### End Exercise 7

---

If variables  $x_1$  and  $x_2$  in the problem are not limited to integer values, the problem has the following optimal solution:  $x_1 = 6.0$ ,  $x_2 = 0.5$  with an objective of 56.5.

Since the primal-simplex iterations have been exercised several times, you do not have to do these in this exercise. Instead use the following tables, which provide the optimal objective value for the linear program that has been extended by the specified additional constraint (you don't need to generate this table for your homework or anything!)

(Note also: The tables are rounded to the nearest 0.1, so  $\frac{1}{3}$  becomes 0.3 and  $\frac{2}{3}$  becomes 0.7. If you would like, you can perform simplex and prove these are the correct values).

$x_1 \leq$	10	9	8	7	6	5	4	3	2	1
obj.	56.5	56.5	56.5	56.5	56.5	51.7	41.0	30.3	19.7	9.0
$x_1$	6	6	6	6	6	5	4	3	2	1
$x_2$	0.5	0.5	0.5	0.5	0.5	1.3	1	0.7	0.3	0

$x_1 \geq$	10	9	8	7	6	5	4	3	2	1
obj.	inf	inf	inf	inf	56.5	56.5	56.5	56.5	56.5	56.5
$x_1$					6	6	6	6	6	6
$x_2$					0.5	0.5	0.5	0.5	0.5	0.5

$x_2 \leq$	3	2	1	0
obj.	56.5	56.5	56.5	54.0
$x_1$	6	6	6	6
$x_2$	0.5	0.5	0.5	0

$x_2 \geq$	3	2	1	0
obj.	inf	inf	56.0	56.5
$x_1$			5.7	6
$x_2$			1	0.5

$x_1 \leq$		7		6		5		4		3
$x_2 \leq$										
3	56.5	(6,0.5)	56.5	(6,0.5)	51.7	(5,1.3)	41	(4,1)	30.3	(3,0.7)
2	56.5	(6,0.5)	56.5	(6,0.5)	51.7	(5,1.3)	41	(4,1)	30.3	(3,0.7)
1	56.5	(6,0.5)	56.5	(6,0.5)	50.0	(5,1)	41	(4,1)	30.3	(3,0.7)
0	54	(6,0)	54	(6,0)	45	(5,0)	36	(4,0)	27	(3,0)

$x_1 \geq$		7		6		5		4		3	
$x_2 \leq$											
3	inf	56.5	(6,0.5)	56.5	(6,0.5)	56.5	(6,0.5)	56.5	(6,0.5)	56.5	(6,0.5)
2	inf	56.5	(6,0.5)	56.5	(6,0.5)	56.5	(6,0.5)	56.5	(6,0.5)	56.5	(6,0.5)
1	inf	56.5	(6,0.5)	56.5	(6,0.5)	56.5	(6,0.5)	56.5	(6,0.5)	56.5	(6,0.5)
0	inf	54	(6,0)	54	(6,0)	54	(6,0)	54	(6,0)	54	(6,0)

$x_1 \leq$	7	6	5	4	3
$x_2 \geq$					
4	inf	inf	inf	inf	inf
3	inf	inf	inf	inf	inf
2	inf	inf	inf	inf	inf
1	56.0 (5.7,1)	56.0 (5.7,1)	51.7 (5,1.3)	41 (4,1)	inf
0	56.5 (6,0.5)	56.5 (6,0.5)	51.7 (5,1.3)	41 (4,1)	30.3 (3,0.7)
$x_1 \geq$	7	6	5	4	3
$x_2 \geq$					
4	inf	inf	inf	inf	inf
3	inf	inf	inf	inf	inf
2	inf	inf	inf	inf	inf
1	inf	inf	56.0 (5.7,1)	56.0 (5.7,1)	56.0 (5.7,1)
0	inf	56.5 (6,0.5)	56.5 (6,0.5)	56.5 (6,0.5)	56.5 (6,0.5)

---

**Exercise 8**

---

**Branch and Bound.** Solve this integer problem by branch-and-bound. Write down the new linear programs, look for the objective value in the tables and construct the corresponding branch and bound tree. After each step write down the current upper and lower bound of the solution.

---

**End Exercise 8**

---

By completing the above exercises, you should now:

- be comfortable with the branch and bound algorithm
- be able to use dual simplex in branch and bound

### 3 Advanced Modeling Techniques

#### 3.1 Big-M

Remember the “candidates meeting the press” problem from the last homework. We let  $A_{i,j,k}$  be a binary variable on whether candidate  $i$  attends reporter  $j$ ’s show at time  $k$ . Candidate  $i$ ’s availability at time  $k$  is denoted as a binary parameter  $c_{i,k}$ .

---

#### Exercise 9

---

1. Now assume that Mr. President is also coming to the press conference. When he is giving an interview, it is broadcasted live and all the other candidates are watching it, thus they cannot be scheduled for an interview at the time that Mr. President is giving one. Formulate a constraint that captures this. Assume that Mr. President is in the set of speakers, and *obama* denotes his index.
2. You probably used a big  $M$  for the previous part. Explain how you would choose  $M$ . What’s the best you can do?

---

#### End Exercise 9

---

#### 3.2 Logical Constraints

Now, assume that the manager of the press conference is short of staff and one of his most important employees needs to take some time off either in the morning (time slots 1 and

2) or in the afternoon (time slots 3 and 4), but he doesn't care when. The consequence is that when that employee is gone, the manager can have at most 10 interviews scheduled for that part of the day.

---

**Exercise 10**

---

1. Model the new logical constraint.
2. You probably used a big  $M$  for the previous part. Explain how you would choose  $M$ . What's the best you can do?

---

**End Exercise 10**

---

By completing the above exercises, you should now:

- be able to use big Ms and write logical constraints

## 4 OPTIONAL: The Minimum Spanning Tree Problem

Let's now take a look at an example to see how different formulations for an IP can be stronger or weaker.

Let  $G = (N, E)$  be an undirected graph with node set  $N$  ( $|N| = n$ ) and edge set  $E$  ( $|E| = m$ ). Recall that an edge in an undirected graph is an unordered pair  $e = \{i, j\}$  of distinct nodes in  $N$ . Every edge  $e \in E$  has an associated cost  $c_e$ . Let  $E_1 \subseteq E$ . If  $(N, E_1)$  forms a tree and the edges touch all nodes this is called a spanning tree for  $G$ . Note that the edges have to be connected?spanning trees cannot have disjoint parts. The cost of a tree is simply the sum of the costs of the edges in the tree. The minimum spanning tree (MST) problem is the problem of finding a spanning tree of minimum cost.

Tree optimization problems arise in the design of transportation, communication, and computer networks, since at the very least such networks should be connected. Our goal in this example is to illustrate the effectiveness of alternative formulations to learn new principles for deriving strong formulations.

### 4.1 Subtour Elimination formulation

---

**Exercise 11**

---

In this exercise you are asked to find an integer programming formulation of the minimum spanning tree problem, in particular the subtour elimination formulation.

1. What are our decision variables in this problem?
2. What is the objective in this problem?
3. How many edges should a spanning tree have? Formulate a constraint that captures that.
4. The chosen edges should not contain a cycle. Try to formulate a subtour elimination constraint that eliminates all subtours (cycles). You might find it useful to use the following definition: for any set  $S \subset N$ , we define  $E(S) = \{\{i, j\} \in E \mid i, j \in S\}$ .
5. Formulate the whole IP.
6. How many constraints (aside from the integrality constraints) does your formulation have?

---

**End Exercise 11**

---

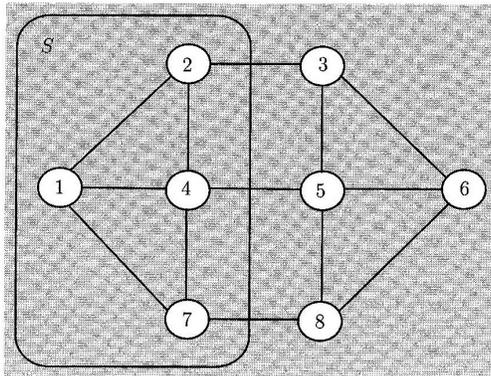
## 4.2 Cutset Formulation

---

**Exercise 12**

---

In this exercise you are asked to find an integer programming formulation of the minimum spanning tree problem, in particular the cutset formulation: The subtour elimination constraint used the definition of a tree as a subgraph containing  $n - 1$  edges and no cycles.



**Figure 10.5:** Let  $S = \{1, 2, 4, 7\}$ . Then,  $\delta(S) = \{\{2, 3\}, \{4, 5\}, \{7, 8\}\}$ , and  $E(S) = \{\{1, 2\}, \{1, 4\}, \{2, 4\}, \{4, 7\}, \{1, 7\}\}$ .

Figure 2: Cutset

Now forget about that formulation from the previous IP. Instead use an alternative, but equivalent definition, a tree is a connected graph containing  $n - 1$  edges.

1. Think about how you can express the connectivity requirement. You might find the following definition helpful: Given a subset  $S$  of  $N$ , we define the cutset  $\delta(S)$  by  $\delta(S) = \{\{i, j\} \in E \mid i \in S, j \notin S\}$  (see figure 2. Note that  $\delta(\{i\})$  is the set of edges incident to  $i$ ).
2. Formulate the whole IP with the new cutset constraint (connectivity).
3. How many constraints (aside from the integrality constraints) does your formulation have?

---

**End Exercise 12**

---

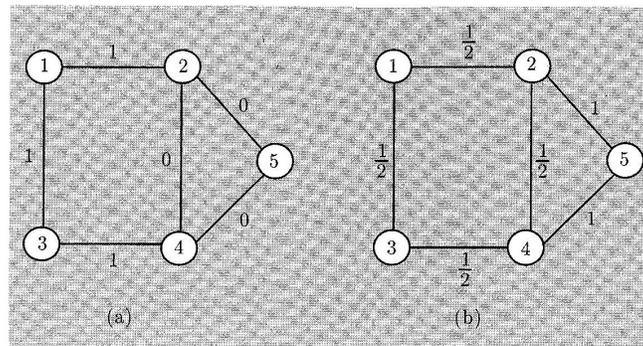
### 4.3 Which formulation is better?

We denote the feasible set of the linear programming relaxation of the two formulation by  $P_{sub}$  and  $P_{cut}$ . Both formulations have an exponential number of constraints. However, we can show that the subtour elimination formulation is stronger than the cutset formulation.

The following properties hold:

1. We have  $P_{sub} \subseteq P_{cut}$ , and there exist examples for which the inclusion is strict.
2. The polyhedron  $P_{cut}$  can have fractional extreme points.
3.  $P_{sub} = CH(T)$ , i.e., the polyhedron  $P_{sub}$  is a representation of the “convex hull” of the set of vectors corresponding to spanning trees. Thus, this is smallest possible polyhedron we can find for the MST problem.

Formal proofs of these properties are too long for these sections notes. Do them at home if you like. However, we can give some intuition for parts of the proof. Consider the graph given in figure 3.



**Figure 10.6:** An example of a graph, in which a minimum spanning tree has cost 2, while the cost of an optimal solution over  $P_{cut}$  is  $3/2$ . (a) The cost coefficients. (b) An optimal solution  $x^*$  over  $P_{cut}$ .

Figure 3: Fractional Solution for  $P_{cut}$

The solution  $x^*$  shown in part (b) belongs to  $P_{cut}$ , but it does not belong to  $P_{sub}$ , since the edges in  $E(S)$  for  $S = \{2, 4, 5\}$  have total weight  $5/2$ , while the constraints defining  $P_{sub}$  dictate that the weight should be less than or equal to  $3-1=2$ . The example shows that the inclusion  $P_{sub} \subseteq P_{cut}$  may be strict. The same example also shows that  $P_{cut}$  may have fractional extreme points. The unique optimal solution to  $P_{cut}$  is the fractional solution  $x^*$  shown in the figure part (b) with a cost of  $3/2$ , while the optimal integer solution has

a cost of 2.

Thus, according to the principle regarding strong formulations, the subtour elimination formulation is stronger than the cutset formulation even though both formulations have an exponential number of constraints.

A last note on this problem: There exists an algorithm that solves the MST problem in  $n \log n$  time (google it). In practice, IPs are not used to solve the MST problem. Nevertheless, the problem is a good example for studying formulation tightness for IPs.

By completing the above exercises, you should now:

- be able to identify and write minimum spanning tree IPs (both subtour elimination and cutset formulations)
- understand why the subtour elimination formulation is stronger.

## 5 Solutions

---

### Solution 1

---

1. The relaxed program (RP) is given by:

$$\begin{aligned} \max \quad & \bar{c}^T x \\ \text{s.t.} \quad & x \in T \subseteq R^n \end{aligned}$$

and  $\bar{c}^T x \geq c^T x \quad \forall x \in S$  and  $S \subseteq T$ .

2. We know that  $\bar{z} \geq z^*$ .
3. We cannot say anything about the IP in that case.
4. If the RP is infeasible, the IP must also be infeasible.
5. If (1)  $x^* \in S$  and 2)  $\bar{c}^T x^* = c^T x^*$ , then we know that  $x^*$  is also an optimal solution for the IP. **Note:** while (1) is also a necessary condition, (2) is not necessary in general.
6. The linear programming relaxation (LPR) is:

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & x \in P \subseteq R^n \end{aligned}$$

where we require that  $P = \{x \in R^n : Ax \leq b, x \geq 0\}$  and  $S = P \cap Z^n$ . In other words, the LPR is:

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b \\ & x \geq 0 \end{aligned}$$

7. Now, the two objective functions are the same. Thus, all we require is that  $x^* \in S$ , i.e.  $x$  is an integer feasible solution.

---

### End Solution 1

---

### Solution 2

---

Strong formulations preclude fractional solutions and improve LP bounding. By having tighter bounds in the LP relaxation, we can do more fathoming by bound in the branch and bound algorithm.

---

### End Solution 2

---

---

**Solution 3**

---

The optimal objective of the relaxed problem may be larger than or equal to that of the integer problem, but the integer program's optimal objective cannot exceed that of the relaxed problem.

---

**End Solution 3**

---

---

**Solution 4**

---

Without solving the problems, this question can't be answered. The optimal solution of one of the subproblems may or may not be larger than the solution of the integer problem.

---

**End Solution 4**

---

---

**Solution 5**

---

Yes. The combined feasible region of both problems is still larger than the feasible region of the integer problem since the additional constraints don't remove any integral values.

---

**End Solution 5**

---

---

**Solution 6**

---

The new constraint can be written as

$$x_1 + s = 2, \quad s \geq 0$$

Adding it to the tableau we get:

$$\left[ \begin{array}{c|ccccccc} \text{vars} & x_1 & x_2 & x_3 & x_4 & x_5 & s & RHS \\ \hline z & 0 & 0 & 4/7 & 1/7 & 0 & 0 & 59/7 \\ x_1 & 1 & 0 & 1/7 & 2/7 & 0 & 0 & 20/7 \\ x_2 & 0 & 1 & 0 & 1 & 0 & 0 & 3 \\ x_5 & 0 & 0 & -2/7 & 10/7 & 1 & 0 & 23/7 \\ s & 1 & 0 & 0 & 0 & 0 & 1 & 2 \end{array} \right]$$

Adding the  $x_1$  basis row to the new constraint we arrive at:

$$\left[ \begin{array}{c|ccccccc} \text{vars} & x_1 & x_2 & x_3 & x_4 & x_5 & s & RHS \\ \hline z & 0 & 0 & 4/7 & 1/7 & 0 & 0 & 59/7 \\ x_1 & 1 & 0 & 1/7 & 2/7 & 0 & 0 & 20/7 \\ x_2 & 0 & 1 & 0 & 1 & 0 & 0 & 3 \\ x_5 & 0 & 0 & -2/7 & 10/7 & 1 & 0 & 23/7 \\ s & 0 & 0 & -1/7 & -2/7 & 0 & 1 & -6/7 \end{array} \right]$$

This is primal infeasible (-ve RHS) but dual feasible! We can do a dual pivot. Now  $s$  leaves the basis and  $x_4$  enters.

$$\left[ \begin{array}{ccccccc|c} \text{vars} & x_1 & x_2 & x_3 & x_4 & x_5 & s & RHS \\ \hline z & 0 & 0 & 1/2 & 0 & 0 & 1/2 & 8 \\ x_1 & 1 & 0 & 0 & 0 & 0 & 1 & 2 \\ x_2 & 0 & 1 & -1/2 & 0 & 0 & 7/2 & 0 \\ x_5 & 0 & 0 & -1 & 0 & 1 & 5 & -1 \\ x_4 & 0 & 0 & 1/2 & 1 & 0 & -7/2 & 3 \end{array} \right]$$

$x_5$  has to leave and  $x_3$  enters.

$$\left[ \begin{array}{ccccccc|c} \text{vars} & x_1 & x_2 & x_3 & x_4 & x_5 & s & RHS \\ \hline z & 0 & 0 & 0 & 0 & 1/2 & 3 & 15/2 \\ x_1 & 1 & 0 & 0 & 0 & 0 & 1 & 2 \\ x_2 & 0 & 1 & 0 & 0 & -1/2 & 1 & 1/2 \\ x_3 & 0 & 0 & 1 & 0 & -1 & -5 & 1 \\ x_4 & 0 & 0 & 0 & 1 & 1/2 & -1 & 5/2 \end{array} \right]$$

Since all elements on the right hand side are positive, we have found a feasible solution. Furthermore, all reduced costs are non-negative, implying that we have found an optimal solution.

---

**End Solution 6**

---

**Solution 7**

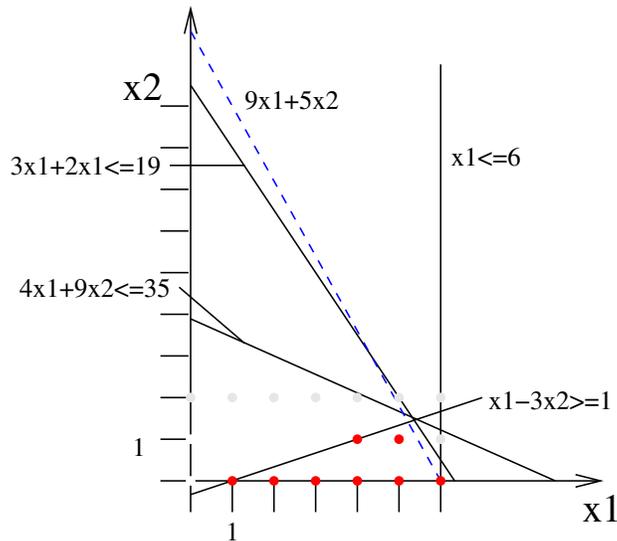
---

The optimum is  $x_1 = 6$ ,  $x_2 = 0$  with value of 54. Red dots denote feasible integer solutions.

---

**End Solution 7**

---




---

**Solution 8**

---

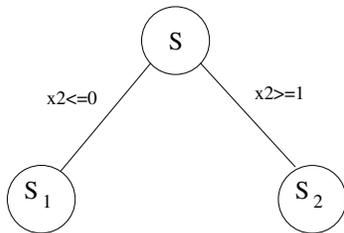
(*Bounding*) Usually the first step is to solve the relaxed problem, which is the linear program without integer constraints. Since the solution of this problem has been provided, we have an initial solution  $(6, 0.5)$  with an objective value of 56.5. Due to the fractional value of  $x_2$  this problem is not yet a feasible solution but allows us to define an upper bound  $\bar{z} = 56.5$ . The lower bound cannot be set yet, because the current solution is not feasible. Thus  $\underline{z} = -\infty$ .

(*Branching*) Now because  $\underline{z} \leq \bar{z}$ , we need to branch. The feasible region  $S$  is split around the fractional value. In our case this is  $x_2 = 0.5$ . We have two new feasible regions:

$$S_1 = S \cap \{x : x_2 \leq \lfloor x_2 \rfloor\}$$

$$S_2 = S \cap \{x : x_2 \geq \lceil x_2 \rceil\}.$$

By adding the constraints  $x_2 \leq 0$  or  $x_2 \geq 1$  to the original problem, we can define integer programs that have  $S_1$  or  $S_2$  as feasible regions.



(*Choosing a node*) Now we have the freedom to choose either of the two problems as a next

step in the optimization process. We arbitrarily chose  $S_1$ .

(*Reoptimizing*) Usually the new constraints would be added to the original problem and a new optimum would be sought by performing a few pivot steps with the dual simplex algorithm. Here we just pick the right solution from the tables.

The solution is  $(6, 0)$  with an objective of 54.0.

(*Bounding*) All variables in the solution are integral which makes it the feasible solution. The new bounds are  $\underline{z} = 54$ ,  $\bar{z} = 56.5$ .

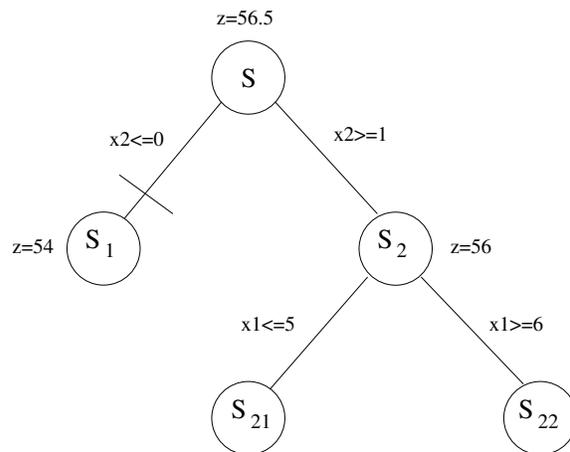
(*Choosing a node*) The fact that the lower bound is still below the upper bound allows for an even better solution. Therefore the node  $S_2$  needs to be checked as well.

(*Reoptimizing*) Again using the provided tables we find that the optimal solution with the additional constraint  $x_2 \geq 1$  is  $(5\frac{2}{3}, 1)$  with an objective of 56, which is above the current lowest bound. Furthermore, the solution is not fractional and requires another branching step.

(*Branching*) The feasible region  $S_2$  is split around the fractional value. In our case this is  $x_1 = 5\frac{2}{3}$ . We have two new feasible regions:

$$\begin{aligned} S_{21} &= S_2 \cap \{x : x_1 \leq \lfloor x_1 \rfloor\} \\ S_{22} &= S_2 \cap \{x : x_1 \geq \lceil x_1 \rceil\}. \end{aligned}$$

By adding the constraints  $x_1 \leq 5$  respectively  $x_1 \geq 6$  to the original problem, we can define linear programs that have  $S_{21}$  or  $S_{22}$  as feasible regions.



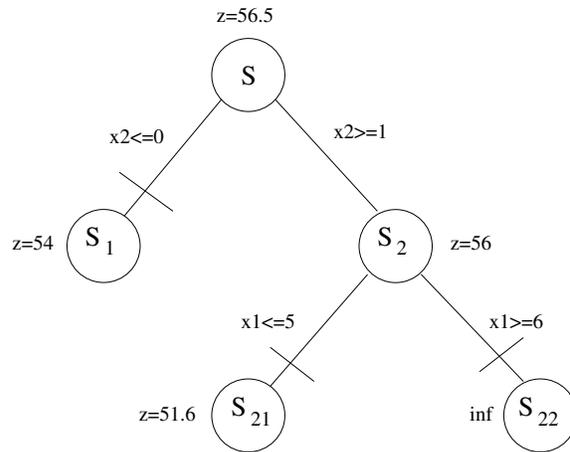
(*Choosing a node*) Again we pick  $S_{21}$  as first optimization target.

(*Reoptimizing*) From the table we find that the optimal solution to the new LP is  $(5, 1\frac{1}{3})$  with objective value  $51\frac{2}{3}$ . This value is below our current lower bound, which allows us

to prune this branch of the tree. If it had been above the current lower bound, another branch would have been required.

(*Choosing a node*) One more node  $S_{22}$  is currently left in the tree, so the choice is obvious.

(*Reoptimizing*) The table reveals that the additional constraints  $x_2 \geq 1$  and  $x_1 \geq 6$  lead to an infeasible problem. Therefore this branch can be pruned as well.



No more nodes are left in the tree. Therefore, the solution that defined the last lower bound is the optimal solution of the integer problem. It was  $(6, 0)$  with an objective value of 54.

---

**End Solution 8**

---

---

**Solution 9**

---

1. We can use the Big-M trick for this. For any reporter  $j$  and any time  $k$ ,  $A_{obama,j,k}$  denotes whether Mr. President gives an interview or not. Thus, the following constraint captures what we want:

$$\sum_i A_{i,j,k} \leq 1 + (1 - \sum_l A_{obama,l,k}) \cdot M \quad \forall j, k$$

where  $M$  is a sufficiently large constant. The constraint expresses that at any time  $k$ , the sum of all speakers speaking to any journalist at that time is at most 1, if the president is speaking to somebody at that time, and is not limited otherwise.

2. Of course, we could simply set  $M$  equal to the number of candidates -1. That would be enough for the constraint to be correct. However, we want  $M$  to be as tight as possible, to get a better formulation. We can use the information given in the input data about the candidate's availability to tighten  $M$ . In particular, for a particular time  $k$  instead of writing  $M$  we could write  $\sum_i c_{i,k} - 1$  which would represent the number of candidates available at time  $k$  -1. Thus, the new constraint could be:

$$\sum_i A_{i,j,k} \leq 1 + (1 - \sum_l A_{obama,l,k}) \cdot \sum_{i'} c_{i',k} \quad \forall j, k$$

---

**End Solution 9**

---

---

**Solution 10**

---

1. Note that we are asked to model a logical relation of the kind  $A \vee B$  where  $A$  denotes the event that less than 10 interviews are scheduled in the morning and  $B$  denotes the event that less than 10 interviews are schedule in the evening. We introduce binary variables  $\alpha_A$  and  $\alpha_B$  to capture the logical relation, and arrive at the following constraints:

$$\begin{aligned} \sum_{i,j} \sum_{k=1}^2 A_{ijk} &\leq 10 + M \cdot (1 - \alpha_A) \\ \sum_{i,j} \sum_{k=3}^4 A_{ijk} &\leq 10 + M \cdot (1 - \alpha_B) \\ \alpha_A + \alpha_B &\geq 1 \\ \alpha_A, \alpha_B &\in \{0, 1\} \end{aligned}$$

where  $M$  is a sufficiently large constant.

Note that  $\alpha_A$  only if 10 or fewer interviews in the morning, and  $\alpha_B$  only if 10 or fewer interviews in the afternoon. The constraint  $\alpha_A + \alpha_B \geq 1$  makes sure that at least one of the events  $A$  and  $B$  has to be activated. The first two constraints then guarantee that if the corresponding event is active, no more than 10 interviews can be schedule for that part of the day.

2. We could simply set  $M$  equal to the number of candidates times 2 minus 10, i.e.,  $M = 2 \cdot n - 10$ . That would be enough to make the constraint correct. But we want  $M$  to be as tight as possible. Similar to before, we set  $M = \sum_i \sum_{k=1}^2 c_{ik} - 10$  for the first constraint and analogous for the second one.

---

**End Solution 10**

---

---

**Solution 11**

---

1. In order to formulate the problem, we define for each  $e \in E$ , a variable  $x_e$  which is equal to one if edge  $d$  is included in the tree, and zero otherwise.
2. The objective is to minimize  $\sum_{e \in E} c_e x_e$ .
3. A spanning tree should have  $n-1$  edges. Thus, we introduce the constraint  $\sum_{e \in E} x_e = n - 1$ .
4. It can be shown that the chosen edges are guaranteed not to contain a cycle if for any nonempty set  $S \subset N$ , the number of edges with both endpoints in  $S$  is less than or equal to  $|S| - 1$ . Using the definition given, i.e., for any set  $S \subset N$ , we define  $E(S) = \{\{i, j\} \in E \mid i, j \in S\}$  we can express the subtour elimination constraints as:  
 $\sum_{e \in E(S)} x_e \leq |S| - 1, \quad \forall S \subset N, S \neq \emptyset, N$ .
5. We get the following IP:

$$\begin{aligned} & \text{minimize } \sum_{e \in E} c_e x_e \\ & \text{s.t. } \sum_{e \in E} x_e = n - 1 \\ & \quad \sum_{e \in E(S)} x_e \leq |S| - 1, \quad \forall S \subset N, S \neq \emptyset, N \\ & \quad x_e \in \{0, 1\} \end{aligned}$$

6. The subtour elimination formulation has an exponential number of constraints, namely  $2^n - 1$ .

---

**End Solution 11**

---

---

**Solution 12**

---

1. Using the definition of a cutset, we can express the connectivity requirement in terms of the constraints:  $\sum_{e \in \delta(S)} x_e \geq 1, \quad \forall S \in N, S \neq \emptyset, N$ .
2. We get the following IP:

$$\begin{aligned} & \text{minimize } \sum_{e \in E} c_e x_e \\ & \text{s.t. } \sum_{e \in E} x_e = n - 1 \\ & \quad \sum_{e \in \delta(S)} x_e \geq 1, \quad \forall S \in N, S \neq \emptyset, N \\ & \quad x_e \in \{0, 1\} \end{aligned}$$

3. The cutset formulation also has an exponential number of constraints, namely  $2^n - 1$ .

---

**End Solution 12**

---