2.098/6.255/15.093 - Recitation 7

Michael Frankovich and Shubham Gupta

November 6, 2009

1 Integer Programming Formulations

1.1 An investment problem:

Suppose you are interested in choosing a set of investments 1,. . . ,7 using binary variables. Model the following constraints:
1. You cannot invest in all of them.
2. You must choose at least one of them.
3. At most one of investments 1 and 3 can be chosen.
4. Investment 4 can be chosen only if investment 2 is also chosen.
5. You must choose either both of investments 1 and 5, or neither.
6. You must choose at least one of investments $1, 2, 3$ or at least two of investments $2, 4, 5, 6$.
Solution: Let x_i be 1 if we choose investment i (0 otherwise).

1. $\sum_{i=1}^{7} x_i \leq 6$
2. $\sum_{i=1}^{7} x_i \geq 1$
3. $x_1 + x_3 \leq 1$
4. $x_4 \leq x_2$
5. $x_1 = x_5$
6. $x_1 + x_2 + x_3 \geq y, \quad x_2 + x_4 + x_5 + x_6 \geq 2(1 - y), \quad y \in \{0, 1\}$

1.2 BT 10.9. The fixed charge network design problem:

We are given a directed graph $G = (\mathcal{N}, \mathcal{A})$ and a demand or supply b_i for each $i \in \mathcal{N}$, such that $\sum_{i \in \mathcal{N}} b_i = 0$. There are two types of costs: transportation costs, c_{ij}, of shipping one unit from node i to node j, and building costs, d_{ij}, of establishing a link (i, j) between nodes i and j of capacity u_{ij}. We would like to build such a network in order to minimize the total building and transportation costs, so that all demand is met. Formulate the problem as an integer programming problem.

Solution. There are two sets of decision variables. The first set $\boldsymbol{x}$ is used to decide which arcs to build ($\boldsymbol{x}$ are binary). The second set of nonnegative variables $\boldsymbol{f}$ is used to route flow on the arcs that are built. The problem formulation can be written as follows

$$\begin{array}{lrl} \min & \sum_{(i,j)\in A} d_{ij}x_{ij} + \sum_{(i,j)\in A} c_{ij}f_{ij} & \\ \text{s.t.} & b_i + \sum_{j\in I(i)} f_{ji} = \sum_{j\in O(i)} f_{ij} & \forall i \in N \\ & 0 \leq f_{ij} \leq u_{ij}x_{ij} & \forall (i,j) \in A \\ & x_{ij} \in \{0,1\} & \forall (i,j) \in A \end{array}$$

The only difference from the normal network flow problem is the additional arc building cost and the constraint $f_{ij} \leq u_{ij}x_{ij}$. The latter constraints simply indicate that if an arc is not built then we cannot route any flow through it. □

2 Exact Integer Algorithms

BT Examples 11.2 and 11.4

We want to solve the following IP:

$$\begin{array}{ll} \min & x_1 - 2x_2 \\ \text{s.t.} & -4x_1 + 6x_2 \leq 9 \\ & x_1 + x_2 \leq 4 \\ & x_1, x_2 \geq 0, \text{integer.} \end{array}$$

Preliminary Note: The term *LP relaxation* simply means the original problem without the integer constraints. Since the LP relaxation has fewer constraints, its optimal solution must be less than or equal to the optimal integral solution of the original IP (for a minimization problem). Thus the LP relaxation gives a lower bound for the optimal IP cost.

2.1 Gomory Cutting Planes

After adding slack variables and solving the LP relaxation, we have the following optimal tableau:

$\frac{35}{10}$	0	0	$\frac{3}{10}$	$\frac{2}{10}$
$x_1 = \frac{15}{10}$	1	0	$-\frac{1}{10}$	$\frac{6}{10}$
$x_2 = \frac{25}{10}$	0	1	$\frac{1}{10}$	$\frac{4}{10}$

Remember that the tableau gives us the coefficients for the equation $\boldsymbol{B}^{-1}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{B}^{-1}\boldsymbol{b}$. The second row of the tableau above thus gives us

$$x_2 + \frac{1}{10}x_3 + \frac{4}{10}x_4 = \frac{25}{10}.$$

Now since all the x_i are nonnegative, we can round down the fractions on the left hand side to get

$$x_2 + 0 \cdot x_3 + 0 \cdot x_4 \leq \frac{25}{10}.$$

And since all the x_i should be integer, we can round down the fraction on the right side to get

$$x_2 \leq 2.$$

In standard form, this constraint is $x_2 + x_5 = 2$, with $x_5 \geq 0$. We can add these new constraints to the LP formulation and solve again. The new optimal solution is $(\frac{3}{4}, 2)$. One of the equations in the optimal tableau is

$$x_1 - \frac{1}{4}x_3 + \frac{6}{4}x_5 = \frac{3}{4}.$$

The new Gomory cut is

$$x_1 - x_3 + x_5 \leq 0,$$

which is equivalent to

$$x_1 - (9 + 4x_1 - 6x_2) + (2 - x_2) = -3x_1 + 5x_2 - 7 \leq 0.$$

Again we add this new constraint to the LP and solve. The new optimal solution is $(1, 2)$, and since it is integer, it is an optimal solution to the original problem.

2.2 Branch and Bound

The basic idea behind branch and bound is divide and conquer, i.e., minimize the objective function over different areas of the feasible region (integer lattice within the polyhedron) and choose the best optimal solution.

- **Branching** The branching part of this method divides the feasible region into two or more disjoint subregions. (Disjointness of subproblems can potentially save computation since each region will not be checked repeatedly.) We can visualize branching as building up a tree from a starting root. For IP minimization problems, we have the following relation between a parent node and its children nodes: all the children subproblems have higher optimal values, because we impose on each child more constraints on top of the ones it inherits from its parent.

- **Bounding** Let z^* be the optimal solution of the original IP minimization problem. The root of the tree is an LP relaxation of the original problem, which provides a lower bound on z^* if the optimal LP solution is not integral (otherwise we have an optimal integral solution and the problem is solved). For a specific subproblem, if the optimal solution of its LP relaxation is integral, then this provides an upper bound of z^*. Throughout the algorithm, we keep track of the best known upper bound U. If the optimal solution of the LP relaxation is not integral, then this gives a lower bound for this specific subproblem. If this lower bound is at least U, then this node can be deleted (or pruned). Why? The optimal values of all its children will be at least at high as its own optimal value, but we know that $z^* \leq U$, so the children nodes cannot possibly be useful in leading us to the optimal solution of the original IP.

- In summary the branching process makes the tree grow while the bounding process restricts the tree from growing too big by cutting off unnecessary branches. The method terminates when there are no more subproblems to consider.

We apply the algorithm to the problem above:

Initialize $U = \infty$. As before, we solve the LP relaxation and find the optimal solution is $(\frac{15}{10}, \frac{25}{10})$, with cost $-\frac{35}{10}$. Now how do we create subproblems? Observe that x_2 is not integer (we could also use the fact that x_1 is not integer). We want the constraints of our subproblems to be violated by $(\frac{15}{10}, \frac{25}{10})$, so we create subproblems by adding the constraints $x_2 \geq 3$ (subproblem F_1) and $x_2 \leq 2$ (subproblem F_2). The LP relaxation of F_1 turns out to be infeasible, so we delete it. The optimal solution of the LP relaxation of F_2 is $(\frac{3}{4}, 2)$, with optimal cost $-\frac{13}{4}$. Now since the new x_1 is not integer, we break F_2 into two subproblems by adding the constraints $x_1 \geq 1$ (subproblem F_3) and $x_1 \leq 0$ (subproblem F_4).

The optimal solution of the LP relaxation of F_3 is $(1, 2)$, with optimal cost -3. Since we have the cost of a feasible integer solution, we can update $U = -3$. We can stop exploring this subproblem since we have an optimal integer solution for it. The optimal solution of the LP relaxation of F_4 is $(0, \frac{3}{2})$, with optimal cost -3. Now since this cost is at least $U = -3$, we do not need to explore F_4 anymore either. At this point, there are no more subproblems to consider. The optimal integer solution to the original problem is thus $(1, 2)$.

2 Integer Programming Duality

2.1 Lagrangean Duality

Integer Programming problems are hard to solve in general. However, there are some sets of constraints that can be sovled *more efficiently* than others. The main idea is to relax difficult constraints by adding *penalties* to the cost function instead of imposing them as constraints.

Consider the primal problem:

$$\begin{array}{ll} \min & c'x \\ \text{s/t.} & Ax \geq b \\ & x \in X \end{array}$$

where $X = \{x \in \mathcal{Z}^n | Dx \geq d\}$. Relax the difficult constraints $Ax \geq b$ with Lagrange multipliers $p \geq 0$ (if the constraints are equalities, then p is unconstrained), we obtain the problem:

$$\begin{array}{ll} \text{Z(p)} = & \min\ c'x + p'(b - Ax) \\ \text{s.t.} & x \in X \end{array}$$

Property: Z(p) is a piecewise concave function.

The Lagrange dual problem is written as follows:

$$Z_D = \max_{p \geq 0} Z(p)$$

Theorem:

$$\begin{array}{ll} Z_D = & \min\ c'x \\ \text{s.t.} & Ax \geq b \\ & x \in \text{conv}(X) \end{array}$$

N.b. X is a discrete set, with a finite number of points, while $\text{conv}(X)$ is a *continuous* set, in fact a *polyhedron*. Why? Try drawing these sets for a small example.

We have that the following integer programming weak duality inequalities always hold (for a minimization problem, of course):

$$Z_{LP} \leq Z_D \leq Z_{IP}.$$

2.2 BT Excercise 11.12 (Comparison of relaxations for an assignment problem with side constraints)

Note that from weak duality for integer programming problems, it follows that $Z_{LP} \leq Z_{D1}, Z_{D2}, Z_{D3}, Z_{D4} \leq Z_{IP}$.

Let $\mathcal{F} = \{x \in \mathcal{Z}^n | Dx \geq d\}$. Then, we have $Z_{LP} = Z_D$, for all cost vectors c, if and only if $\text{conv}(\mathcal{F}) = \{x | Dx \geq d\}$. Why?

Note the following relations:

- $\mathcal{F}_{D1} = \{x \in \mathcal{Z}^{n \times n} | \sum_{i=1}^{n} x_{ij} = 1, \forall j; \ \sum_{j=1}^{n} x_{ij} = 1, \forall i\}$
- $\mathcal{F}_{D2} = \{x \in \mathcal{Z}^{n \times n} | \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \geq b\}$
- $\mathcal{F}_{D3} = \{x \in \mathcal{Z}^{n \times n} | \sum_{j=1}^{n} x_{ij} = 1, \forall i; \ \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \geq b\}$
- $\mathcal{F}_{D4} = \{x \in \mathcal{Z}^{n \times n} | \sum_{j=1}^{n} x_{ij} = 1, \forall i\}$

Let P_{D1}, P_{D2}, P_{D3} and P_{D4} denote the polyhedra corresponding to the LP relaxation of the constraints in $\mathcal{F}_{D1}, \mathcal{F}_{D2}, \mathcal{F}_{D3}$ and $\mathcal{F}_{D4}$, respectively. Then the following relations hold:

- $\text{conv}(\mathcal{F}_{D1}) = P_{D1}$
- $\text{conv}(\mathcal{F}_{D2}) \subset P_{D2}$
- $\text{conv}(\mathcal{F}_{D3}) \subset P_{D3}$
- $\text{conv}(\mathcal{F}_{D4}) = P_{D4}$

Why are these true? Well, we always have $\text{conv}(\mathcal{F}_{Di}) \subset P_{Di}$. And we have equality when the constraints $Dx \geq d$ define an integral polyhedron (a polyhedron whose vertices are all integer points), so $\text{conv}(\mathcal{F}) = \text{conv}(\{z \in \mathcal{Z}^n | Dx \geq d\}) = \text{conv}(\{x | Dx \geq d\}) = \{x | Dx \geq d\} = P_D$ (the 2[nd] equality makes use of the fact that the polyhedron is integral).

Hence, it follows that $Z_{LP} = Z_{D1} = Z_{D4}$. Moreover, $Z_{LP} \leq Z_{D2}, Z_{D3} \leq Z_{IP}$. Finally, since $\text{conv}(\mathcal{F}_{D3}) \subset \text{conv}(\mathcal{F}_{D2})$, hence $Z_{D2} \leq Z_{D3}$. Therefore, we have $Z_{LP} = Z_{D1} = Z_{D4} \leq Z_{D2} \leq Z_{D3} \leq Z_{IP}$.