Optimization Methods	Date: *27/02/2018*
Instructor: *Sujit Prakash Gujar*	Scribes: Mohammad Saad Khan Shyam Sunder Reddy

Lecture 14: Integer Programming - The Branch and Bound Method and Bala's Algorithm

1 Recap

- Weighted edge matching in a bipartite graph.
- Integrality Gap:
$$\frac{OPT_{LP}}{OPT_{IP}} \geq 1 \ \ (Maximization)$$
- Simplex method is exponential in the worst case. Solving the LP relaxation to an IP and rounding off the solution may not give the correct solution.
- Also, the ellipsoid and interior point methods will not work for IP as they may give non integral solution.
- In an n variable IP, If each variable can take k values the number of possible assignments is k^n, which is exponential. Therefore, we need a clever way (a heuristic) to reduce the search space.

2 Branch and Bound

- Branch: The branch and bound approach is based on the principle that the total set of feasible solutions can be partitioned into smaller subsets of solutions. Therefore, it is a "Divide and Conquer" algorithm.
- Bound: Estimating how good a solution we can get for each smaller problems.This is done by solving the LP relaxation of the sub-problem. The optimal value from that will tell us whether there is a need to further divide or not.
- For the current solution, if we get integral objective value then we do not branch ahead. Otherwise, at least one of the variables has non-integral solution and we split the current problem to get two new ones, by adding new constraints to the original program:

  $P_1 : x_j \leq \lfloor x_j \rfloor$
  $P_2 : x_j \geq \lceil x_j \rceil$
- These smaller subsets can then be evaluated systematically until the best solution is found.
- Thus, Branch and Bound tries to reduce enumerating all decision variables whenever possible.

2.1 Algorithm

1. Let the best possible solution be denoted by x^- and lower bound be denoted by l^-
   $x^- = \phi$, $l^- = -\infty$

2. Let problem be denoted by P.
   P = maximize $c^T x$, $Ax \leq b$, $x \geq 0$
   $PUSH(P, LIST)$
3. $While(LIST \neq \phi)$
   (a) $Prb = POP(LIST)$
   (b) Solve Prb
   (c) If $x^* \in 2^n$ & $c^T x^* \geq l^-$:
   $x^- = x^*$, $l^- = c^T x*$
   (d) else if $\lfloor c^T x^* \rfloor \geq l^-$:
   $\exists x_j \in Z$
   $P_1 = Prb \cup x_j \leq \lfloor x_j \rfloor$
   $P_2 = Prb \cup x_j \geq \lceil x_j \rceil$
   $PUSH(P_1, LIST)$
   $PUSH(P_2, LIST)$

- If multiple variables are non integral, we have to choose on which to split ,among them (Lexicographic ordering is one option). The choice of the variable may influence the running time.

2.2 Example

Consider the following LP :

$$\begin{aligned} Maximize \quad & x_1 + x_2 \\ \text{subject to} \quad & x_1 + x_2 \leq 35 \\ & x_2 \leq x_1 \\ & 2x_1 + x_2 \leq 35.5 \\ & x_1, x_2 \geq 0 \\ & x_1, x_2 \in Z \end{aligned}$$

- The optimal solution by relaxing the integral constraints is ($x1 = 11.833$, $x2 = 11.833$) and corresponding objective value OPT=23.667. This is not integral, so we use branch and bound to try and get an integral solution.
- Using branch and bound we get:
- P1: solving we get $l^- = 22$ and $x^- = (11, 11)$. Since, we have an integral solution we will not branch further. We move on to P2.
- P21: after adding new constraints the LP becomes:

$$\begin{aligned} Maximize \quad & x_1 + x_2 \\ \text{subject to} \quad & x_1 + x_2 \leq 35 \\ & x_2 \leq x_1 \\ & 2x_1 + x_2 \leq 35.5 \\ & x_1 \geq 12 \\ & x_2 \leq 11 \\ & x_1, x_2 \geq 0 \\ & x_1, x_2 \in Z \end{aligned}$$

  Solving the LP relaxation, we get $Z = 23.25$ and $x^* = (12.25, 11)$. Since the optimal solution here is not integral, we do not update l^- and x^-. We branch further on x1.

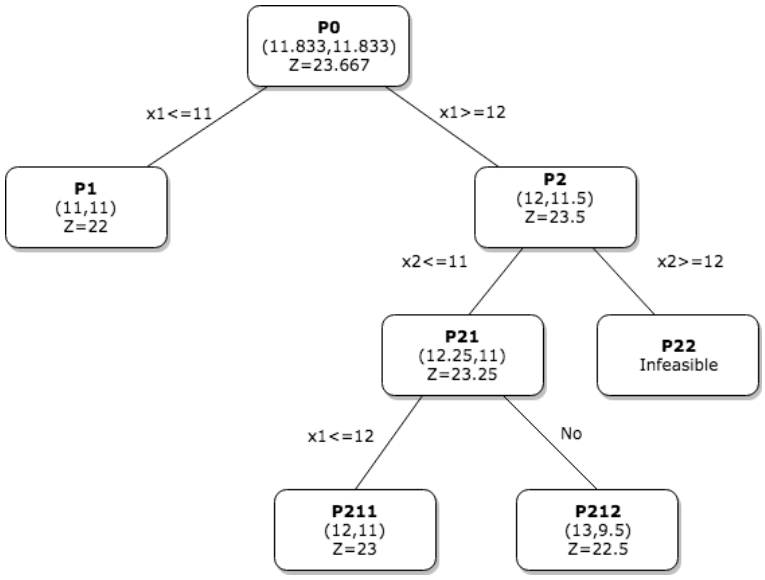

Figure 1: Branch and Bound example

- P22: according to the constraints:

$$x_2 \leq x_1$$
$$x_1 \geq 12$$
$$x_2 \geq 12$$

  we see that $x_1 = x_2 = 12$ but that does not satisfy $x_1 + x_2 \leq 35$. Therefore, this IP has no solution.

- P211: added constraint is $x_1 \leq 12$. Solving the new IP using LP relaxation we get $Z = 23$ and $x^* = (12, 11)$. Since, $Z > l^-$ we update $l^- = 23$ and $x^- = (12, 11)$.

- P212: Solving the LP relaxation we get $Z = 22.5$ and $x^* = (13, 9.5)$. We can see that branching further on x_2 will not help as we will not get a better objective value than 23.

- So, the optimal solution to the IP is $x_1 = 12$, $x_2 = 11$ and $Z = 23$.

2.3 Why are Integer Programs hard?

- The feasible region formed is not convex but rather consists of points where the decision variables are integral.

- If we try to take integers closer to the boundary of the feasible region of the LP relaxation, it has still exponential number of integral feasible solutions.

- This is because the volume of a higher dimensional feasible region is always concentrated at the surface. Consider a cube, in 1D i.e. $n = 1$ if we increase the edge of cube from 1 to 1.1, volume increases by 10%. For 2D i.e. $n = 2$, it increases by $1.21 - 1 = 21\%$. Similarly, for 8D i.e. $n = 8$ it increase by $1.1^8 - 1 = 114\%$ i.e. more than half the volume is closer to the boundary. Hence, even if we decide to greedily check only integral points closer to the boundary, still there are exponential assignments to be verified.

2.4 An Interesting Problem

- Let us put a Non-Linear Problem given below into a Partially Integer problem :

$$|x| \geq 3$$

- It can be written as:

$$x \leq -3 \quad (or) \quad x \geq 3$$

- But the Linear program needs to have (and) between the conditions. So, let us introduce a binary integer $y \in \{0, 1\}$ and very large integer L. We can formulate the above Non-Linear Problem as:

$$\begin{aligned} x &\leq -3 + yL \\ x &\leq -3 + (1-y)L \\ y &\in \{0, 1\} \end{aligned}$$

- Therefore by using some techniques certain Non-Linear Programs can be converted into Partially Linear or Integer Programs.

3 Binary Integer Programs

3.1 Introduction

- When all variables are Binary, this class of problems are known as Binary Integer problems. i.e. each variable can only take on the value of 0 or 1 ($x_i \in \{0, 1\}$).
- When all variables are integer valued, and the constraints are linear, then this class is known as Mixed integer problems.
- Like Mixed Integer programming, BIP doesnt guarantee polynomial time solution and many BIP comes under NP-complete problems. It is an NP-hard problem in combinatorial optimization. The decision version of the problem though, is NP-Complete.

3.2 Bala's Algorithm

Like Branch and Bound to solve IP (as seen in previous section) we have Bala's additive algorithm to solve Binary integer programs.
Bala's algorithm requires problem to be in the following form.
The objective function has the form:

$$\begin{gathered} \text{Minimize} \qquad \sum_i c_i x_i \\ \text{subject to} \\ Ax \geq b \\ x_i \in \{0, 1\} \\ c_i \geq 0 \forall c_i \in C \\ 0 \leq c_1 \leq c_2 ... \leq c_n \end{gathered}$$

3.3 Bala's Algorithm Cases

Now there can be few cases which arises due to these restriction on standard form. Some of them are:

- Case 1: for some c_i and c_j. If $c_j \leq c_i$ and $i \leq j$. Then we rewrite $x_i = y_k$ and $x_j = y_l$ where $l \leq k$ (reordering). E.g.: $z = 2x_1 + x_2$ will be rewritten as $y_1 + 2y_2$, where $x_1 = y_2$ and $x_2 = y_1$
- Case 2: if objective function is maximized than minimize it. Max $C^T x = \min -C^T x$
- Case 3: if some c_i is negative, then we can do as follows,

$$\begin{aligned} &\text{e.g:- } -3x_1 \\ &\text{let } y_1 = 1 - x_1 \\ \implies &3y_1 = 3 - 3x_1 \\ \implies &-3x_1 = 3y_1 - 3 \end{aligned}$$

Now Let us take an Integer Program and convert it into Bala's Input form:

$$\begin{aligned} Maximize \quad & 8x_1 + 11x_2 + 6x_3 + 4x_4 \\ \text{subject to} \quad & 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\ & x_i \in \{0, 1\} \end{aligned}$$

Using Case 2 condition in the above cases it becomes:

$$\begin{aligned} Minimize \quad & -8x_1 - 11x_2 - 6x_3 - 4x_4 \\ \text{subject to} \quad & -5x_1 - 7x_2 - 4x_3 - 3x_4 \geq -14 \\ & x_i \in \{0, 1\} \end{aligned}$$

Using Case 3 condition in the above cases consider $y_i \in \{0, 1\} \ni x_i = 1 - y_i \implies -8x_1 = -8 + 8y_1$ similarly others we get:

$$\begin{aligned} Minimize \quad & 8y_1 + 11y_2 + 6y_3 + 4y_4 - 29 \\ \text{subject to} \quad & 5y_1 + 7y_2 + 4y_3 + 3y_4 \geq 5 \\ & y_i \in \{0, 1\} \end{aligned}$$

also $0 \leq c_1 \leq c_2 ... \leq c_n$ should be satisfied
Consider $z_i \in \{0, 1\} \ni z_1 = y_4, z_2 = y_3, z_3 = y_1, z_4 = y_2$ we get :

$$\begin{aligned} Minimize \quad & 4z_1 + 6z_2 + 8z_3 + 11z_4 - 29 \\ \text{subject to} \quad & 3z_1 + 4z_2 + 5z_3 + 7z_4 \geq 5 \\ & z_i \in \{0, 1\} \end{aligned}$$

The above objective function is in the format required by the Bala's Algorithm.

Therefore, by the above mentioned rules any Integer Program can be converted into objective function required by Bala's Algorithm

3.4 Bala's Algorithm Uses

A few keypoints about BIPs that Bala's algorithm uses, are:-

- The objective sense of the function is minimization, and hence the algorithm prefers to assign all the variables 0, to get the lowest value of z
- If we cannot set all values of x to 0, then we prefer to set the x with the smallest index to 1, so as to increase the value of the objective the least.
- Bala's algorithm assumes that if $x_i = 1$, then it might provide a feasible solution which is bounded by $\sum_i c_i x_i$

- If $x_i = 0$, we need to set atleast one other variable than the current x_i's to 0, and the next cheapest one to set would be x_{i+1}.
- If there are n variables, we need to evaluate 2^n possible configuration of the x.

3.5 Bala's Algorithm Example

Consider the example:

$$\begin{aligned} Minimize \quad & z = 3x_1 + 5x_2 + 6x_3 + 9x_4 + 10x_5 + 10x_6 \\ \text{subject to} \quad & -2x_1 + 6x_2 - 3x_3 + 4x_4 + x_5 - 2x_6 \geq 2 \\ & -5x_1 - 3x_2 + x_3 + 3x_4 - 2x_5 + x_6 \geq -2 \\ & 5x_1 - x_2 + 4x_3 - 2x_4 + 2x_5 - x_6 \geq 3 \\ & x_i \in \{0, 1\} \end{aligned}$$

- Nodes obtained by assigning 1 to their values are known as 1-nodes. Similarly, there exist 0-nodes.If a 1-node is feasible, then it is said to be fathomed.
- If it is a 1-node, check the feasibility by evaluating all constraints. 0-nodes are in-feasible for the same reasons that their parents aren't feasible.
- For both 1 and 0 nodes, check all decisions made thus far and if they are violated, prune this node.
- If it is a 0-node, then check whether the bounding function solution is feasible, if it is, then the node is fathomed with the bounding function solution.
- Using Bala's Algorithm we get:
- Optimal value Z = 11 with $x_1 = 0$, $x_2 = 1$, $x_3 = 1$, $x_4 = 0$, $x_5 = 0$, $x_6 = 0$

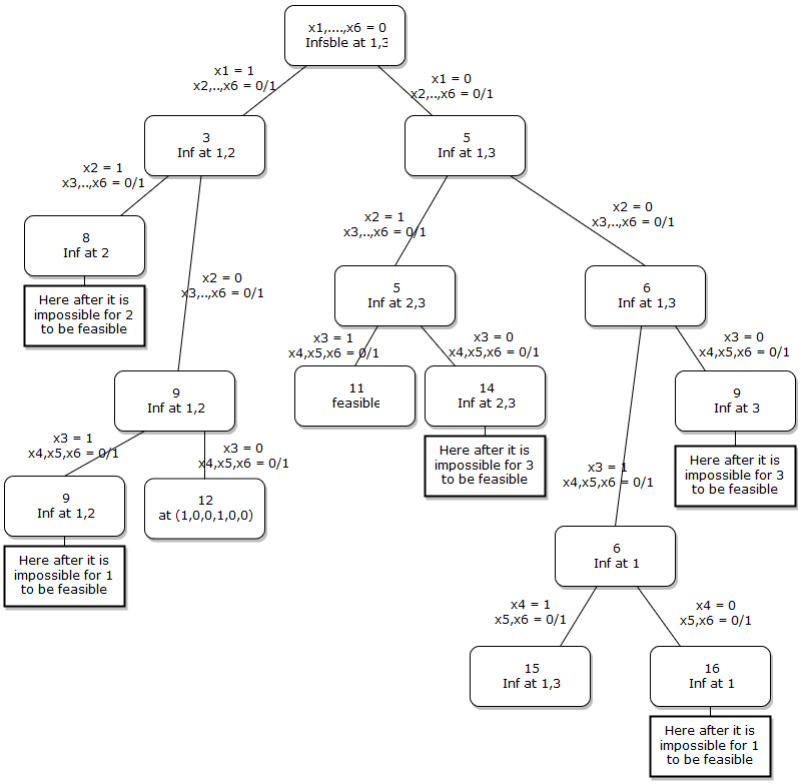

Figure 2: Bala's Algorithm Example