Stanford MS&E 236 / CS 225: Lecture 11
Integer programming solvers

Ellen Vitercik*

May 12, 2024

In this lecture, we cover the branch-and-bound (B&B) algorithm [1], which is the primary algorithm used to solve integer programs (IPs). It is the algorithm used by commercial solvers like CPLEX and Gurobi. For simplicity, we will focus on binary IPs:

$$\begin{array}{ll} \text{maximize} & \boldsymbol{c}^T\boldsymbol{x} \\ \text{subject to} & A\boldsymbol{x} \le \boldsymbol{b} \\ & \boldsymbol{x} \in \{0,1\}^n. \end{array} \tag{1}$$

However, all of the ideas we discuss will apply to general integer programs, which may have continuous variables as well. This linear programming (LP) relaxation of a binary integer program has the form

$$\begin{array}{ll} \text{maximize} & \boldsymbol{c}^T\boldsymbol{x} \\ \text{subject to} & A\boldsymbol{x} \le \boldsymbol{b} \\ & x_i \in [0,1] \quad \text{for all } i \in [n]. \end{array} \tag{2}$$

Throughout this lecture, we will use the following important observation.

Observation 1. Let $\boldsymbol{x'}$ be the solution to the LP relaxation with an additional constraint, for example, $x_i = 0$:

$$\begin{array}{ll} \text{maximize} & \boldsymbol{c}^T\boldsymbol{x} \\ \text{subject to} & A\boldsymbol{x} \le \boldsymbol{b} \\ & x_i = 0 \\ & x_j \in [0,1] \quad \text{for all } j \neq i. \end{array}$$

Let $\boldsymbol{x}^*_{\mathrm{LP}}$ be the optimal solution to LP 2. Then $\boldsymbol{c}^T\boldsymbol{x'} \le \boldsymbol{c}^T\boldsymbol{x}^*_{\mathrm{LP}}$: by further constraining the LP, we can only decrease the objective value.

1 Branch-and-bound

To describe B&B, we will use the following running example:

$$\begin{array}{ll} \text{maximize} & 15x_1 + 12x_2 + 4x_3 + 2x_4 \\ \text{subject to} & 8x_1 + 5x_2 + 3x_3 + 2x_4 \le 10 \\ & x_1, x_2, x_3, x_4 \in \{0,1\}. \end{array} \tag{3}$$

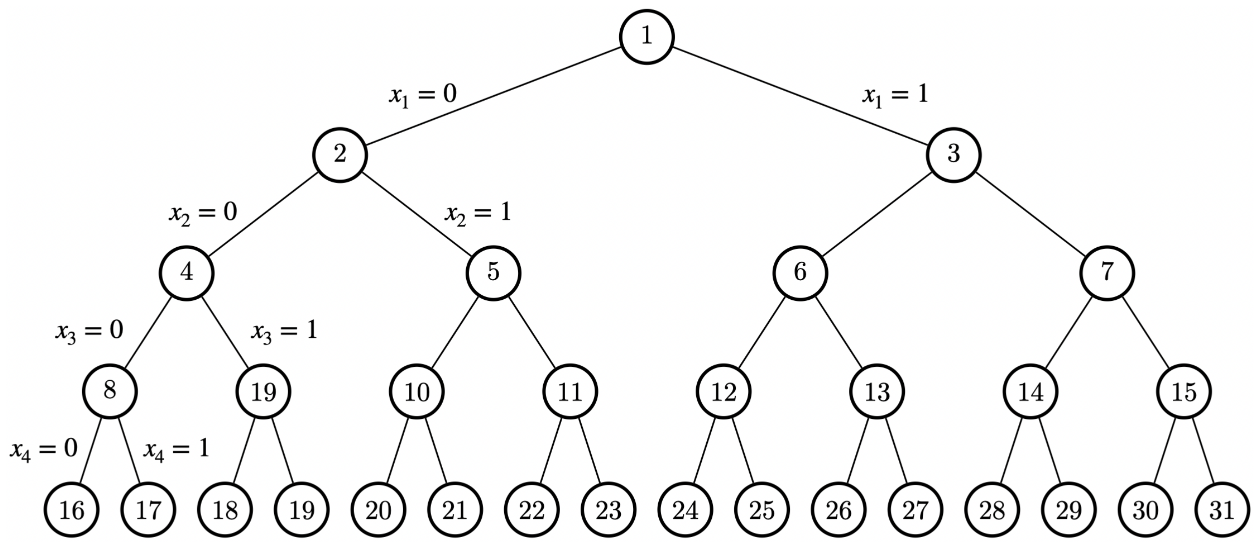

Figure 1: Enumeration tree.

Number of variables	**Solution times**
30	1 second
40	17 minutes
50	11.6 days
60	31 years
70	31,000 years

Table 1: Runtime to brute-force search for the optimal solution to an IP if we can evaluate 1 billion solutions per second [2].

The first thing we might try to solve this integer program is brute force search, which can be organized using the *enumeration tree* in Figure 1. At each node of this tree, we *branch* on a variable (x_1 in the first layer, then x_2, and so on). On each branch, the variable is restricted to equal either 0 or 1. Thus, each of the leaves corresponds to a possible solution to the IP. For example, the leaf labeled 16 corresponds to the solution $\boldsymbol{x} = \mathbf{0}$, and the leaf labeled 26 corresponds to the solution $\boldsymbol{x} = (1, 0, 1, 0)$. To perform brute-force search, we could compute the IP's objective for each leaf and pick the best one.

To understand the enormous runtime of brute-force search, suppose we could evaluate 1 billion solutions per second. Table 1 shows how long it would take to evaluate every leaf of the enumeration tree for relatively small IPs. Clearly, this is not a reasonable approach!

The key idea of B&B is that by using LP relations, we can bound the optimal solutions in subtrees of the enumeration tree. By doing so, we can eliminate entire subtrees when we recognize they only contain solutions with low objective values. To illustrate, we begin with several thought experiments.

*These notes are course material and have not undergone formal peer review. Please feel free to send me any typos or comments.

Thought experiment #1. Suppose we solve the LP relaxation of the problem corresponding to node 7 of the enumeration tree in Figure 1, which we refer to as Problem(7). Specifically, Problem(7) corresponds to the following LP:

$$\begin{array}{ll} \text{maximize} & 15x_1 + 12x_2 + 4x_3 + 2x_4 \\ \text{subject to} & 8x_1 + 5x_2 + 3x_3 + 2x_4 \leq 10 \\ & x_1 = 1 \\ & x_2 = 1 \\ & x_3, x_4 \in [0, 1]. \end{array}$$

This LP is infeasible since if $x_1 = x_2 = 1$, then the inequality constraint becomes $13 + 3x_3 + 2x_4 \leq 10$, which is not possible for $x_3, x_4 \in [0, 1]$. Moreover, there is no way we can make this LP feasible by adding more constraints on x_3 or x_4. Therefore, we can eliminate, or *prune*, the subtree rooted at node 7: the optimal solution cannot be any of its leaves.

Thought experiment #2. Suppose we have discovered a feasible binary solution $\boldsymbol{x}^* = (0, 1, 1, 1)$ with $\boldsymbol{c}^T\boldsymbol{x}^* = 18$, but we do not yet know if it is the optimal integer solution. Suppose we solve the LP relaxation corresponding to node 6, i.e., Problem(6):

$$\begin{array}{ll} \text{maximize} & 15x_1 + 12x_2 + 4x_3 + 2x_4 \\ \text{subject to} & 8x_1 + 5x_2 + 3x_3 + 2x_4 \leq 10 \\ & x_1 = 1 \\ & x_2 = 0 \\ & x_3, x_4 \in [0, 1]. \end{array}$$

Its solution is $\left(1, 0, \frac{2}{3}, 0\right)$, with an objective value of 17.66. Since $17.66 < 18$, we can prune the entire subtree rooted at node 6: by further constraining x_3 and x_4, the objective value will only become worse, as we discussed in Observation 1.

We are now ready to present B&B, Algorithm 1. To start, we assume that we have a feasible solution $\boldsymbol{x}^*$ to IP (1), e.g., $\boldsymbol{x}^* = \boldsymbol{0}$. In each iteration of B&B, $\boldsymbol{x}^*$ is called the *incument solution*, i.e., "best so far." We describe the algorithm by working step-by-step through how it would solve IP (3) in Figure 2-7. As these figures illustrate, to find the optimal solution, we only have to explore 7 nodes, rather than all 31 nodes in Figure 1. Thus, by pruning subtrees, B&B can significantly speed up the time it takes to solve IPs.

In the next class, we will discuss two key policies underlying B&B: node selection and variable selection. There is no reason why we should branch on variable x_1 first, then x_2, and so on in Figure 1. In fact, the variable selection policy can have a huge impact on the runtime of B&B. Similarly, the policy governing which active node to explore next is an important determinant of B&B's runtime.

References

[1] Ailsa H Land and Alison G Doig. An automatic method of solving discrete programming problems. *Econometrica*, pages 497–520, 1960.

[2] James Orlin and Ebrahim Nasrabadi. IP techniques 1. branch and bound, 2013. URL https://ocw.mit.edu/courses/

Algorithm 1 Vanilla B&B algorithm

Input: Binary integer program (Equation (1)) with feasible solution $\boldsymbol{x}^*$

1: $z^* \leftarrow \boldsymbol{c}^T \boldsymbol{x}^*$
2: Mark the root node of the enumeration tree as "active"
3: **while** there remain active nodes **do**
4:     Select an active node j and mark it as "inactive"
5:     **if** Problem(j) is infeasible **then**
6:         Prune node j
7:     **else**
8:         $\boldsymbol{x}(j) \leftarrow$ optimal solution of the LP relaxation to Problem(j)
9:         $z(j) = \boldsymbol{c}^T \boldsymbol{x}(j)$
10:         **if** $z^* < z(j)$ and $\boldsymbol{x}(j) \notin \{0,1\}^n$ **then**
            ▷ `It may be possible to find a better incumbent among` j`'s descendants`
11:             Mark the direct descendants of node j as "active"
12:         **else if** $z^* < z(j)$ and $\boldsymbol{x}(j) \in \{0,1\}^n$ **then**
13:             $\boldsymbol{x}^* \leftarrow \boldsymbol{x}(j)$ ▷ $\boldsymbol{x}^*$ `may be the optimal solution`
14:             Prune node j
15:         **else**
16:             Prune node j

Output: $\boldsymbol{x}^*$

LP relaxation of Problem(1):

$$\begin{array}{ll} \text{maximize} & 15x_1 + 12x_2 + 4x_3 + 2x_4 \\ \text{subject to} & 8x_1 + 5x_2 + 3x_3 + 2x_4 \leq 10 \\ & x_1, x_2, x_3, x_4 \in [0,1] \end{array}$$

Optimal solution of LP relaxation:

$$\boldsymbol{x}(1) = \left(\frac{5}{8}, 1, 0, 0\right), \quad z(1) = 21.38$$

Incumbent solution:

$$\boldsymbol{x}^* = (0,0,0,0), \quad z^* = 0$$

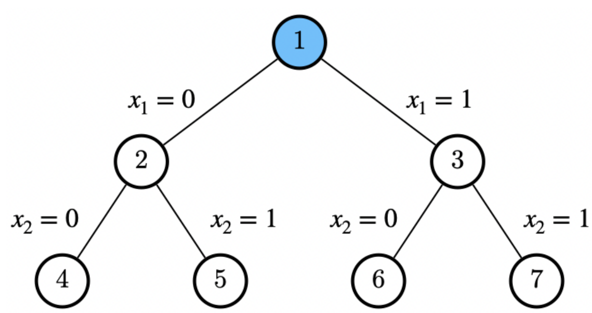

Figure 2: First, B&B marks the root node of the enumeration tree as "active," represented by the blue node. It solves the LP relaxation of Problem(1). Since $\boldsymbol{x}(1) = \left(\frac{5}{8}, 1, 0, 0\right) \notin \{0,1\}$ and $z(1) = 21.38 > z^* = 0$, we follow the directions of the if statement in Line 10 and mark nodes 2 and 3 as "active," as illustrated in Figure 3.

LP relaxation of Problem(2):

$$
\begin{array}{ll}
\text{maximize} & 15x_1 + 12x_2 + 4x_3 + 2x_4 \\
\text{subject to} & 8x_1 + 5x_2 + 3x_3 + 2x_4 \leq 10 \\
 & x_1 = 0 \\
 & x_2, x_3, x_4 \in [0, 1]
\end{array}
$$

Optimal solution of LP relaxation:

$\boldsymbol{x}(2) = (0, 1, 1, 1)\,, \quad z(2) = 18$

(New) incumbent solution:

$\boldsymbol{x}^* = (0, 1, 1, 1), \quad z^* = 18$

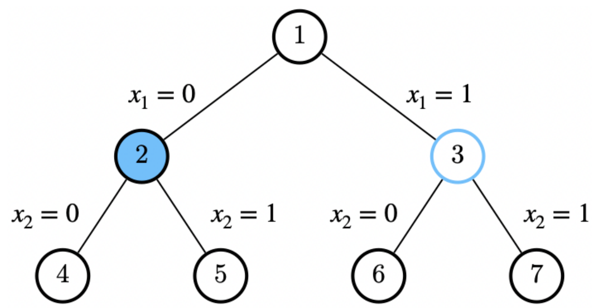

Figure 3: Next, we explore node 2. In Problem(2), we add the additional constraint that $x_1 = 0$. The optimal solution to the LP relaxation of Problem(2) is $\boldsymbol{x}(2) = (0, 1, 1, 1)$, which is integral and thus could be the optimal solution to the integer program. Since $z(2) = 18 > z^* = 0$, we follow the directions of the if statement in Line 12: we set $\boldsymbol{x}^* \leftarrow (0, 1, 1, 1)$ and prune node 2, as illustrated in Figure 4.

LP relaxation of Problem(3):

$$
\begin{array}{ll}
\text{maximize} & 15x_1 + 12x_2 + 4x_3 + 2x_4 \\
\text{subject to} & 8x_1 + 5x_2 + 3x_3 + 2x_4 \leq 10 \\
 & x_1 = 1 \\
 & x_2, x_3, x_4 \in [0, 1]
\end{array}
$$

Optimal solution of LP relaxation:

$\boldsymbol{x}(1) = \left(1, \frac{2}{5}, 0, 0\right), \quad z(1) = 19.8$

Incumbent solution:

$\boldsymbol{x}^* = (0, 1, 1, 1), \quad z^* = 18$

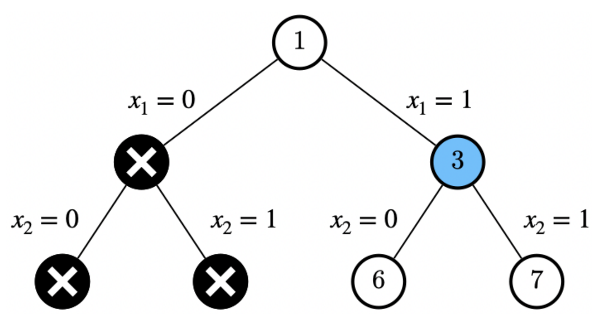

Figure 4: Next, we explore node 3. Since $\boldsymbol{x}(3) = \left(1, \frac{2}{5}, 0, 0\right)$ and $z(3) = 19.8 > z^* = 18$, we follow the if statement in Line 10 and mark the descendants of node 3 as "active," as illustrated in Figure 5.

LP relaxation of Problem(6):

$$\begin{array}{ll} \text{maximize} & 15x_1 + 12x_2 + 4x_3 + 2x_4 \\ \text{subject to} & 8x_1 + 5x_2 + 3x_3 + 2x_4 \leq 10 \\ & x_1 = 1 \\ & x_2 = 0 \\ & x_3, x_4 \in [0,1] \end{array}$$

Optimal solution of LP relaxation:

$$\boldsymbol{x}(6) = \left(1, 0, \frac{2}{3}, 0\right), \quad z(6) = 17.66$$

Incumbent solution:

$$\boldsymbol{x}^* = (0,1,1,1), \quad z^* = 18$$

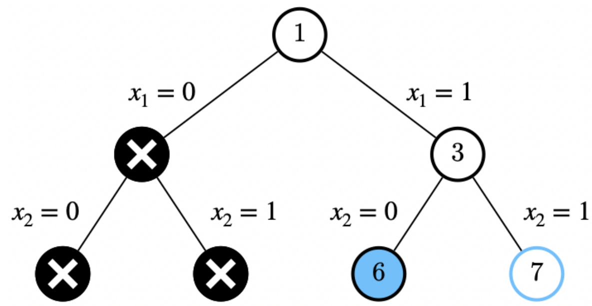

Figure 5: Next, we explore node 6. Since $z(6) = 17.66 < 18 = z^*$, we follow the directions of the if statement in Line 15 and simply prune node 6, as illustrated in Figure 6.

LP relaxation of Problem(7):

$$\begin{array}{ll} \text{maximize} & 15x_1 + 12x_2 + 4x_3 + 2x_4 \\ \text{subject to} & 8x_1 + 5x_2 + 3x_3 + 2x_4 \leq 10 \\ & x_1 = 1 \\ & x_2 = 1 \\ & x_3, x_4 \in [0,1] \end{array}$$

LP relaxation is *infeasible*

Incumbent solution:

$$\boldsymbol{x}^* = (0,1,1,1), \quad z^* = 18$$

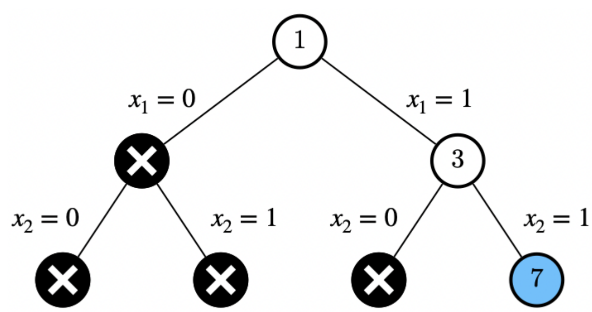

Figure 6: The only active node is node 7. The LP relaxation of Problem(7) is infeasible. Therefore, we prune node 7, as instructed by Step 5 and illustrated in Figure 7.

Optimal solution:

$$\boldsymbol{x}^* = (0,1,1,1), \quad z^* = 18$$

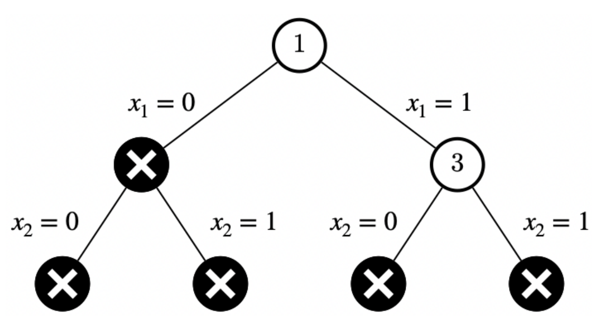

Figure 7: Finally, since there are no more active nodes, as illustrated in Figure 7, B&B is terminated. We are now guaranteed that the incumbent solution $\boldsymbol{x}^* = (0,1,1,1)$ is in fact the optimal solution.

15-053-optimization-methods-in-management-science-spring-2013/resources/mit15_053s13_lec12/. [Online; accessed 5-May-2024].