Branch and Bound

We will explain branch and bound by using the capital budgeting example from the previous section. In that problem, the model is

$$\begin{array}{ll} \text{Maximize} & 8x_1 + 11x_2 + 6x_3 + 4x_4 \\ \text{subject to} & 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\ & x_j \in \{0,1\}\ j = 1, \ldots 4. \end{array}$$

The linear relaxation solution is $x_1 = 1, x_2 = 1, x_3 = 0.5, x_4 = 0$ with a value of 22. We know that no integer solution will have value more than 22. Unfortunately, since x_3 is not integer, we do not have an integer solution yet.

We want to force x_3 to be integer. To do so, we *branch* on x_3, creating two new problems. In one, we will add the constraint $x_3 = 0$. In the other, we add the constraint $x_3 = 1$. This is illustrated in Figure 2.

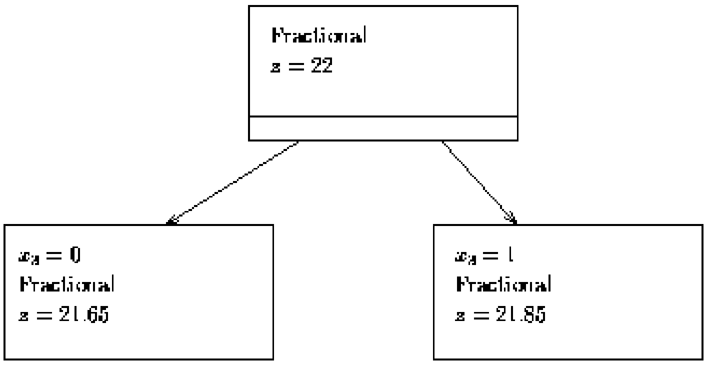

Figure 2: First Branching

Note that any optimal solution to the overall problem must be feasible to one of the subproblems. If we solve the linear relaxations of the subproblems, we get the following solutions:

- $x_3 = 0$: objective 21.65, $x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 0.667$;
- $x_3 = 1$: objective 21.85, $x_1 = 1, x_2 = 0.714, x_3 = 1, x_4 = 0$.

At this point we know that the optimal integer solution is no more than 21.85 (we actually know it is less than or equal to 21 (Why?)), but we still do not have any feasible integer solution. So, we will take a subproblem and branch on one of its variables. In general, we will choose the subproblem as follows:

- We will choose an *active* subproblem, which so far only means one we have not chosen before, and
- We will choose the subproblem with the highest solution value (for maximization) (lowest for minimization).

In this case, we will choose the subproblem with $x_3 = 1$, and branch on x_2. After solving the resulting subproblems, we have the branch and bound tree in Figure 3.

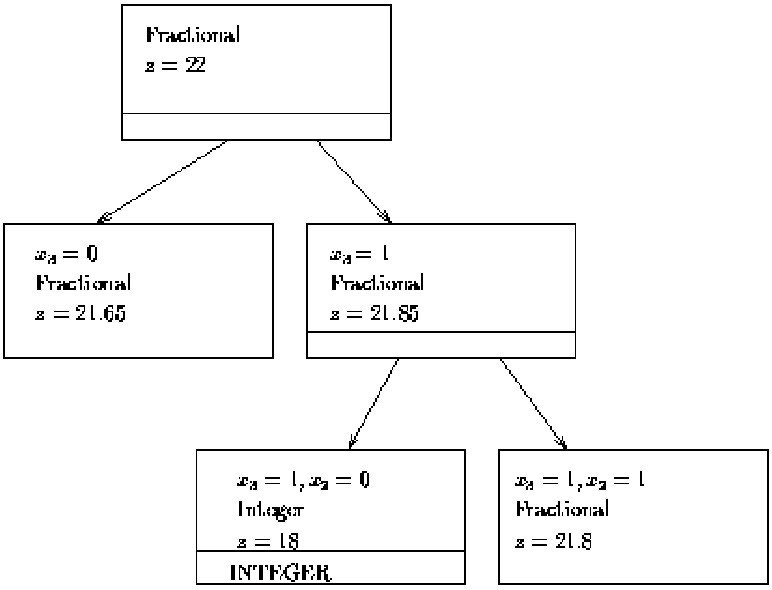

Figure 3: Second Branching

The solutions are:

- $x_3 = 1, x_2 = 0$: objective 18, $x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 1$;
- $x_3 = 1, x_2 = 1$: objective 21.8, $x_1 = 0.6, x_2 = 1, x_3 = 1, x_4 = 0$.

We now have a feasible integer solution with value 18. Furthermore, since the $x_3 = 1, x_2 = 0$ problem gave an integer solution, no further branching on that problem is necessary. It is not active due to integrality of solution. There are still active subproblems that might give values more than 18. Using our rules, we will branch on problem $x_3 = 1, x_2 = 1$ by branching on x_1 to get Figure 4.

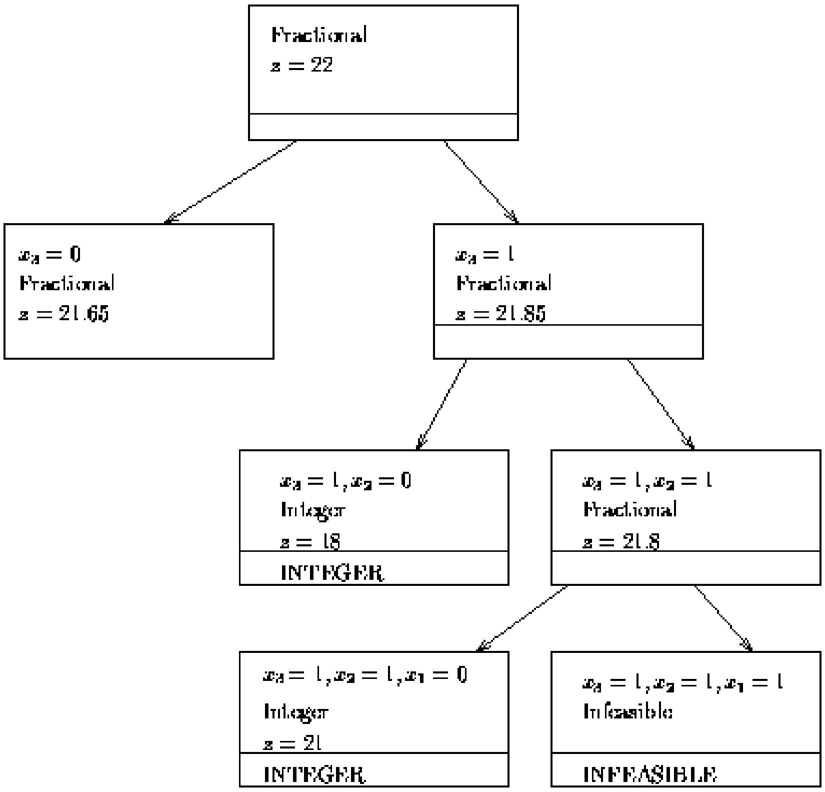

Figure 4: Third Branching

The solutions are:

- $x_3 = 1, x_2 = 1, x_1 = 0$: objective 21, $x_1 = 0, x_2 = 1, x_3 = 1, x_4 = 1$;
- $x_3 = 1, x_2 = 1, x_1 = 1$: infeasible.

Our best integer solution now has value 21. The subproblem that generates that is not active due to integrality of solution. The other subproblem generated is not active due to infeasibility. There is still a subproblem that is active. It is the subproblem with solution value 21.65. By our ‘‘round-down’’ result, there is no better solution for this subproblem than 21. But we already have a solution with value 21. It is not useful to search for another such solution. We can *fathom* this subproblem based on the above bounding argument and mark it not active. There are no longer any active subproblems, so the optimal solution value is 21.

We have seen all parts of the branch and bound algorithm. The essence of the algorithm is as follows:

1. Solve the linear relaxation of the problem. If the solution is integer, then we are done. Otherwise create two new subproblems by branching on a fractional variable.
2. A subproblem is not active when any of the following occurs:
    1. You used the subproblem to branch on,
    2. All variables in the solution are integer,
    3. The subproblem is infeasible,

4. You can fathom the subproblem by a bounding argument.

3. Choose an active subproblem and branch on a fractional variable. Repeat until there are no active subproblems.

That's all there is to branch and bound! Depending on the type of problem, the branching rule may change somewhat. For instance, if **x** is restricted to be integer (but not necessarily 0 or 1), then if $x = 4.27$ your would branch with the constraints $x \leq 4$ and $x \geq 5$ (not on **x=4** and **x=5**).

In the worst case, the number of subproblems can get huge. For many problems in practice, however, the number of subproblems is quite reasonable.

For an example of a huge number of subproblems, try the following in LINGO:

```
model:
  sets:
    a /1..17/: x;
  endsets

  max = -x0 + @sum(a: 2 * x);
  x0 + @sum(a: 2 * x) < 17;
  @for (a: @bin(x));
end
```

Note that this problem has only 18 variables and only a single constraint. LINDO looks at 48,619 subproblems, taking about 20 minutes on a Sun Sparc workstation, before deciding the optimal objective is 16. LINGO on a 16MHz 386 PC (with math coprocessor) looks at 48,000+ subproblems and takes about five hours. CPLEX on a Sun SPARC 10 takes about 50 seconds to examine 61,497 subproblems (counting those that are fathomed without solving the LP). The 100 variable version of this problem would take about 10^{29} subproblems or about 3×10^{18} years (at 1000 subproblems per second). Luckily, most problems take far less time.

Exercise 5 (Optional) *Solve the following problem by the branch and bound algorithm. For convenience, always select x_1 as the branching variable when both x_1 and x_2 are fractional.*

$$\begin{array}{ll} \textit{Maximize} & x_1 + x_2 \\ \textit{subject to} & 2x_1 + 5x_2 \leq 16 \\ & 6x_1 + 5x_2 \leq 30 \\ & x_1, x_2 \geq 0 \textit{ and integer.} \end{array}$$

Exercise 6 (Optional) *Repeat the preceding exercise assuming that x_1 only is restricted to integer values.*

Exercise 7 (Optional) *Consider the following* cargo-loading *problem, where five items are to be loaded on a vessel. The weights w_i and the volume v_i per unit of the different items as well as their corresponding values r_i are tabulated as follows.*

Item i	w_i	v_i	r_i
1	5	1	4
2	8	8	7
3	3	6	6
4	2	5	5
5	7	4	4

The maximum cargo weight and volume are given by $W = 112$ and $V = 109$, respectively. It is required to determine the most valuable cargo load in discrete units of each item. Formulate the problem as an integer program and solve by LINDO.

Michael A. Trick
Thu Oct 26 12:05:11 EDT 1995