Optimisation and Operations Research

Lecture 19: Branch and Bound, part II

Matthew Roughan
<matthew.roughan@adelaide.edu.au>
http://www.maths.adelaide.edu.au/matthew.roughan/notes/OORII/

School of Mathematical Sciences,
University of Adelaide

August 13, 2019

Section 1

Branch and Bound, Part II

Branch and Bound Example

Example (Knapsack problem)

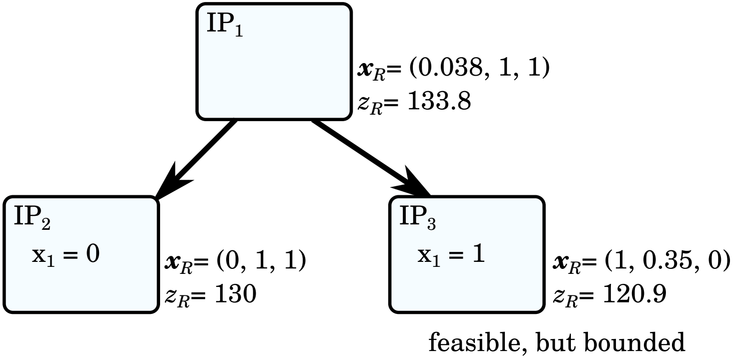
Consider the binary linear program

$$\begin{array}{lcrcrcrcl} \max z & = & 100x_1 & + & 60x_2 & + & 70x_3 & & \\ \text{s.t.} & & 52x_1 & + & 23x_2 & + & 35x_3 & \leq & 60 \\ & & x_1 & & & & & \in & \{0,1\} \\ & & & & x_2 & & & \in & \{0,1\} \\ & & & & & & x_3 & \in & \{0,1\} \end{array}$$

This is a knapsack problem where no more than 1 of each item is to be packed.

NB: the relative merit of the items, is $(\frac{100}{52}, \frac{60}{23}, \frac{70}{35})$ so a Greedy algorithm would get the solution $(1, 0, 0)$ with a value of $z = 100$.

Branch and Bound Example



Branch and Bound

Example (continued)

The relaxation of IP_1 has optimal solution

$$z^* = 133\frac{44}{52} \quad \text{at} \quad \mathbf{x}^* = \left(\frac{2}{52}, 1, 1\right)^T.$$

Since the first entry is non-integral, this cannot be the optimal solution of the (ILP) and so we branch on x_1 as follows

(a) (IP_2) : (IP_1) with $x_1 = 0$

(b) (IP_3) : (IP_1) with $x_1 = 1$.

So the list of problems is $\mathcal{L} = \{IP_2, IP_3\}$

Branch and Bound

Example (continued)

The relaxation of IP_2 has optimal solution

$$z^* = 130 \quad \text{at} \quad \mathbf{x}^* = (0, 1, 1)^T .$$

This is integer, feasible, so it is a viable solution to the original ILP. Thus we store its objective value

$$z_{ip} = z^* = 130$$

and this branch is considered fathomed.

Branch and Bound

Example (continued)

The relaxation of IP_3 has optimal solution

$$z^* = 120\frac{20}{23} \quad \text{at} \quad \mathbf{x}^* = \left(1, \frac{8}{23}, 0\right)^T.$$

This is non-integral, but z^* (the upper bound for IP_3 obtained from the relaxation) is already below $z_{ip} = 130$, so this solution is bounded, and hence fathomed.

But it's interesting to consider what we would have done if we tried to solve IP_3 *before* IP_2. Then we would have branched on this case, and had a longer list of problems to solve!

Branch and Bound on ILPs

Theorem

Assume we solve the relaxation of a subproblem IP_k, and get a solution $\mathbf{x}^*$, and we choose to branch on the ith variable at $c = x_i^*$, i.e., the branch is segmented along the value obtained from the relaxed LP.

Then instead of adding the constraint $x_i \leq \lfloor c \rfloor$ to the left branch, we can equivalently add the constraint $x_i = \lfloor c \rfloor$.

- This is an advantage because it reduces the dimension of the search space (x_i is no longer free), so as we progress down the tree, the problems become simpler to solve.
- We have a similar result for the right-hand branch.

Branch and Bound on ILPs Proof (1)

Pseudo-proof

Take integer program ILP_0

$$\mathbf{z}^* = \max\{\mathbf{c}^T\mathbf{x} \,|A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq 0, \mathbf{x} \in \mathbb{Z}^n\}$$

and its relaxed linear program LP_0

$$\mathbf{z}_R^* = \max\{\mathbf{c}^T\mathbf{x} \mid A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq 0, \mathbf{x} \in \mathbb{R}^n\}$$

with solution

$$\mathbf{x}_R^* = (x_1^0, x_2^0, \ldots, x_n^0)^T$$

Branch on the first variable at $c = x_1^0$, *i.e.*, create two new subproblems with search regions

$$R_1 = R_0 \cap \{x_1 \leq i\} \quad \text{and} \quad R_2 = R_0 \cap \{x_1 \geq i+1\}$$

where $i = \lfloor c \rfloor$.

Assume the optimal solution of LP_1 is strictly less than i, and prove there is instead an optimal solution where the solution is $=$ to i

Branch and Bound on ILPs Proof (2)

Suppose that the optimal solution $\mathbf{x}_R^*$ to (LP_1) has $x_1^* < i$.

(a) We show that there is some $\lambda \in (0,1)$ such that $\mathbf{x} = \lambda \mathbf{x}^0 + (1-\lambda)\mathbf{x}^*$ satisfies the following:
   (i) The value of x_1 in $\mathbf{x}$ equals i;
   (ii) $\mathbf{x}$ is feasible for (LP_0);
   (iii) The z-value for $\mathbf{x}$ is less than or equal to the z-value for $\mathbf{x}^*$.

If we establish (a), then:

(b) We can conclude that there is at least one optimal solution to (LP_1) with $x_1 = i$.

(c) This means we can set $x_1 = i$ in (LP_1), and

Similarly there is at least one optimal solution to (LP_2) with $x_1 = i + 1$, and so we can set $x_1 = i + 1$ in (LP_2)

Then this is true for each (LP_i) by induction.

Branch and Bound on ILPs Proof (3)

Pseudo-proof (cont)
Establishing (a):

$$\begin{aligned} \mathbf{x}^{*(0)} &= \text{solution to } LP_0 \\ \mathbf{x}^{*(1)} &= \text{solution to } LP_1 \end{aligned}$$

and by assumptions $x_1^{*(0)} > i$ and $x_1^{*(1)} < i$. Now take a point between the two solutions

$$\mathbf{x} = \lambda \mathbf{x}^{*(0)} + (1-\lambda)\mathbf{x}^{*(1)}$$

where, by the Intermediate Value Theorem, there exists $\lambda \in (0,1)$ such that

$$x_1 = i$$

that is, we can choose an intermediate point where the first value is on the boundary.

Branch and Bound on ILPs Proof (picture)

Branch and Bound on ILPs Proof (4)

- We know LPs are convex, and that $\mathbf{x}^{*(0)}, \mathbf{x}^{*(1)} \in R_0$, so we know $\mathbf{x} \in R_0$
- $\mathbf{x}$ also satisfies $x_1 \leq i$, and hence $\mathbf{x} \in R_1$, as that was the only extra constraint on R_1
- We added an extra constraint in going from LP_0 to LP_1, so the objective function cannot increase, *i.e.*, for all $\mathbf{x} \in R_1$
$$z(\mathbf{x}) \leq z\left(\mathbf{x}^{*(0)}\right)$$
- However, we started by assuming that $\mathbf{x}^{*(1)}$ is optimal in R_1, so
$$z(\mathbf{x}) \leq z\left(\mathbf{x}^{*(1)}\right)$$
- The objective is linear, so for any λ
$$z(\mathbf{x}) = z\left(\lambda \mathbf{x}^{*(0)} + (1-\lambda)\mathbf{x}^{*(1)}\right) = \lambda z\left(\mathbf{x}^{*(0)}\right) + (1-\lambda) z\left(\mathbf{x}^{*(1)}\right)$$
- The only way the three statements can be valid is if
$$z(\mathbf{x}) = z\left(\mathbf{x}^{*(1)}\right)$$
that is, if $\mathbf{x}$ is also optimal for LP_1

Example ILP

Example

Consider the Integer Linear Program

$$\begin{array}{rrcrcl} \max z & = & x & + & y & \\ \text{s.t.} & & -x & + & 2y & \leq & 8 \\ & & 23x & + & 10y & \leq & 138 \\ & & & & x, y & \in & \mathbb{Z}^+ \end{array}$$

Let's see what AMPL/lpsolve does

INPUT:

```
# example from lecture 18
var x integer >=0;
var y integer >=0;

maximize z: x + y;

subject to c1:   -x +  2*y <= 8;
subject to c2: 23*x + 10*y <= 138;
```

OUTPUT:

```
LP_SOLVE 4.0.1.0: optimal, objective 8
8 simplex iterations
9 branch & bound nodes: depth 4
```

SOLUTION: $x = 3, y = 5, z = 8$

Example Branching

Example (continued)

Relaxed solution to the original LP is $(x^*, y^*)^T = \left(3\frac{1}{2}, 5\frac{3}{4}\right)^T$
This is not integer feasible, so we select one these (non-integer) variables, such as $x^* = 3\frac{1}{2}$ and branch as follows

(a) $x \leq 3$

(b) $x \geq 4$

Note that we now have 2 regions, which are mutually exclusive. This forced dichotomy is the "branching" part of "branch-and-bound". Here we have chosen x as the branching variable (and have "cut off" a strip of x values $3 < x < 4$). This helps force our solution of the (relaxed) (LP) to be integer and hopefully towards the solution of the original (ILP).

Example Branching

Example (continued)

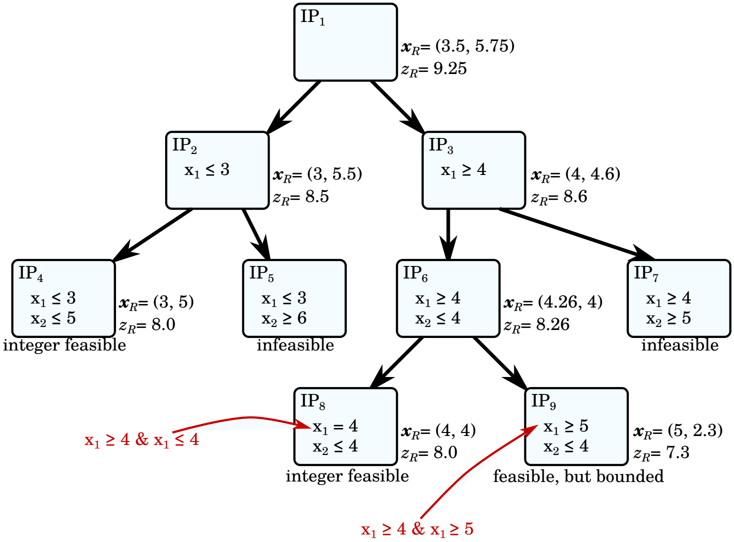
After 1st branching, we have 2 (LP) problems to solve:

$$\begin{array}{llcrcrcr}
(LP_2) & \max z & = & x & + & y & & \\
\text{s.t.} & & - & x & + & 2y & \leq & 8 \\
 & & & 23x & + & 10y & \leq & 138 \\
 & & & x & & & \leq & 3 \\
 & & & & & x, y & \geq & 0
\end{array}$$

and

$$\begin{array}{llcrcrcr}
(LP_3) & \max z & = & x & + & y & & \\
\text{s.t.} & & - & x & + & 2y & \leq & 8 \\
 & & & 23x & + & 10y & \leq & 138 \\
 & & & x & & & \geq & 4 \\
 & & & & & x, y & \geq & 0
\end{array}$$

Example B&B



IP_1
$\boldsymbol{x}_R$= (3.5, 5.75)
z_R= 9.25
IP_2 $x_1 \le 3$
$\boldsymbol{x}_R$= (3, 5.5)
z_R= 8.5
IP_3 $x_1 \ge 4$
$\boldsymbol{x}_R$= (4, 4.6)
z_R= 8.6
IP_4 $x_1 \le 3$ $x_2 \le 5$
$\boldsymbol{x}_R$= (3, 5)
z_R= 8.0
integer feasible
IP_5 $x_1 \le 3$ $x_2 \ge 6$
infeasible
IP_6 $x_1 \ge 4$ $x_2 \le 4$
$\boldsymbol{x}_R$= (4.26, 4)
z_R= 8.26
IP_7 $x_1 \ge 4$ $x_2 \ge 5$
infeasible
$x_1 \ge 4$ & $x_1 \le 4$
IP_8 $x_1 = 4$ $x_2 \le 4$
$\boldsymbol{x}_R$= (4, 4)
z_R= 8.0
integer feasible
IP_9 $x_1 \ge 5$ $x_2 \le 4$
$\boldsymbol{x}_R$= (5, 2.3)
z_R= 7.3
feasible, but bounded
$x_1 \ge 4$ & $x_1 \ge 5$

Better Branching

Example (continued)

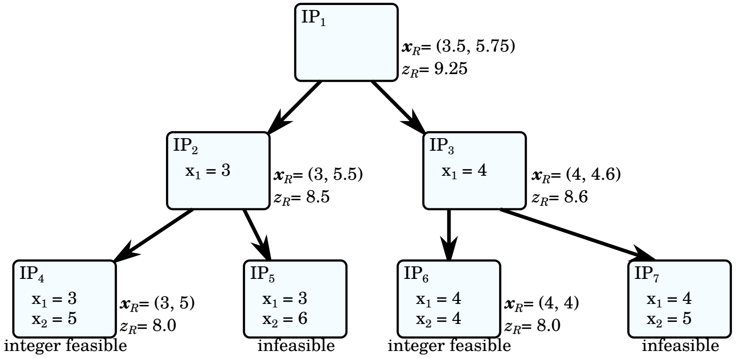
We still have two 2 (LP) problems to solve, but they are "tighter"

$$\begin{array}{llcrcrcrl}
(LP_2) & \max z & = & x & + & y & & & \\
\text{s.t.} & & - & x & + & 2y & \leq & 8 & \\
 & & & 23x & + & 10y & \leq & 138 & \\
 & & & x & & & = & 3 & \text{note the equality} \\
 & & & & & x, y & \geq & 0 &
\end{array}$$

and

$$\begin{array}{llcrcrcrl}
(LP_3) & \max z & = & x & + & y & & & \\
\text{s.t.} & & - & x & + & 2y & \leq & 8 & \\
 & & & 23x & + & 10y & \leq & 138 & \\
 & & & x & & & = & 4 & \text{note the equality} \\
 & & & & & x, y & \geq & 0 &
\end{array}$$

Example B&B



- Note that now we don't have to solve as many sub-problems as previous case
- In fact, we only need to go two steps down any branch, at worst

Branch and Bound's Major Problem

The order of problem selection matters!

- I have just done them in the order added
- could do other simple approachs: *e.g.*, depth first or breadth first
- other approaches
  - **Best bound rule**:
    We partition the subset with the lowest bound, hoping that this gives the best chance of an optimal solution and of being able to discard other, larger, subsets by fathoming.
  - **Newest bound rule**:
    We partition the most recently created subset, breaking ties with the best bound rule. This has book-keeping advantages, in that we don't need to jump around the tree too often. It can also save some computational effort involved in calculating bounds.

But there is no universal "best" order.

Branch and Bound

- B&B can be made even faster
  - sometimes we know something specific about the problem that helps derive better bounds
  - we don't solve each relaxation from scratch – we already have a starting point (*e.g.*, see sensitivity analysis when we add a constraint)
- B&B is a very general algorithm
  - as described above we seek the optimum
  - can also be used as part of a heuristic
- different strategies available for each step above
  - can use heuristics inside B&B
  - pre-processing of the problem can be good
- no single strategy stands out as best for all problems
  - but sometimes we can exploit properties of a particular problem to do better

Branch and Bound

More information and examples of Branch and bound can be found at
http://mat.gsia.cmu.edu/orclass/integer/integer.html
http://en.wikipedia.org/wiki/Branch_and_bound
http://mathworld.wolfram.com/BranchandBoundAlgorithm.html

An instructive paper is
http://www.rpi.edu/~mitchj/papers/leeejem.html

A list of implementations can be found at
http://www.mat.univie.ac.at/~neum/glopt/software_g.html

Take Aways

- We have a variety of approaches to attack ILPs
  - heuristics
    - ★ simple to program
    - ★ very problem dependent
    - ★ often quite fast
  - B&B
    - ★ more general (solves general ILPs)
    - ★ harder work to program (not too much harder)
    - ★ potentially slow for big problems
  - others ...
- But there is no "one size fits all" solution

Further reading I