Branch & Bound

Lorenzo Castelli, Università degli Studi di Trieste.

UNIVERSITÀ
DEGLI STUDI DI TRIESTE

Divide and conquer

Branch and bound uses a "divide and conquer" approach to explore the set of feasible integer solutions. However, instead of exploring the entire feasible set, it uses bounds on the optimal value to avoid certain parts of the set of feasible integer solutions.

Consider the problem

$$\boldsymbol{z} = \max\{\boldsymbol{cx} : \boldsymbol{x} \in \boldsymbol{S}\}$$

How can we break the problem into a series of subproblems that are easier, solve the smaller problems, and then put the information together again to solve the original problem?

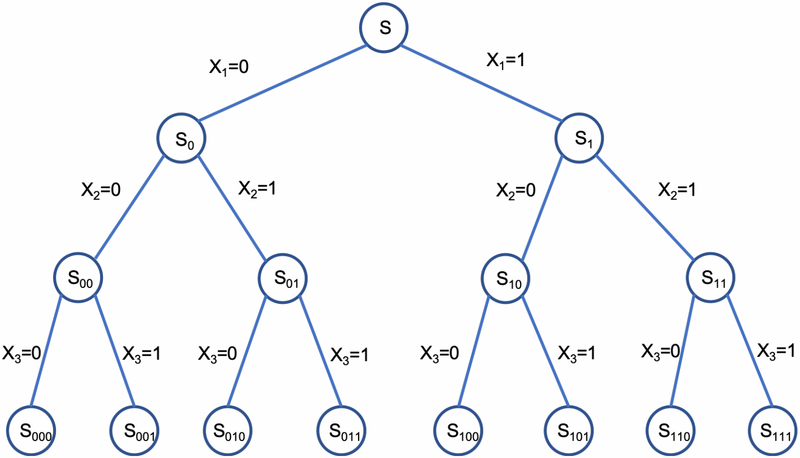
Proposition

Proposition 1

Let $S = S_1 \cup \ldots \cup S_K$ be a decomposition of S into smaller sets, and let $z^k = \max\{cx : x \in S_k\}$ for $k = 1, \ldots, K$. Then $z = max_k z^k$

A typical way to represent such a divide and conquer approach in via an enumeration tree. For instance if $S \subseteq \{0, 1\}^3$, we might construct the enumeration tree shown in slide (3).

Enumeration tree



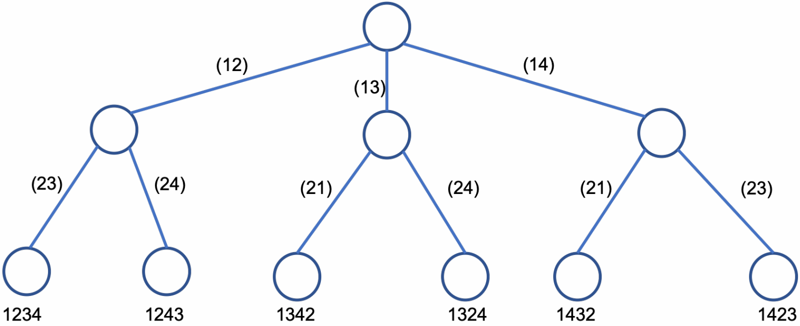
Enumeration tree

We first divide S into $S_0 = \{x \in S : x_1 = 0\}$ and $S_1 = \{x \in S : x_1 = 1\}$, then $S_{00} = \{x \in S_0 : x_2 = 0\} = \{x \in S : x_1 = x_2 = 0\}$, $S_{01} = \{x \in S_0 : x_2 = 1\} = \{x \in S : x_1 = 0, x_2 = 1\}$, and so on. Note that a leaf of the tree $S_{i_1 i_2 i_3}$ is nonempty if and only if $x = (i_1, i_2, i_3)$ is in S. Thus the leaves of tree correspond precisely to the point of $\{0, 1\}^3$ that one would examine if one carried out complete enumeration. By convention the tree is drawn upside down with the root at the top.

Enumeration tree for the TSP

- Enumeration of all the tours of the TSP
- First, we divide S the set of all tours on 4 cities into $S_{(12)}, S_{(13)}, S_{(14)}$ where $S_{(ij)}$ is the set of all tours containing arc (i, j).
- Then $S_{(12)}$ is divided again in $S_{(12)(23)}$ and $S_{(12)(24)}$, and so on.
- At the first level we have arbitrarily chosen to branch on the arcs leaving node 1, and at the second level on the arcs leaving node 2 that do not immediately create a subtour with the previous branching arc.
- The six leaves of the tree correspond to the $(n-1)!$ tours, where $i_1 i_2 i_3 i_4$ means that the cities are visited in the order $i_1 i_2 i_3 i_4$ respectively.
- This is an example of multiway as opposed to binary branching, where a set can be divided into more than two parts.

Enumeration tree



Implicit enumeration – motivation

- Complete enumeration is totally impossible for most problems as son as the number of variables in an integer program, or nodes in a graph exceeds 20 or 30. So we need to do more than just divide indefinitely.
- How can we use some bounds on the values of $\{z^k\}$ intelligently?
- First, how can we put together bound information?

Implicit enumeration - bounds

Proposition

Let $S = S_1 \cup \ldots \cup S_K$ be a decomposition of S into smaller sets, and let $z^k = \max\{cx : x \in S_k\}$ for $k = 1, \ldots, K$, $\overline{z}^k$ be an upper bound on z^k and $\underline{z}^k$ be a lower bound on z^k. Then

$\overline{z} = \max_k \overline{z}^k$ is an upper bound on z

$\underline{z} = \max_k \underline{z}^k$ is a lower bound on z

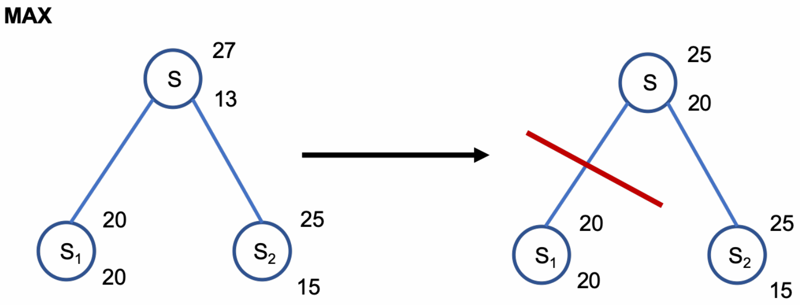
What can be deduced about lower and upper bounds on the optimal value z and which sets need further examination in order to find an optimal value?

Pruned by optimality

- In slide (10) we show a decomposition of S into two sets S_1 and S_2 as well as upper and lower bounds on the corresponding problems.
- We note that $\bar{z} = \max_k \bar{z}^k = \max\{20, 25\} = 25$.
- We note that $\underline{z} = \max_k \underline{z}^k = \max\{20, 15\} = 20$.
- We observe that as lower and upper bound on z^1 are equal, $z^1 = 20$, and there is no further reason to examine the set S_1.

The branch S_1 of the enumeration tree can be pruned by optimality.

Pruned by optimality

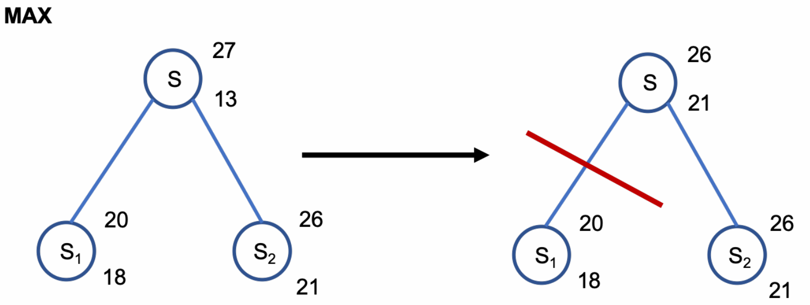


Pruned by bound

- In slide (12) we show a decomposition of S into two sets S_1 and S_2 as well as upper and lower bounds on the corresponding problems.
- We note that $\overline{z} = \max_k \overline{z}^k = \max\{20, 26\} = 26$.
- We note that $\underline{z} = \max_k \underline{z}^k = \max\{18, 21\} = 21$.
- We observe that as the optimal value has value at least 21, and the upper bound $\overline{z}^1 = 20$, no optimal solution can lie in the set S_1.

The branch S_1 of the enumeration tree can be pruned by bound.

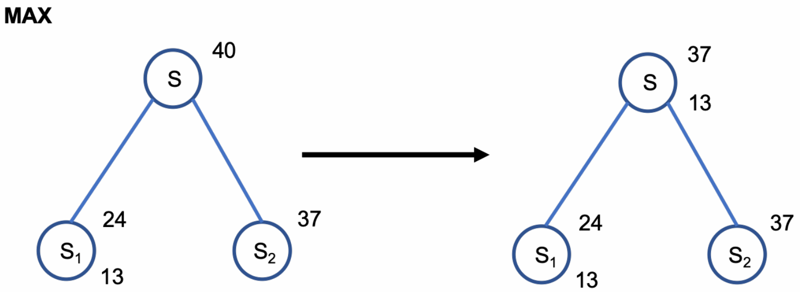
Pruned by bound



No pruning possible

- In slide (14) we show a decomposition of $\boldsymbol{S}$ into two sets $\boldsymbol{S_1}$ and $\boldsymbol{S_2}$ as well as upper and lower bounds on the corresponding problems.
- We note that $\overline{z} = \max_k \overline{z}^k = \max\{24, 37\} = 37$.
- We note that $\underline{z} = \max_k \underline{z}^k = \max\{13, -\} = 13$.
- Here no other conclusions can be drawn and we need to explore both sets $\boldsymbol{S_1}$ and $\boldsymbol{S_2}$ further.

No pruning possible



Upper and lower bounds

How the bounds are to be obtained?

- The primal (lower) bounds are provided by feasible solutions.
- The dual (upper) bounds are provided by relaxation or duality.

Building an implicit enumeration algorithm based on the above ideas is now in principle a fairly straightforward task. There are, however, many questions that must be addressed before such an algorithm is well-defined.

How to build a B&B algorithm

- What relaxation or dual problem should be used to provide an upper bound?
- How should one choose between a fairly weak bound that can be calculated very rapidly and a stronger bound whose calculation takes a considerable time?
- How should a feasible region be separated into smaller regions $\boldsymbol{S} = \boldsymbol{S}_1 \cup \ldots \cup \boldsymbol{S}_K$?
- Should one separate into two or more parts?
- Should one use a fixed a priori rule for dividing up the set, or should the division evolve as a function of the bounds and solutions obtained en route?
- In what order should the subproblems be examined?
- Typically there is a list of active problems that have not yet been pruned. Should the next one be chosen on a basis of last-in first-out, of best/largest upper bound first, or of some totally different criterion?

B&B – an example

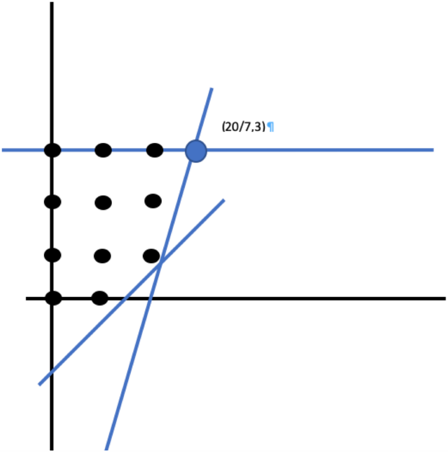
Let's consider the following linear integer program

$$
\begin{aligned}
z = \max 4x_1 - x_2 \\
7x_1 - 2x_2 \leq 14 \\
x_2 \leq 3 \\
2x_1 - 2x_2 \leq 3 \\
x \in \mathbb{Z}_+^2
\end{aligned}
$$

Bounding

- To obtain a first upper bound, we solve the linear programming relaxation in which the integrality constraints are dropped.
- We obtain an upper bound $\overline{z} = \mathbf{59/7} = \mathbf{8.43}$ and a non-integral solution $(\mathbf{20/7}, \mathbf{3})$.
- Is there any straightforward way to find a feasible solution? In this case it is $(\mathbf{0}, \mathbf{0})$ and $\underline{z} = \mathbf{0}$.
- By convention, if no feasible solution is yet available, we take as lower bound $\underline{z} = -\infty$

LP relaxation



Branching

Now because $\underline{z} < \overline{z}$, we need to divide or branch.

- How should we split up the feasible region?.

One simple idea is to choose an integer variable which is nonzero and fractional in the linear programming solution, and split the problem into two about this fractional value. If $x_j = \overline{x_j} \notin \mathbb{Z}^1$, one can take:

$$S_1 = S \cap \{x : x_j \leq \lfloor \overline{x}_j \rfloor\}$$
$$S_2 = S \cap \{x : x_j \geq \lceil \overline{x}_j \rceil\}$$

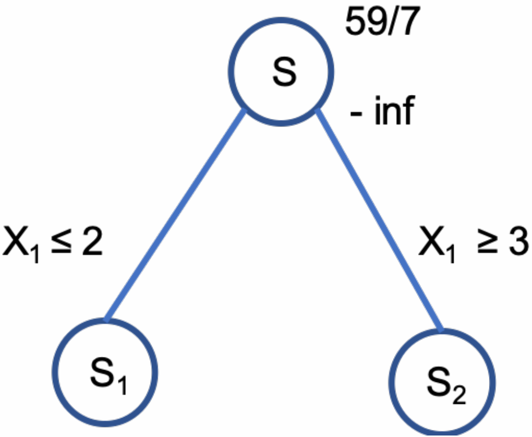
It is clear that $S = S_1 \cup S_2$ and $S_1 \cap S_2 = \emptyset$.

Branching

Another reason for this choice is that the solution $\bar{\boldsymbol{x}}$ of $\boldsymbol{LP}(\boldsymbol{S})$ is not feasible in either $\boldsymbol{LP}(\boldsymbol{S_1})$ or $\boldsymbol{LP}(\boldsymbol{S_2})$. This implies that (if there are not multiple LP solutions) $\max\{\bar{z}^1, \bar{z}^2\} < \bar{z}$, so the upper bound will strictly decrease.

Following this idea, as $\bar{x}_1 = 20/7 \notin \mathbb{Z}^1$, we take $S_1 = S \cap \{x : x_1 \leq 2\}$ and $S_2 = S \cap \{x : x_1 \geq 3\}$. The subproblems (nodes) that must still be examined are called active.

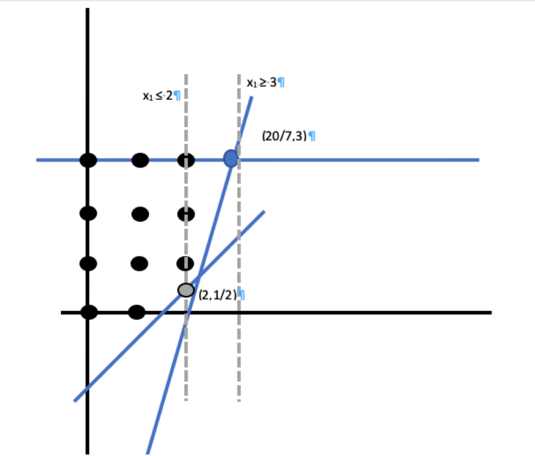
LP relaxation



Next steps - Node S_1

- *Choosing a Node.* The list of active problems (nodes) to be examined now contains S_1, S_2. We arbitrarily choose S_1.
- *Reoptimizing.* We solve the linear relaxation of S_1. The results is $\overline{z}^1 = 15/2 = 7.5$, and $(\overline{x}_1^1, \overline{x}_2^1) = (2, 1/2)$.
- *Branching.* S_1 cannot be pruned, so using the same branching rule as before, we create two new nodes $S_{11} = S_1 \cap \{x : x_2 \leq 0\}$ and $S_{12} = S_1 \cap \{x : x_2 \geq 1\}$ and add them to the node list.

LP relaxation



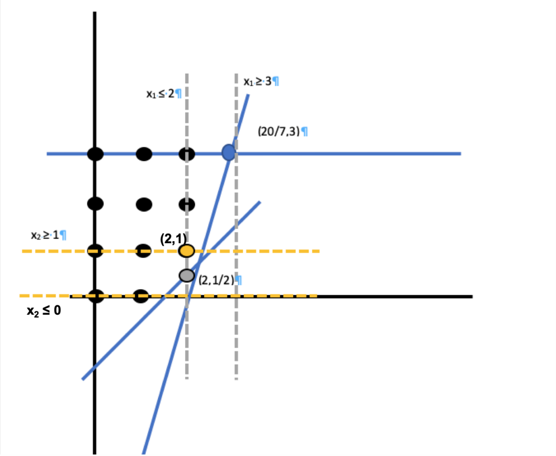
Next steps - Node S_2

- *Choosing a Node.* The list of active problems (nodes) to be examined now contains $\boldsymbol{S_2, S_{11}, S_{12}}$. We arbitrarily choose $\boldsymbol{S_2}$, and remove it by the node list.
- *Reoptimizing.* We solve the linear relaxation of $\boldsymbol{S_2}$. We see that the linear program is infeasible. Hence node $\boldsymbol{S_2}$ is pruned by infeasibility.

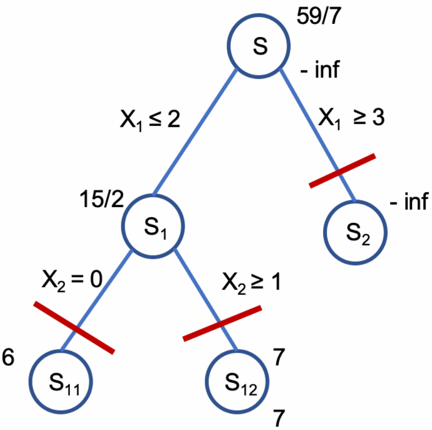
Next steps - Node S_{12}

- *Choosing a Node.* The node list now contains S_{11}, S_{12}. We arbitrarily choose S_{12} and remove it from the list.
- *Reoptimizing.* $S_{12} = S \cap \{x : x_1 \leq 2, x_2 \geq 1\}$. The resulting linear program has optimal solution $\overline{x}^{12} = (2, 1)$ and $\overline{z}_{12} = 7$. Since $\overline{x}^{12}$ is integer $\overline{z}^{12} = z^{12} = 7$.
- *Updating the incumbent.* As the solution of $LP(S_{12})$ is integer, we update the value of the best feasible solution found $\underline{z} \leftarrow max\{\underline{z} = 0, 7\}$ and store the corresponding solution $(2, 1)$. S_{12} is now pruned by optimality.

LP relaxation



LP relaxation



Next steps - Node S_{11}

- *Choosing a Node.* The node list now contains only S_{11}.
- *Reoptimizing.* $S_{11} = S \cap \{x : x_1 \leq 2, x_2 \leq 0\}$. The resulting linear program has optimal solution $\overline{x}^{11} = (3/2, 0)$ and $\overline{z}^{11} = 6$. Since $\underline{z} = 7 > \overline{x}^{11} = 6$ the node is pruned by bound.
- *Choosing a node.* As the node list is empty, the algorithm terminates. The incumbent solution $x = (2, 1)$ with $z = 7$ is optimal.

Pruning reasons

Pruning by optimality. $z^t = \max\{cx : x \in S_t\}$ has been solved.

Pruning by bound. $\overline{z}^t \leq \underline{z}$.

Pruning by infeasibility. $S_t = \emptyset$.