COMP 6651 / Winter 2022
Dr. B. Jaumard

Lecture on Branch-and-Bound Methods

March 25, 2022

Outline

Definitions

For a given optimization problem $\max / \min\{z(x) : x \in S\}$

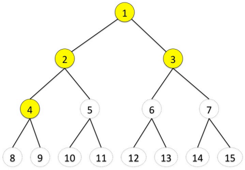
- Feasible Solution is a point in the search space of the optimization problem that satisfies all the constraints
- Optimal Solution is a feasible solution with the best value of the objective function
- The optimal value is unique, but the optimal solution is not necessarily unique
- In practice, usually interested in finding one optimal solution, but not necessarily all of them

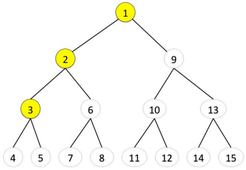
Requirements of a Branch-and-Bound Method

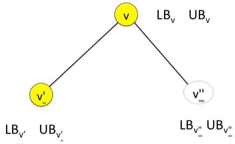
1. A lower bound on the optimal value
2. An upper bound on the optimal value
3. A separation scheme: how to split the current problem into subproblems?
4. A branching scheme: what is the next node to be explored in the search tree?
5. An exploration scheme: in which order the subproblems are investigated?

Search tree

- Exploration of the search tree
  - Depth First Search (DFS)
  - Best First Search
  - Breadth First Search
- Shape of the search tree
  - Binary search tree
  - Multi-tree
  - Load balanced search tree
  - ...







Notations

Maximization problem

$$\max cx$$

subject to:

$$Ax \leq b$$
$$x \geq 0$$

Minimization problem

$$\min cx$$

subject to:

$$Ax \geq b$$
$$x \geq 0$$

Assume $A = (a_{ij})$ and $b = (b_j)$.

Note that if all $a_{ij} \geq 0$ and $b_j \geq 0$

$\min\{cx : Ax \leq b, x \geq 0\}$ and $\max\{cx : Ax \geq b, x \geq 0\}$

are trivial problems: the optimal solution of the minimization problem is $x_j = 0$ for all j, and the maximization problem is unbounded.

Relaxation

- **Optimization problem** (P):

  $$z^\star = \max\{z(x) : x \in S\}, \text{ e.g., } z(x) = cx, S = \{x \in \mathrm{Z}^{\mathrm{n}}_{+} : \mathrm{Ax} \leq \mathrm{b}\},$$

  where c is an n-vector with integral coefficient and (A, b) is an $m \times (n+1)$ matrix with integral coefficients.

- **Relaxation** (R) **of** (P) is any maximization problem such that

  $$z_R^\star = \max\{z_R(x) : x \in S_R\},$$

  with the following two properties:

  $$\begin{aligned} & S \subseteq S_R \\ & z(x) \leq z_R(x) \qquad \text{for } x \in S \end{aligned}$$

Linear Programming (LP) Relaxation

- **Optimization problem** (P):

$$z^\star = \max\{z(x) = cx : Ax \leq b, x \in \{0,1\}^n\},$$

where c is an n-vector with integral coefficient and (A, b) is an $m \times (n+1)$ matrix with integral coefficients.

- **LP Relaxation of** (P)

$$z^\star_{\text{LP}} = \max\{z(x) = cx : Ax \leq b, 0 \leq x \leq 1\}.$$

Branch-and-Bound - Minimization problem (1/2)

- Objective function $z(x)$, variable vector $x = (x_1, x_2, \ldots, x_n)$
- Arbitrary node in the search tree, indexed by k
- Lower bound: $\underline{z}_k$ (usually obtained by solving **exactly** a relaxation of the problem)
- Upper bound: $\overline{z}_k$ (usually a feasible solution obtained by a heuristic)
- Update the incumbent value: $z_{\mathrm{BEST}} \leftarrow \min\{z_{\mathrm{BEST}}, \overline{z}_k\}$
- (Local) Optimality test: If $\underline{z}_k$ corresponds to a feasible solution and $z_R = z$, then $\underline{x}_k$ is an optimal solution of subproblem $\#k$.

Branch-and-Bound - Minimization problem - Pruning Tests (2/2)

- If subproblem #k has no feasible solution, prune subproblem #k
- If $\underline{z}_k > z_{\text{BEST}} = (\min_k \overline{z}_k)$, prune the subproblem #$k$ as it cannot contain a solution better than the incumbent one.

Branch-and-Bound Algorithm - Minimization Problem

$\underline{z} \leftarrow -\infty$; $\overline{z} \leftarrow z^{\text{BEST}}$ (using, e.g., a greedy heuristic)

The Assignment Problem: Definition

- n people to assign to execute n jobs, one person per job
  - each person is assigned to exactly one job
  - each job is assigned to exactly one person
- Cost c_{ij} when person i is assigned to job j

How to assign n people to n jobs so that the total cost of the assignment is as small as possible.

	job 1	job 2	job 3	job 4
person *a*	9	2	7	8
person *b*	6	4	3	7
person *c*	5	8	1	8
person *d*	7	6	9	4

The Assignment Problem: Lower Bounds

- Lower Bound 1
  - Sum of the smallest elements in each row of the matrix
  - Example: 2+3+1+4=10 (but not a feasible solution)
- Lower Bound 2
  - ILP formulation of the assignment problem

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

subject to:

$$\begin{cases} \sum_{j=1}^{n} x_{ij} = 1 & i = 1, 2, ..., n \\ \sum_{i=1}^{n} x_{ij} = 1 & j = 1, 2, ..., n \\ x_{ij} \in \{0, 1\} & i, j = 1, 2, ..., n \end{cases}$$

  - LP relaxation bound: $x_{ij} \in \{0, 1\} \rightarrow x_{ij} \in [0, 1]$

Levels 0 and 1 of the Search Tree

	job 1	job 2	job 3	job 4
person a	9	2	7	8
person b	6	4	3	7
person c	5	8	1	8
person d	7	6	9	4

- Level 0 Root: Node 0

Node 0
start
LB_0 = 2+3+1+4=10

- Level 1 Sons of the root

Node 1	Node 2	Node 3	Node 4
$a \rightarrow 1$	$a \rightarrow 2$	$a \rightarrow 3$	$a \rightarrow 4$
LB_1 = 9+3+1+4=17	LB_2 = 2+3+1+4=10	LB_3 = 7+4+5+4=20	LB_4 = 8+3+1+6=18

	job 1	job 2	job 3	job 4
person *a*	9	2	7	8
person *b*	6	4	3	7
person *c*	5	8	1	8
person *d*	7	6	9	4

- Level 0 Root: Node 0

Node 0
start
LB_0 =10

- Level 1 Sons of the root

Node 1	Node 2	Node 3	Node 4
$a \rightarrow 1$	$a \rightarrow 2$	$a \rightarrow 3$	$a \rightarrow 4$
LB_1 =17	LB_2 = 10	LB_3 = 20	LB_4 = 18

- Level 2 Sons of Node 2

Node 5	Node 6	Node 7
$b \rightarrow 1$	$b \rightarrow 3$	$b \rightarrow 4$
LB_5 =13	LB_6 = 14	LB_7 = 17

	job 1	job 2	job 3	job 4
person *a*	9	2	7	8
person *b*	6	4	3	7
person *c*	5	8	1	8
person *d*	7	6	9	4

- **Level 0** Root: Node 0

Node 0
start
$LB_0 = 10$

- **Level 1** Sons of the root

Node 1	Node 2	Node 3	Node 4
$a \rightarrow 1$	$a \rightarrow 2$	$a \rightarrow 3$	$a \rightarrow 4$
$LB_1 = 17$	$LB_2 = 10$	$LB_3 = 20$	$LB_4 = 18$

- **Level 2** Sons of Node 2

Node 5	Node 6	Node 7
$b \rightarrow 1$	$b \rightarrow 3$	$b \rightarrow 4$
$LB_5 = 13$	$LB_6 = 14$	$LB_7 = 17$

- **Level 3** Sons of Node 5

Node 8	Node 9
$c \rightarrow 3$	$c \rightarrow 4$
$LB_8 = 13$	$LB_9 = 25$
feasible solution	inferior solution

The Assignment Problem: Upper Bounds

So far, we have developed a B & B without the use of an upper bound, however, it is easy to compute one.

- Let $J = \{1, 2, \ldots, n\}$ be the set of available jobs
- Select a first person, say i_1 and assign him the job with the lowest cost j_1. $J \leftarrow J \setminus \{j_1\}$.
- Select another person with no job, and assign him the job in J with the lowest cost. Update J accordingly, and so on.
- Example: Select the persons in the lexicographical order

	job 1	job 2	job 3	job 4
person *a*	9	2	7	8
person *b*	6	4	3	7
person *c*	5	8	1	8
person *d*	7	6	9	4

Upper bound: 2 + 3 + 5 + 4 = 14

The Knapsack Problem: Definition

- n items of known weights w_i and values v_i, $i = 1, 2, \ldots, n$
- a knapsack of capacity W
- find the most valuable subset of items that fit in the knapsack

The Knapsack Problem: Upper Bound

- ILP formulation of the knapsack problem

$$\max \sum_{i=1}^{n} v_i x_i \quad \text{subject to:} \quad \begin{cases} \sum_{j=1}^{n} w_i x_i \leq W \\ x_i \in \{0,1\} \qquad i = 1, 2, ..., n \end{cases}$$

- LP relaxation bound: $x_i \in \{0,1\} \rightarrow x_i \in [0,1]$
  - Assume: $v_1/w_1 \geq v_2/w_2 \geq ... \geq v_n/w_n$ (after possibly re-indexing the variables). An optimal solution of the linear relaxation, hence an upper bound of the ILP knapsack:

$$x_j = 1 \text{ for } j = 1, 2, ..., r-1, \qquad x_r = \frac{W - \sum_{j=1}^{r-1} w_j}{w_r}$$

$$x_j = 0 \text{ for } j = r+1, ..., n,$$

where r is such that $\sum_{j=1}^{r-1} w_j \leq W$ and $\sum_{j=1}^{r} w_j > W$.

The Knapsack Problem: Lower Bound

Easy to devise a greedy heuristic that leads to a feasible vector.

Start with $x = (0, 0, ..., 0)$ and, iteratively fix one variable to 1 as long as the capacity is not exceeded.

The Knapsack Problem: Branching Scheme

Branching variable: the fractional variable in the optimal solution of the linear relaxation.

Why? Only variable that offers the possibility to improve the upper bounds in BOTH son nodes of the current node.

The Traveling Salesman Problem: Definition

- n cities and a traveling salesman who has to visit each city once and return to the first visited city.
- $c_{ij} =$ direct travel time from city i to city j
- find a (Hamiltonian) tour with the least total travel time

The Traveling Salesman Problem: Lower Bound 1

- For each city i, $1 \leq i \leq n$, find the sum s_i of the distances from city i to the two nearest cities
- Compute the sum s of these n numbers
- Divide the result by 2
- If all distances are integer values, round up the result to the nearest integer: $LB = \lceil s/2 \rceil$

The Traveling Salesman Problem: Lower Bound 2

Linear relaxation of the ILP formulation

$$x_{ij} = \begin{cases} 1 & \text{if city } j \text{ follows immediately city } i \text{ on the tour,} \\ 0 & \text{otherwise.} \end{cases}$$

$$\min \sum_{(i,j) \in \mathcal{A}} c_{ij} x_{ij}$$

$$\sum_{(i,j) \in \mathcal{A}} x_{ij} = 1 \qquad j \in V$$

$$\sum_{(i,j) \in \mathcal{A}} x_{ij} = 1 \qquad i \in V$$

$$\sum_{(i,j) \in \mathcal{A} : i \in U, j \in V \setminus U} x_{ij} \geq 1 \qquad U \subset V, 2 \leq |U| \leq |V| - 2$$

$$x_{ij} \in \{0, 1\} \qquad i, j \in V$$

The Traveling Salesman Problem: Upper Bound

Any Hamiltonian tour provides an upper bound