Branch and Bound Algorithms

Backtracking

We hope many NP problems, by employing our techniques.

- Exhaustive search: generate all candidate solutions and then identify one that satisfies the required properties.

Backtracking is an intelligent version of this.

We construct a tree called the space tree.

- Tree paths: promising and non-promising paths.

N Queen Problem

The problem is to place n queens on an nxn board so that no two queens attack each other by:

- Same row, same column, or same diagonal.

Diagram of queen placements:

1 2 3 4

1 2 3 4

(queen placement diagrams)
Hamiltonian Circuit Problem

Subset Sum Problem

Given set $S = \{s_1, s_2, \ldots, s_n\}$ of positive integers, find a subset $S'$, whose sum is equal to a given integer $d$.

$S = \{1, 2, 3, 4, 5\}$ and $d = 9$.

$S' = \{1, 2, 6\}$

[Diagram of a tree structure with nodes and branches representing the subset sum process.]
Branch and Bound Algos.

deal with an optimization problem
feasible solution - satisfies all the problem constraints
optimal solution - a feasible solution, best value of the objective function.

* At every node of the s-tree, a bound is the
  best value of the objective function on any solution
  that can be obtained by fixing components to the
  partial soln represented by the node.

* The value of the best solution so far,
  the value of the node's bound is not better than the
  best bound so far.

* The node represents no feasible solution. Because
  all constraints of the problem are already violated

* The subset of feasible values represented by the node
  contains a single point \( \Rightarrow \) no further divided
n people to be assigned to n jobs - one person/job

Find an assignment with the minimal cost.

Lower bound: Sum of lowest in each row

2 + 3 + 1 + 4 = 10 → This is not the cost of an optimal solution but just the lower bound

Even the optimal solution cannot be better.

```
Start
0 → 1
9 + 3 + 1 + 4 = 17

a → 2
2 + 3 + 1 + 4 = 10

b → 3
5 → 1
11 → 9

2 → 3 + 4
3 → 3
4 → 3

(Not Optimal) 234
```
Knapsack Problem

N items of known weight $w_i$ and value $v_i$ ($i = 1, 2, ..., n$)

Capacity = $W$

$v_i/w_i > v_j/w_j > v_3/w_3 > ... > v_n/w_n$

Upper bound

$W_b = W + (W - W)(v_i/v_i + 1)$

<table>
<thead>
<tr>
<th>Item</th>
<th>Weight</th>
<th>Value</th>
<th>Value/Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>$40</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>$42</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>$25</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>$12</td>
<td>4</td>
</tr>
</tbody>
</table>

At the root node no item is chosen

$W_b = 0 + 10(10) = 100$

$W = 0, v = 0$

$W = 10, v = 100$

$W = 4, v = 40$

$W = 76$

$W = 2$

$W = 11$

$X$

$W = 9, v = 40$

$W = 70$

$W = 2$

$W = 9, v = 45$

$W = 69$

$W = 11$

$W = 4$

$W = 64$

$W = 60$

$W = 60$

$W = 60$

$W = 64$

$W = 60$
TSP

\[ L_b = \frac{\sum s_{ij}}{2} \]

\[ s_{ij} = \text{Distance from city } i \text{ to city } j \]

\[ \text{The shortest path} \]

\[ \text{Min. Max algo} \{ \}

\[ \alpha, \beta \text{ - Annxin} \]