UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa

Tree search and branch and bound

Jose M Sallan

Introduction to metaheuristics for combinatorial problems

Outline

1. Tree search optimization
2. Implementing branch and bound
3. A branch and bound for the TSP
4. Concluding remarks

1. Tree search optimization
2. Implementing branch and bound
3. A branch and bound for the TSP
4. Concluding remarks

Tree search optimization

Branch and bound

- Systematic enumeration of candidate solutions through solution space search
- Search is carried defining subsets of solutions in a rooted tree structure with full set at the root **(branch)**
- To discard some subsets of solutions, it is convenient to find a **bound** for each subset

First proposed by Land and Doig for discrete programming (e. g., linear integer programming)

There are other algorithms using tree search (e. g., branch and cut)

1 Tree search optimization

2 Implementing branch and bound

3 A branch and bound for the TSP

4 Concluding remarks

Branching

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa

Branching: a rule to find a partition of a subset of the solution space: each element of the subset is included in one and only one of the subsets of the partition.

A **candidate solution** is obtained when a subset has only one element.

A possible way of branching: **picking or not picking** an element from a list of candidate solution elements (KP: items, TSP: nodes or edges).

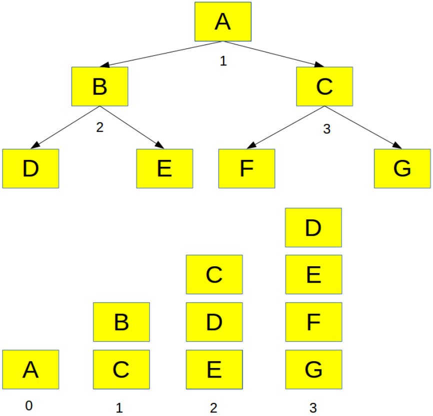
Branching order

Several branching orders can be implemented:

- **FIFO:** oldest subset is branched
- **LIFO:** newest subset is branched
- **Lower cost:** subset of lower cost is branched

Branching order

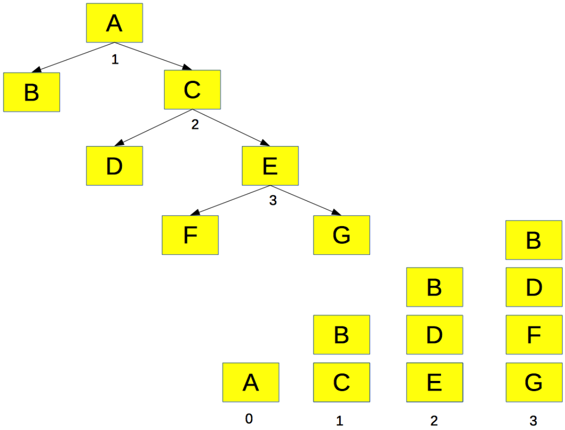
FIFO



Queue of candidate nodes

Branching order

LIFO



Stack of candidate nodes

Bounding

Search space can be reduced finding a **bound** of each subset:

- MAXimisation problems $\Rightarrow$ upper bound
- MINimisation problems $\Rightarrow$ lower bound

All subsets with a bound worse than the best current found solution can be **pruned** (discarded)

1 Tree search optimization

2 Implementing branch and bound

3 A branch and bound for the TSP

4 Concluding remarks

Savings-based constructive heuristic

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa

Saving of edge ij from k
Distance saved replacing $k-i-k-j-k$ by $k-i-j-k$

$$s_{ij}^{k} = (d_{ik} + d_{kj}) - d_{ij}$$

$$\begin{tikzcd}
i \arrow[rr, "d_{ij}"] \arrow[dr, "d_{ik}"'] & & j \\
& k \arrow[ur, "d_{kj}"'] &
\end{tikzcd}$$

$$\text{MIN} \sum_{\text{cycle}} d_{ij} \Leftrightarrow \text{MAX} \sum_{\text{cycle}} s_{ij}^{k}$$

A solution is defined with $n-2$ edges not passing through k
Heuristic: picking the $n-2$ compatible edges with maximal total savings

A small TSP instance

Matrix distance

	A	B	C	D	E
A	-	3	4	2	7
B	3	-	4	6	3
C	4	4	-	5	8
D	2	6	5	-	6
E	7	3	8	6	-

A TSP instance of size $n = 5$

A small TSP instance

List of edge savings from B

Edge	s_{ij}^B
AD	7
CD	5
AC	3
DE	3
AE	-1
CE	-1

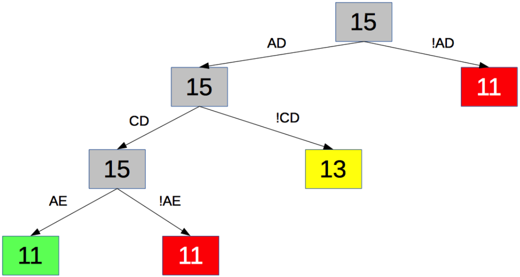
A branch and bound for the TSP

Defining the branch and bound

- **Branching** in two subsets: picking and not picking the compatible edge of maximum saving not considered in subset definition
- **Bounding:** completing the edges of subset definition with (possibly non compatible) edges of maximum savings value up to $n - 2$

A branch and bound for the TSP

First terminal node

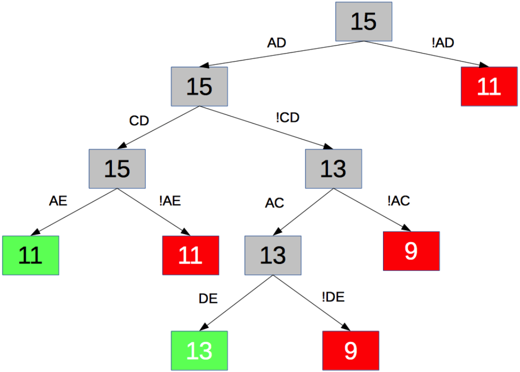


First candidate solution found: edges AD, CD, AE.

We can keep exploring yellow node (bound 13)

A branch and bound for the TSP

End of branch and bound



All possible nodes are pruned (no bound better is than the best solution).

A branch and bound for the TSP

Optimal solution

Edge	s_{ij}^B
AD	7
CD	5
AC	3
DE	3
AE	-1
CE	-1

- The optimal solution is ACBED
- Total savings: 13 (maximum), total distance: 19 (minimum)

1 Tree search optimization

2 Implementing branch and bound

3 A branch and bound for the TSP

4 Concluding remarks

Concluding remarks

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa

- The aim of tree search is to find the optimum through exploration of possible solutions
- Exploration is carried out through a tree search, **branching** subsets of candidate solutions
- Subsets not worth exploring can be detected through bounding the objective function in each subset
- Tree search (branch and bound, branch and cut) are costly algorithms, looking for high-quality (often optimal) solutions