

B&B

## Structure

1: listOfChildren (getChildren) (node: Node)

→ Set

2: cost (c)  $\Delta$  Pas de Node en arg!

(Voir être appelée constructeur)

$\Rightarrow$  (val, path, cost, parent) (signature)

3: addToLiveNodes (maintain PQ Structure)

i.e. sorted

4: nextENode : just pop

5: P (defines if a Node is a solution)

## 2.1 fonction de coût optimale?

Pour qu'elle soit optimale  $\Rightarrow$  deux ~~autres~~ prémisses doivent être vues.

i.e. pour une liste des live nodes  $L$  et  $x^*$  une sol.

on doit avoir  $\hat{C}(x^*) = C(x^*)$  et  $\hat{C}(x) \leq C(x)$

Comme ça on obtient que si on maintient une priority queue q. l'E-node choisit à toujours le plus petit coût de  $L$  i.e.

\* Si  $x^*$  est choisi alors

$$\hat{C}(x^*) = C(x^*) \leq \hat{C}(l) \leq C(l)$$

$x^*$  est bien de coût min. i.e est la solution optimale, (par déf de  $C$ ,  $C(\text{root}) = C(\text{optimal})$ )

$$C(\text{root}) = \sum_{j=0}^M \min_{x \in I_j} (C(x) + \hat{C}(x))$$

que la somme des coût min est bien le min des coût (par déf de min).

ici  $C$  est donné explicitement, on peut juste prendre  $\hat{C} = C$ .  $\rightarrow C$  prend pas en compte  $\hat{C}$  le fait que  $\hat{C} \leq C$