Branch and bound - Approximations
Advanced Algorithms

Master CPS2/DSC/MLDM

Amaury Habrard

amaury.habrard@univ-st-etienne.fr

Laboratoire Hubert Curien, UMR CNRS 5516
Université Jean Monnet Saint-Étienne
amaury.habrard@univ-st-etienne.fr

Semester 1

Question 1 **Task Assignment**

Q1 cost and search tree

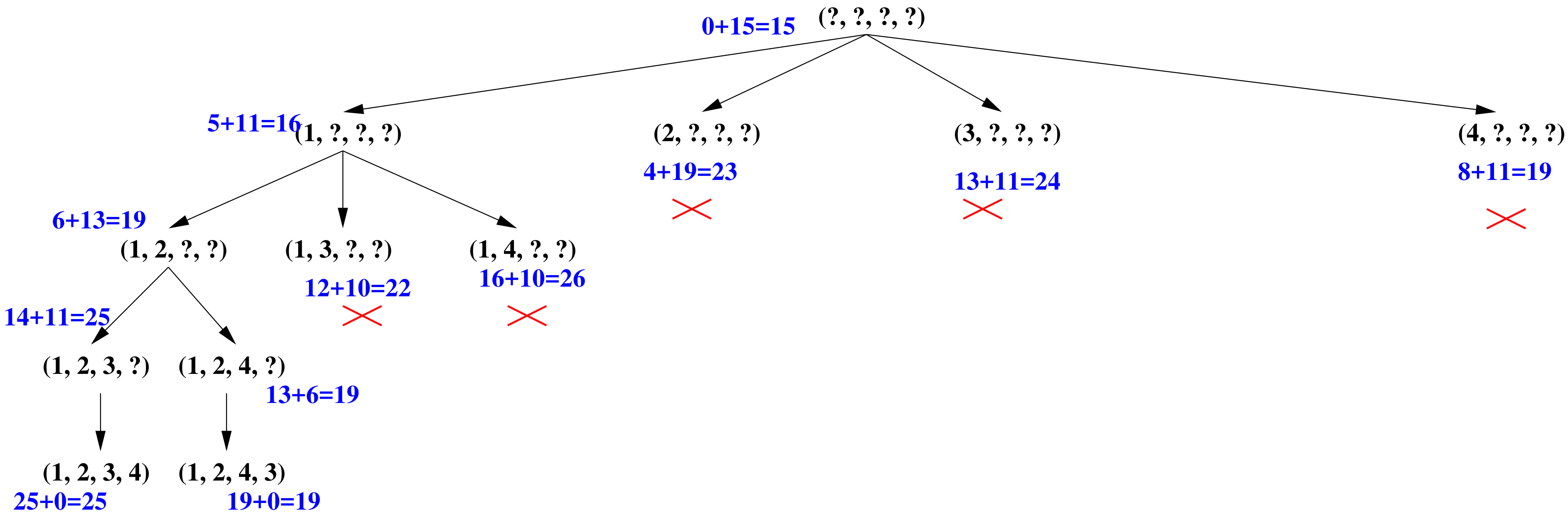
Similarly as in the lecture, imagine that we have currently affected a task for k agents out of n, for a corresponding solution vector $\mathbf{v}$, we can define the following quantities:

- $g^*(\mathbf{v}) = \sum_{i=1}^{k} c[i, \mathbf{v}[i]]$
  the sum of the costs of the tasks affected to the first k agents
-
  $$f(\mathbf{v}) = \sum_{i=k+1}^{n} \min_{\substack{1 \leq j \leq n \\ j \neq \mathbf{v}[l] \text{ for } 1 \leq l \leq k}} c[i, j]$$

  we take the minimum for each agent (line) among the available quantities

Q1: Search Tree

Here is the new search tree obtained:



Similarly, each node defines a (partial) solution written in black. We associate the quantity $g^* + h = f$ in blue next (or below) to each node (first value before $=$ is g^*, second is h). The red crosses indicate when the search can be cut. You can see that more branches can cut when a finer evaluation function is used.

Q2: example

Consider the following cost table with 2 agents and 2 functions:

Agent\Task	1	2
1	2	1
2	8	2

The strategy implies to assign task 2 to agent 1, then we need to associate task 1 to agent 2. The final cost is 8+1=9.
However, the optimal solution consists in assigning task 1 to agent 1 and task 2 to agent 2, leading to a final cost of 2+2=4.
This strategy is clearly not optimal. In particular, with this example you can see that $h = 8$ because the other values are removed leading to an evaluation function that can be higher than the optimal result.
$\Rightarrow$ **your evaluation function must always provide a smaller result than the optimal solution.**