CentraleSupélec

Algorithmics and complexity

TD 7/7 – Solving NP-hard problems

Training exercises

Note: The solution elements given here are not complete. Their purpose is only to guide you. We encourage you to write a proper answer, as you would do for the exam. If any question remains, feel free to ask your tutorial supervisor for help.

Exercice 1 : Greedy

Question 1

Propose a greedy algorithm for the change making problem.

Solution elements :

Select the coin with the highest value but less than the remaining sum. Update the remaining sum and repeat the process until reaching 0.

Question 2

Does this algorithm return the optimal solution? Prove it.

Solution elements :

Not for the general case. A counter example: the set of coins $1, 3, 4$ and the amount 6 The greedy algorithm uses 3 coins while the optimal solution only uses 2.

There exists sets of coins for which this greedy algorithm is optimal. Such sets are called canonical.

Exercice 2 : Approximation

Let's consider the vertex cover problem:

VERTEX COVER
Inputs :

- $G = (V, E)$ a non-oriented graph with $E = V \times V$
- $k \in \mathbb{N}$

Question : Is there a subset $S \subseteq V$ of size k such that each edge $(u, v) \in E$ is connected to at least on vertex of S (i.e. $u \in S \vee v \in S$)?

The corresponding optimisation problem (minimising the size of S) has practical applications life the installation of a video surveillance system for the subway which cover every allay with a minimal budget.

The vertex cover problem is NP-hard.

Let's consider the greedy algorithm that repeat those steps while there are still uncovered edges: Take one edge randomly from all the uncovered edges. Add **both extremities** to the current solution. Update the remaining uncovered edges.

Formally:

Function `greedy_edges(`*graph* $G = (V, E)$`)`
  $S \leftarrow \emptyset$;
  uncovered_edges $\leftarrow E$;
  while *uncovered_edges* $\neq \emptyset$ **do**
    $e = \{u, v\} \leftarrow$ a random edge in uncovered_edges;
    $S \leftarrow S \cup \{u, v\}$;
    v_cover $\leftarrow \{e : e \in E \wedge e = \{v, w\}\}$;
    u_cover $\leftarrow \{e : e \in E \wedge e = \{u, w\}\}$;
    uncovered_edges $\leftarrow$ uncovered_edges $-$ v_cover $-$ u_cover
  end
  return S

Question 1

Prove that this algorithm has an approximation ratio of 2.

Solution elements :

An approximation ratio of 2 means that the ratio between the solution returned by the algorithm and the optimal solution can never be more than 2.

Let S be a solution returned by the greedy algorithm and S^* an optimal solution for *Minimum Vertex Cover*.

Let A be the set of edges selected by the greedy algorithm.

$|S| = 2|A|$

$\bigwedge_{a,b \in A} a \cap b = \emptyset$

Therefore $|A| \leq |S^*|$

We deduce that $|A| \leq |S^*| \leq |S| = 2|A| \leq 2|S^*|$

Finally $1 \leq \frac{|S|}{|S^*|} \leq 2$

The greedy algorithm is a **2-approximation**.