

## Greedy 2-approximation for MAX-CUT on weighted graphs

Prove that the greedy algorithm for MAX-CUT described in class gives also a 2-approximation for weighted graphs with positive weights.

### Solution

First of all we redefine the problem in the context of weighted graphs. A weighted graph  $G$  is defined as the pair of  $V$  the set of nodes and  $w$  the function which assign weights to the edges. Since we have no directed edges the domain of  $w$  is made of sets of two nodes (not of ordered pairs). We don't store information about the presence/absence of edges directly, an edge which is not in the graph simply has weight 0.

$$G(V, w)$$

$$w : \{V, V\} \longrightarrow \mathbb{R}_+$$

Let  $w_{tot}$  be total weight of the graph.

$$w_{tot} = \sum_{u, v \in V} w(\{u, v\})$$

Now given two sets  $S$  and  $\bar{S} = V - S$ , the  $cut(S, \bar{S})$  has weight

$$cut\_weight(S, \bar{S}) = \sum_{u \in S, v \in \bar{S}} w(\{u, v\})$$

We want to find the cut with maximal weight.

We will use the same algorithm seen in class<sup>1</sup>.

In the algorithm we consider the nodes in some order, we use this ordering to name the nodes  $v_1, v_2 \dots v_n$  where the algorithm decide the position of  $v_i$  before the one of  $v_{i+1}$ . For each edge we define a responsible node which is the last one (in order of visiting) of the pair of nodes connected by the edge. The responsible node of an edge will decide whether it is better to take the edge in the cut or to discard it, and because of the algorithm policy it will make a greedy choice.

Let  $r_i$  be the total weight  $v_i$  is responsible for, i.e. the sum of the weights of all the nodes  $v_i$  is responsible for. Since we have named the nodes in the same order they will be considered by the algorithm it holds:

$$r_i = \sum_{j < i} w(\{v_j, v_i\})$$

Since each edge has exactly one responsible node it holds also that

$$w_{tot} = \sum_{v_1, v_2 \in V} w(\{v_1, v_2\})$$

$$= \sum_{v_i \in V} r_i$$

Note that the total weight added by the algorithm when considering each responsible node  $v_i$  must be greater or equal than  $r_i/2$  (otherwise the algorithm would choose the put the node in the other set and to take the complementary set of edges). The algorithm will process all the  $n$  nodes in order, let  $w_{res}$  be the returned estimation of  $w_{opt}$ .

$$w_{res} \geq \sum_{v_i \in V} \frac{r_i}{2} = \frac{1}{2} \sum_{v_i \in V} r_i = \frac{1}{2} w_{tot}$$

$$w_{res} \geq \frac{1}{2} w_{tot} \geq \frac{1}{2} w_{opt}$$

where  $w_{opt}$  is the weight of the optimal cut<sup>2</sup>.

<sup>1</sup>The condition that it evaluates in order to choose in which set to put a node is the same, but now the greedy optimal choice depends on the total weight of the cut and not on the number of edges.

<sup>2</sup>Since the set of edges in each cut is a subset of all the edges in  $G$  it holds that  $w_{opt} \leq w_{tot}$ .