ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

HS 2021

Prof. R. Wattenhofer
Peter Belcak

Computational Thinking
Sample Solution to Exercise 4

1 Bicolored Edges

a) Consider the example graph and initial coloring shown in Figure 1a. One can observe that none of the nodes in this graph is wasteful, so the algorithm does not even enter the loop once, and returns this initial coloring with 4 bicolored edges.

On the other hand, the optimal solution on Figure 1b is a coloring with 5 bicolored edges.

(a) (b)

Figure 1: Example where the greedy algorithms is worse than the optimum.

b) Whenever the algorithm switches the color of a wasteful node (i.e. in each iteration of the main loop), the number of bicolored edges in the graph strictly increases. Since the number of bicolored edges is ≥ 0 in the beginning, and cannot ever increase over $|E| = O(n^2)$, the algorithm executes at most $O(n^2)$ such steps.

c) Let $m = |E|$. Note that we have $v^* \leq m$ for the optimum v^*, since there are only m edges in the graph. Thus to prove that the algorithm is a 2-approximation, it is enough to show that it always returns a coloring with at least $\frac{m}{2}$ bicolored edges.

Consider the final coloring returned by the algorithm. Since the main loop of the algorithm has terminated, we know that under this coloring, G does not have any wasteful nodes anymore. Let deg_v denote the degree of v, and bic_v denote the number of bicolored edges where v is one of the endpoints. As v is not wasteful, we have $\mathrm{bic}_v \geq \frac{1}{2} \cdot \mathrm{deg}_v$ for each node v. Summing this up for all nodes, we get

$$\sum_{v \in V} \mathrm{bic}_v \geq \sum_{v \in V} \frac{1}{2} \cdot \mathrm{deg}_v = \frac{1}{2} \cdot \sum_{v \in V} \mathrm{deg}_v = m\,.$$

The sum on the left-hand side of this equation counts each bicolored edge twice (at both of its endpoints), so the number of bicolored edges in the graph is

$$\frac{1}{2} \cdot \sum_{v \in V} \mathrm{bic}_v \geq \frac{m}{2}\,.$$

2 Finding 4-segments

Let S denote the set of disjoint 4-segments returned by the algorithm, and S^* the optimal set of disjoint 4-segments. To show that our algorithm is a 4-approximation, we need to prove that $|S^*| \leq 4 \cdot |S|$ for any possible input graph.

Assume for a contradiction that the opposite is true, and we have $|S^*| \geq 4 \cdot |S| + 1$ for some graph. On the one hand, the 4-segments in S altogether consists of exactly $4 \cdot |S|$ edges. On the other hand, in S^* we have at least $4 \cdot |S| + 1$ distinct 4-segments; since these are all disjoint, all the $4 \cdot |S|$ edges of S can only appear in one of these 4-segments. Thus there has to be at least one 4-segment $s \in S^*$ that does not contain any of the edges of the 4-segments in S. However, this means that this s is still free with respect to S, so our greedy algorithm should not have terminated without adding s to S.

This gives us a contradiction, so we must indeed have $|S^*| \leq 4 \cdot |S|$.