CS-E4510 Distributed Algorithms / Juho Hirvonen, Jukka Suomela
exam, 26 February 2020

Instructions. There is only one question, which is an open-ended small research project. Your answer has to demonstrate ***at least*** that you understand both the PN and LOCAL models of distributed computing, you can design efficient distributed algorithms, and you can also prove negative results; you do not need to do more than what is reasonable to expect to finish in ***3 hours***.

You are free to look at any source material (this includes lecture notes, textbooks, and anything you can find with Google), but you are not allowed to collaborate with anyone else or ask for anyone's help (this includes collaboration with other students and asking for help in online forums). You are free to use any results from the lecture notes directly (including the results from the regular exercises). You are free to use computers and computer programs to find solutions.

* * *

Definition. Let $G = (V, E)$ be a graph, and let a and b be real numbers. We say that a function $f : V \to [0, 1]$ is an (a, b)-labeling if the following holds: for all edges $\{u, v\} \in E$ we have got $a \le \left|f(u) - f(v)\right| \le b$.

* * *

Question. Your task is to study distributed algorithms for finding (a, b)-labelings, for different values of parameters a and b. You can first consider the case that graph G is a ***cycle***. What can you say about the problem in that case?

- For what values of a and b a solution always exists in any cycle?
- For what values the problem cannot be solved at all in the deterministic PN model?
- For what values the problem can be solved in $O(1)$ rounds in the deterministic LOCAL model?
- For what values the problem can be solved in $o(n)$ rounds in the deterministic LOCAL model? [Note, it is small-o, not big-O.]
- For what values the problem requires $\Omega(n)$ rounds in the deterministic LOCAL model?

You do not need to give a full characterization that covers all possible (a, b) pairs, but you should be able to say at least something positive and something negative. If you have time or you run out of questions you can solve, you can also consider similar questions in other graph families, e.g. ***paths*** or ***trees***, and/or other models of computing, e.g., randomized PN or deterministic CONGEST.

* * *

Hints. To get started, you can consider e.g. the following questions:

- What does a $(0, 0)$-labeling look like in a cycle? Does it always exist, is it easy to find?
- What does a $(0, 1)$-labeling look like in a cycle? Does it always exist, is it easy to find?
- What does a $(1, 1)$-labeling look like in a cycle? Does it always exist, is it easy to find?
- What does a $(0.5, 1)$-labeling look like in a cycle? Does it always exist, is it easy to find?
- What does a $(0.1, 0.1)$-labeling look like in a cycle? Does it always exist, is it easy to find?
- What does a $(0.1, 0.2)$-labeling look like in a cycle? Does it always exist, is it easy to find?