School of Computer Science and Technology
University of Science and Technology of China

Exercise Sheet 6 for Design and Analysis of Algorithms Autumn 2022

Solution

Exercise 1 (40 points, graded by Yudong Zhang)
Consider the synchronous network model for distributed algorithms. Let G be a graph with n nodes and m edges, and diameter D. Let s be a distinguished source (or root) node in G.

(a) Assume that the graph is unweighted, i.e., every edge has length 1. Give an algorithm such that after $O(D)$ rounds every node knows its distance to s. **(20 points)**

(b) Assume that the graph is weighted with edge length in the range $1, 2, \cdots, k$. Give an algorithm that after $O(kD)$ rounds every node knows its distance to s. We do not assume that the edge lengths influence the time needed for sending messages. In each round, every node can send to each neighbor one message, regardless of the edge lengths. **(20 points)**

You do not have to prove the correctness of your algorithms explicitly, but you should argue about their time complexity and message complexity.

Solution.

(a) Modify the algorithm for the BFS problem from the class, such that every node learns and outputs its distance to s. Specifically, each node records its distance to s in a new *dist* variable, initially 0 for s, and ∞ for the others. Include the *dist* value d in every *search* message; when an unmarked node receives a $search(d)$ message, it sets its own *dist* to $d + 1$.

The algorithm for node i is given as follows:

Round 1:

- If $i = s$, node i sends a $search(0)$ message to its neighbors
- If node i receives a $search(0)$ message, it:
  - marks itself
  - sets its *dist* value to 1
  - plans to send at the next round

Round $r > 1$:

- If node i planned to send, it sends a $search(d)$ message to its neighbors where d is the value of its own *dist* variable
- If node i is not marked and receives a $search(d)$ message, it:
  - marks itself
  - sets its *dist* value to $d + 1$
  - plans to send at the next round

Time complexity: The number of rounds until all nodes output their distance to s is the maximum distance of any node to s, which is $O(D)$.

Message complexity: For each edge, at most 2 messages are passed through during the entire execution. Therefore, the number of messages sent by all nodes is at most $2m$, which is $O(m)$.

(b) Each node i keeps track of *dist*, the shortest distance to s it knows so far, initially 0 for node s and ∞ for the others. The algorithm for node i is given as follows, which is very similar to the Bellman-Ford shortest paths algorithm:

At each round:

(1) Send a distance (*dist*) message to all neighbors.

(2) Receive messages from neighbors; let d_j be the distance received from neighbor j.

(3) Perform a relaxation step:

$$dist := \min\{dist, \min_j(d_j + w_{ij})\}$$

Time complexity: For every r, if a node i updates its distance to s at the r-th round, the new path from s to i consists of exactly r edges, which also implies that the new distance to s is at least r. Note that the maximum distance of any node to s is kD. After kD rounds, if any node updates its distance to s, the distance would be greater than kD, which leads to a contradiction. Therefore, after $O(kD)$ rounds, all the distances stabilize to their final values.

Message complexity: At each round, each node sends a distance message to all neighbors, so for each edge, at most 2 messages are passed through. Therefore, the number of messages sent by all nodes *at each round* is $O(m)$. Considering the time complexity is $O(kD)$, the message complexity during the entire execution is $O(kDm)$.

Exercise 2 **(60 points, graded by Yinhao Dong)**
Read Chapter 1.1 and Chapter 2.1 in the book `https://people.csail.mit.edu/ghaffari/DA22/Notes/DGA.pdf` for the definitions of LOCAL model and CONGEST model.
Consider the synchronous distributed algorithms for Breadth-First Search Tree and the (Bellman-Ford) shortest paths from the class. Which algorithm(s) work in the LOCAL model and which work in the CONGEST model?

Solution. According to the definitions, the LOCAL model does not assume any limitation on the size of the messages. However, in the CONGEST model, we take the bandwidth limitations into account. Specifically, per round, each node/process sends one B-bit message to each of its neighbors. Usually, we assume that $B = O(\log n)$, which, e.g., implies that each message can describe constant many edges or vertices of the network.
Therefore, the algorithm for Breadth-First Search Tree works in both models because each process sends a 1-bit message (*search*) per round. However, the algorithm for the (Bellman-Ford) shortest paths only works in the LOCAL model because, at each round, each process sends its distance from the root to all neighbors, while the distance can be exponentially large.
