University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
Z. Parsaeian, G. Schmid

Distributed Graph Algorithms
Sample Solution Exercise Sheet 7

1 Edge Coloring

Use network decomposition to compute a $(2\Delta - 1)$-edge coloring in $O(\text{poly}(\log n))$ rounds, without using line graphs.
Hint: What extra property should your network decomposition hold so that iterating through the colors of the network decomposition does not creates any conflicts?

Sample Solution

The process begins by calculating an $(O(\log n), O(\log^3 n))$ network decomposition on the squared graph G^2. This is feasible in $O(\text{poly}(\log n))$ rounds because simulating an algorithm designed for G^2 on the base graph G only results in a constant-factor increase in communication rounds.
A key advantage of performing the decomposition on G^2 is that any two clusters belonging to the same color class are guaranteed to have a minimum distance of 3 in the original graph G. This separation allows us to process the clusters of each color sequentially:

- For each color class in the decomposition, every cluster gathers its own topology and that of its adjacent nodes and edges.
- Each cluster then determines colors for any incident edges that remain uncolored.

Correctness and Complexity Because we have a budget of $2\Delta - 1$ colors, clusters can simply use a local greedy approach to color their edges. The distance-3 gap between clusters of the same color is crucial; it ensures that edges colored simultaneously by different clusters are never adjacent to one another. This confirms that the final $(2\Delta - 1)$-edge coloring is proper.
The total runtime is dominated by the initial network decomposition and the subsequent gathering and coloring steps. Since the decomposition takes $O(\text{poly}(\log n))$ rounds and the coloring phase takes approximately $O(\log^4 n)$ rounds, the overall complexity remains $O(\text{poly}(\log n))$ rounds.

2 Algorithm

In the lecture, we established that network decomposition is a powerful tool for solving various distributed problems. We specifically proved the existence of an $(O(\log n), O(\log n))$ strong-diameter network decomposition.
Let us now focus on the construction. Show how to compute such a decomposition **deterministically** in $O(\text{poly}(\log n))$ rounds.

Sample Solution

The question is taken from the 2021 paper "Polylogarithmic-time deterministic network decomposition and distributed derandomization" by Rozhoň and Ghaffari.

We define an auxiliary graph $G' := G^{10 \log n}$. In this graph, an edge exists between any two vertices $u, v \in V$ if their distance in G is at most $10 \log n$. We then apply the $(O(\log n), O(\log^3 n))$ weak-diameter network decomposition on graph G' in $O(\log^7 n)$ rounds. This simulation on G requires $O(\log^8 n)$ rounds. The resulting decomposition consists of $q = O(\log n)$ colors, where each cluster has a weak diameter of $O(\log^3 n)$ in G', which translates to a weak diameter of $O(\log^4 n)$ in the original graph G.

Parallelized Ball Carving We determine the first color of our final output decomposition by iterating through the q colors of the decomposition in sequence. For a fixed color, we process its clusters in parallel as follows:

- **Local Topology Gathering**: Each cluster elects a leader to collect the topology of all currently active nodes within a distance of $\log n$ of vertices of the cluster. This step is completed in $O(\log^4 n)$ rounds because the clusters have a limited weak diameter.
- **Disjoint Processing**: Since clusters of the same color are separated by a distance of at least $10 \log n$, the local neighborhoods gathered by different clusters are vertex-disjoint.
- **Ball Carving Procedure**: Each cluster C independently executes a sequential ball-growing process on its collected topology. Starting from an unclustered node v, we expand a ball hop-by-hop. **Growth Condition**: We increment the ball radius as long as the number of nodes on the outside boundary is greater than or equal to the number of nodes already inside the ball.
- **Clustering and Boundary Removal**: Once this growth condition fails, the ball is designated as an output cluster. All nodes immediately adjacent to this ball are labeled "dead" and removed from the current phase to provide separation. (Then, if any node v' of cluster C remains unclustered (for the output decomposition), we start a similar ball growing process from v', but only on the graph induced by the remaining nodes. We continue similarly until all nodes of cluster C are clustered for the output decomposition.)

Analysis and Convergence Because the number of nodes in a ball at least doubles during every successful expansion step, the carving process must stop before the radius exceeds $\log n$. This guarantees that all output clusters possess a strong diameter of $O(\log n)$. Furthermore, the distance gap provided by the decomposition ensures that the ball-growing processes of different clusters never interfere with one another.
At the end of each stage, the ratio of clustered nodes to "dead" nodes is favorable: at least half of the participating nodes are successfully clustered. After all q colors are processed, we restore the dead nodes and repeat the entire construction on the remaining subgraph to find the next output color. Since the number of remaining nodes decreases by a factor of 2 in each repetition, the algorithm terminates in $O(\log n)$ main phases.

3 Constant Degree

Show that, on constant-degree graphs, an $(O(\log n), O(\log n))$ strong-diameter network decomposition can be computed in $O(\log^* n)$ rounds.
Hint: you can compute even an $(O(1), O(1))$ strong-diameter network decomposition in $O(\log^\star n)$ rounds.

Sample Solution

On graphs with constant degree, i.e., $\Delta = O(1)$, a $(\Delta + 1)$-coloring is indeed an $(O(1), O(1))$ strong-diameter network decomposition. As a result, we can compute a $(\Delta + 1)$-coloring in $O(\Delta + \log^\star n)$ rounds, which having $\Delta = O(1)$ it results in a runtime of $O(\log^\star n)$ rounds.