Project of Algorithms on Node Labeling

Kevin Lei

August 2, 2024

1 Introduction

In this project we discuss the "Node Labeling Problem", in which we attempt to label the nodes of a graph with unique labels from a set of labels. We have the following definitions:

- Let $G = (V, E)$ be an undirected graph.
- Let $d(u, v)$ be the distance between nodes u and v.
- For all nodes $v \in V$, let $N(v, h) \subseteq V$ be the set of nodes that are at most h hops away from v.
- Let $K = \{0, 1, \ldots, k-1\}$ be the set of k integers, where $k \leq |V|$.
- For all $v \in V$, let $c(v) \in K$ be the label of node v, where different nodes may have the same label.
- Let $C(v, h)$ be the set of labels of nodes in $N(v, h)$.
- A labelling of the nodes is *valid* if every label in K is used at least once.
- Let $r(v)$ be the smallest integer such that the node v has all the labels in K in $N(v, r(v))$.
- Let $m(v)$ be the smallest integer such that the node v has at least k nodes in $N(v, m(v))$.

Formally, our relevant sets and values can be defined as follows:

$$\begin{aligned}
N(v, h) &\triangleq \{u \in V \mid d(u, v) \leq h\} \\
C(v, h) &\triangleq \{c(u) \mid u \in N(v, h)\} \\
r(v) &\triangleq \min\{h \mid |C(v, h)| = k\} \\
m(v) &\triangleq \min\{h \mid |N(v, h)| \geq k\}.
\end{aligned}$$

Note that in general, we have $|C(v, h)| \leq |N(v, h)|$, since the labels of nodes in $N(v, h)$ are not necessarily distinct, and $r(v) \geq m(v)$, since there must be at least one label per node.

The Node-Labelling Decision Problem is defined as follows:

Given:

- An undirected graph $G = (V, E)$
- A set of $k \leq |V|$ labels $K = \{0, 1, \ldots, k-1\}$
- A nonnegative integer R,

does there exist a labeling $c(v)$ for all $v \in V$ such that $|C(v, R)| = k$ for all $v \in V$?

Now consider this as an optimization problem. The Node-Labelling Optimization Problem is defined as follows:

Given:

- An undirected graph $G = (V, E)$
- A set of $k \leq |V|$ labels $K = \{0, 1, \ldots, k-1\}$,

find a valid labeling for all the nodes such that $\max_{v \in V} \frac{r(v)}{m(v)}$ is minimized.

In the case of the optimization problem, if an algorithm that solves it has $\max_{v \in V} \frac{r(v)}{m(v)} \leq \rho$ for all possible instances, then we say that the algorithm has a *proximity ratio* of ρ, and the algorithm is a ρ-proximity algorithm.

First, we will prove that the Node-Labelling Decision Problem is NP-Complete. Then, we will present a polynomial-time algorithm for the Node-Labelling Optimization Problem where the graph is a tree, analyze the proximity ratio of the algorithm, and finally analyze the runtime complexity of the algorithm.

2 NP-Completeness Proof

Theorem 1. *The Node-Labelling Decision Problem is NP-Complete.*

Proof. A problem is NP-Complete if it is in NP and every problem in NP can be reduced to it in polynomial time. We will show the former by presenting a polynomial time algorithm to verify a solution to the Node-Labelling Decision Problem, and the latter by reducing ... to the Node-Labelling Decision Problem. □

3 Main Idea of the Algorithm

4 Pseudocode

Algorithm 1:

Input:

Output:

5 Proximity Ratio Analysis

6 Runtime Complexity Analysis