Implementing and Evaluating Space Efficient Algorithms for Detecting Large Neighbourhoods in Graph Streams

Student: Dominic Hutchinson, Supervisor: Dr. Christian Konrad, Project Type: Research

University of Bristol, Department of Computer Science

Graph Streams & The `Neighbourhood Detection Problem`

Graph Streams are a sequence of instructions which describe how to construct a graph. Each instruction provides details about an edge in the graph. Graph Streams come in two forms

- *Insertion-Only Streams* only ever insert new edges into the graph with each instruction.
- *Insertion-Deletion Streams* either insert a new edge, or remove an existing edge, from the graph with each instruction.

In the `Neighbourhood Detection Problem`(n, d) we are given a graph with n vertices & at least one vertex of degree d. We are tasked to return a vertex with at least $\frac{d}{c}$ of its neighbours, from some approximation factor $c \geq 1$.
There are some good motivativing applications of the `Neighbourhood Detection Problem`.

- Given a network of social media accounts, detect popular influencers and analyse the demographics they attract in order to plan targetted advertising campaigns.
- Given a log of traffic to a network detect whether a DDOS attack has occurred and, if so, by whom.
- Given a list of sales made on a website detect which items are most popular and which items they are commonly bought with.

1. Insertion-Stream Algorithm

Below is a proposed algorithm for solving the `Neighbourhood Detection Problem` for *Insertion-Only Streams*. This algorithm requires $O(n \log n + n^{\frac{1}{c}} d \log^2 n)$ space in theory.

Algorithm 1: One-pass c-approximation Insertion-Only Streaming Algorithm for `Neighbourhood Detection`

require: Space s, degree bound d.
$s \leftarrow \lceil \ln(n) \cdot n^{\frac{1}{c}} \rceil$
for $i \in [0, c-1]$ **in parallel do**
  $(a_i, S_i) \leftarrow$ `Deg-Res-Sampling` $\left(\max\left\{1, i \cdot \frac{d}{c}\right\}, \frac{d}{c}, s\right)$
return *Uniform random neighbourhood* (a_i, S_i) *from successful runs*

`Deg-Res-Sampling`(d_1, d_2, s) uniformly samples s times from the set of nodes with degree at least d_1. For each of these sampled nodes a neighbourhood of size $\min\{d_2, d - d_1 + 1\}$ is stored.

2. Insertion-Deletion Algorithm

Below is a proposed algorithm for solving the `Neighbourhood Detection Problem` for *Insertion-Deletion Streams* by first sample a set of vertices. This algorithm requires $O(\frac{xd}{c} \log^k n)$ space in theory.

Algorithm 2: One-pass c-approximation Insertion-Deletion Streaming Algorithm for `Neighbourhood Detection`. (Vertex Sampling)

require: Space s, degree bound d.
Let $x = \max\left\{\frac{n}{c}, \sqrt{n}\right\}$
Sample a uniform random subset $A' \subseteq A$ of size $10\, x \ln n$ of vertices.
for $a \in A'$ **do**
  Run $10\frac{d}{c} \ln n$ l_0-samplers on the set of edges incident to a.
return *Any neighbourhood of size* $\frac{d}{c}$ *among the stored edges, if there is one.*

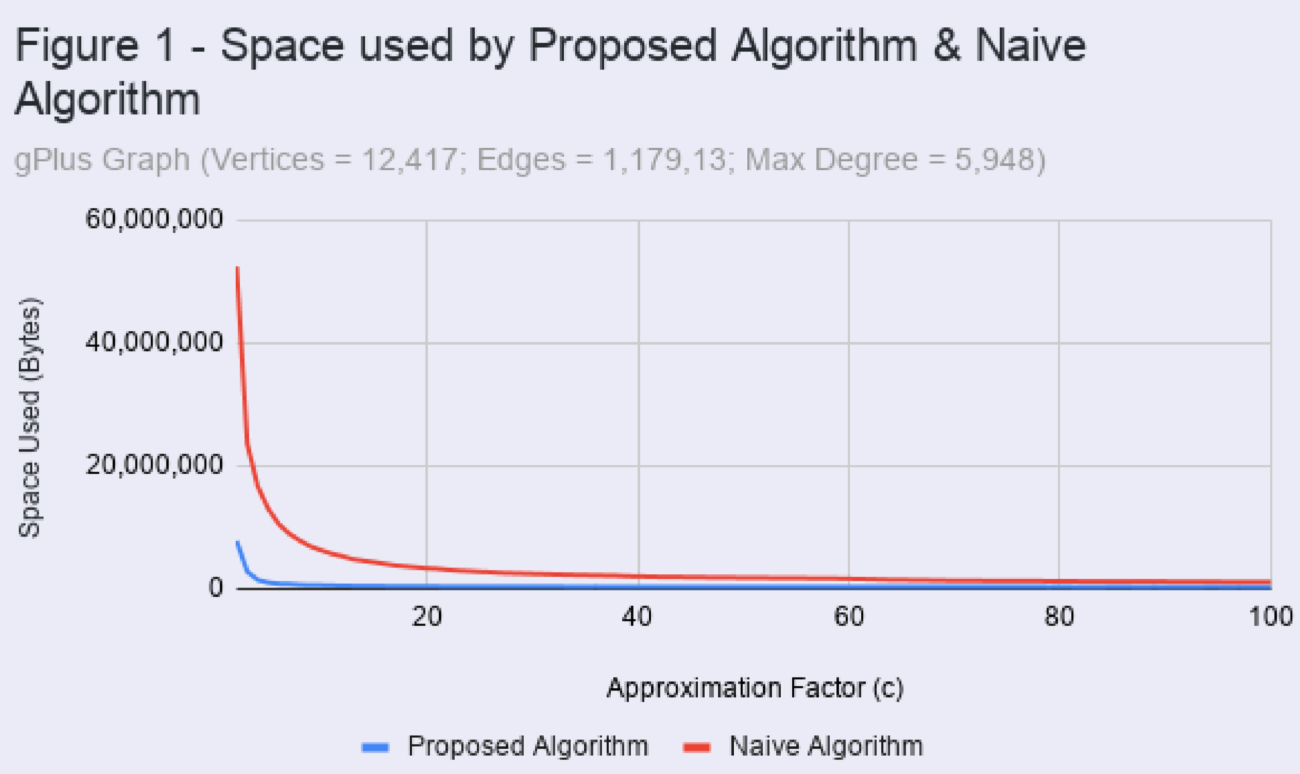
l_0 *Samplers* return an index which has been uniformly sampled from the indicies of non-zero elements of a vector.

3. Preliminary Results

I have implemented the *Insertion-Stream Algorithm* and a naïve algorithm for solving the problem in order to compare the results.

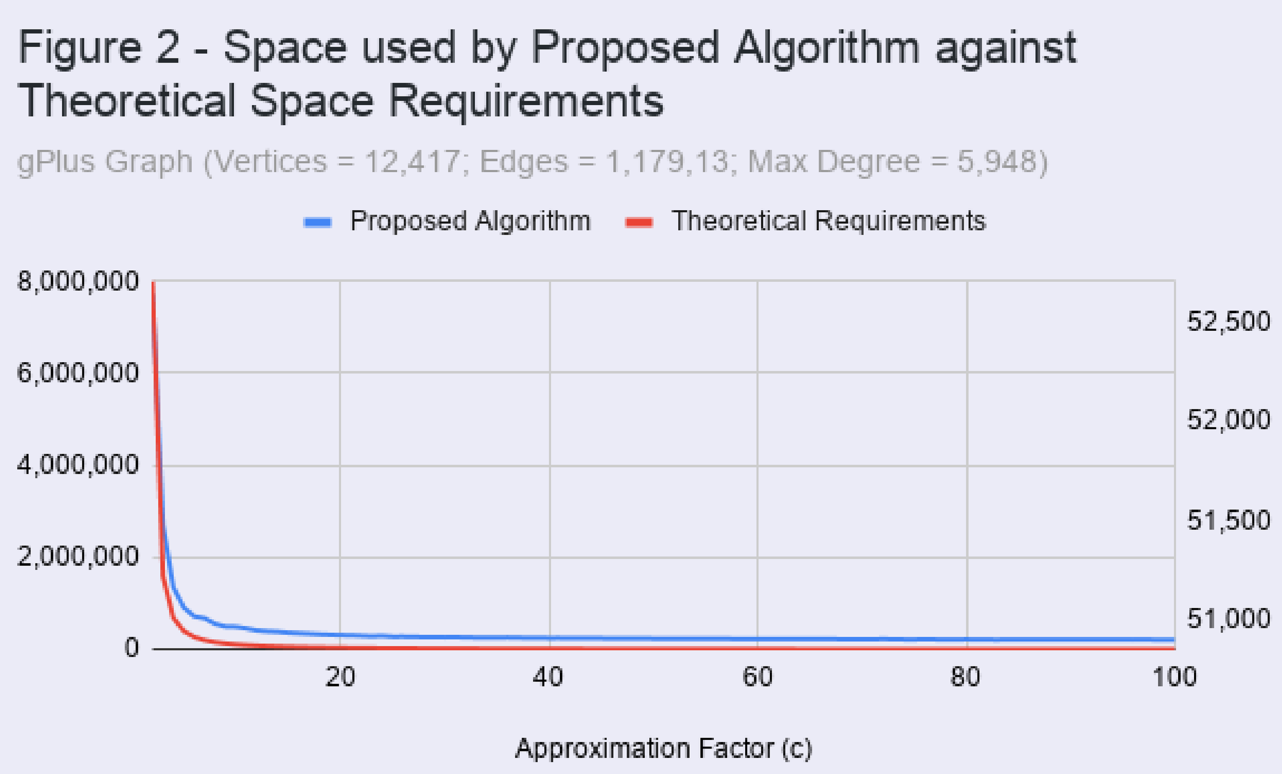
Fig1 Space requirements for each algorithm as the approximation factor is varied on the same graph.
Fig2 Space requirement for the proposed algorithm against it's theoretical space requirements.

Both figures show promising results for the proposed algorithm.



4. Further Aims

- Perform tests on different size graphs for the same approximation factor.
- Perform tests on different density graphs for the same approximation factor.
- Implement the proposed algorithm for *Insertion-Deletion Streams.*
- Investigate an application of these algorithms.



University of Bristol