Info Document: Classified Rank Maximal Matching Algorithm

Overview

The Classified Rank Maximal Matching Algorithm aims to compute the maximal matching from an input bipartite graph while considering the rank of the edges and classification of vertices.

Files

1. **GraphDefs.h**: Defines classes and typedefs for graph processing, including `VertexAlt`, `TreeNode`, `ClassificationListElement`, `Edge`, `Graph`, and typedefs for convenience.
2. **GraphReaderAlt.h**: Declares a class responsible for reading and parsing a graph from input data.
3. **NetworkBuilder.h**: Declares a class responsible for constructing the cumulative graph network used in the algorithm.
4. **Algorithm.h**: Declares a class responsible for executing the algorithm on the network to compute the rank maximal matching.

Files

1. **GraphReaderAlt.cc**: Implements functions of the `GraphReaderAlt` class for reading the Classifications, Orignal Edges and Preferences of the vertices.
2. **NetworkBuilder.cc**: Implements functions of the `NetworkBuilder` class for building the final graph network from the given Classification and Preference inputs.
3. **Algorithm.cc**: Implements functions of the `Algorithm` class for executing the flow algorithm.

Key Components

- **TreeNode**: Represents the information stored in the classification `TreeNode`
- **Graph**: Contains the following information

> `vector<VertexAlt> partA, partP` - Partitions of the graph.
> `vector<vector<Edge>> edges` - Edges in the input graph (Part A to Part P).
> `vector<vector<int>> network` - Adjacency Matrix of the final cumulative graph.

- **NetworkBuilder**: Constructs the cumulative graph network from the input bipartite graph.
- **Algorithm**: Executes the flow algorithm on the network to compute the maximal matching.

Process

1. **Read and Parse Graph**: Graph data is read and parsed to extract vertices, edges, and their classifications.
2. **Build Classification Tree**: Classification trees are constructed for each vertex based on their classifications.
3. **Construct Cumulative Graph Network**: The cumulative graph network is built by adding edges based on the rank and classification of vertices.
4. **Apply Flow Algorithm**: The flow algorithm is applied on the network to compute the classified rank maximal matching by augmenting the flow.

Input File Format

First Line contains the number of vertices n in part A

Then for the n lines, we give the `id` of the vertex and the `quota` for the vertex in the part A.

Next Line contains the number of vertices m in part P

Then for the m lines, we give the `id` of the vertex and the `quota` for the vertex in the part P.

Next Line contains the number of ranks k in the graph.

Then for each of the ranks 1 to k, we provide the

1. Number of edges l
2. Each of the next l lines represent the endpoints of the edge.

Input File Format

For each vertex we provide the number of classifications it proposes n_c

We provide the classifications for the vertices in the format
`ID` `Quota` `NumVertices`

We provide the Vertices by their IDs that are part of the class too

How to run

- We have assigned the flaq `-q` to our algorithm
- Save the input as mentioned in the input file format into `input.txt`
- If you wish to save your output to `output.txt` by running the algorithm run the following command
    - `./graphmatching -q < input.txt > output.txt`

Details

- The code outputs the first violating pair it encounters for non-laminar flow (if any) and terminates the execution.
- The output format prints the `Rank(i)` edges of the graph in the i^{th} iteration of the loop.

Future Work

- Make use of current Data Structures (`Vertex` and `BipartiteGraph`)
- Integrate it to seamlessly run on existing Algorithms
- Modify the input and output format to more user-friendly methods
- Modify the parser accordingly
- Throw every conflicting pair
- Check the duplicate classifications (quota = 0) single element
- Multiple definitions of same quota (Throw Error)
- Include lower_quota in the classification element class
- Document s and t as 0 and 1 labelled
- Given a input graph and then give maxflow and its S,T,U decomposition
- Name the functions better and make it more generic

Thank You