Directed Edge Collapser Artifact Evaluation

The code is presented as a Docker container. The whole set of experiments can be run using the following commands:

```
docker build -t dircollexps .
mkdir -p ./results
docker run -it --rm -v "$(pwd)/results:/workspace/experiments/results" dircollexps ./runme.sh
```

Use `runme_mini.sh` for the mini version.

Structure

```
.
├── Dockerfile                  # Docker specifications
├── runme.sh                    # Main orchestrator
├── 700_graph_exp.sh            # Experiment 1
├── 50_graph_exp.sh             # Experiment 1 (reduced to 50 vertices)
├── generate_flag.cpp           # Ordered complete graph generator
├── generate_flag_random.cpp    # Random directed graph generator
├── sparse_vs_mem.sh            # Experiment 2
├── benchmark_sparse.cpp        # Getting runtime results of sparse algorithm
├── benchmark_mem.cpp           # Getting runtime results of memory algorithm
├── er_edge_removal.sh          # Experiment 3
├── get_plot_homology.cpp       # Edge reduction percentages for homology computation
├── get_plot_pers.cpp           # Edge reduction percentages for persistence computation
├── test_flags/                 # Experiment 4 structure
│   ├── homology/
│   │   ├── hom_collapser.sh                # Runs algorithm on flag files in pre-collapse folder (homology)
│   │   ├── pre_collapse/
│   │   │   ├── *.flag                      # All data files (for homology)
│   │   │   └── run_flagser_hom_pre.sh      # Flagser homology computation on pre-collapse files
│   │   └── post_collapse/
│   │       ├── *.flag                      # Data files post collapse
│   │       └── run_flagser_hom_post.sh     # Flagser homology computation on post-collapse files
│   └── persistence/
│       ├── pers_collapser.sh               # Runs algorithm on flag files in pre-collapse folder (persistence)
│       ├── pre_collapse/
│       │   ├── *.flag                      # All data files (for persistence)
│       │   └── run_flagser_pers_pre.sh     # Flagser persistence computation on pre-collapse files
│       └── post_collapse/
│           ├── *.flag                      # Data files post collapse
│           └── run_flagser_pers_post.sh    # Flagser persistence computation on post-collapse files
├── test_flags_mini/            # Experiment 4 mini-version structure (same as full version)
└── results/                    # Output directory (will be created once the experiments are run)
```

Experiments

We provide all the experiments that are described in the paper.

1. 700-vertex (or 50-vertex) complete graph edge collapse analysis
2. Runtime comparison: sparse vs memory-optimized algorithms
3. ER graph edge removal percentage across densities
4. Flagser homology/persistence computation pre/post collapse (mini version included)

The results for all the experiments can be seen in the `./results/` directory.

Results

The `./results/` directory will contain files for each experiment as follows:

- Experiment 1 - `700_complete_graph_output.txt` will contain the edge reduction details for each iteration for ordered and random graphs. (`50_complete_graph_output.txt` for the 50-vertex version)
- Experiment 2 - `runtime_comparison.png` would be the plot of runtimes for the 2 algorithms. The runtime numbers can be found in `runtime_results_sparse.csv` and `runtime_results_mem.csv` .
- Experiment 3 - `edge_removal_er.png` would be the edge reduction percentage plot. `results_er_homology.csv` and `results_er_persistence.csv` would be the edge reduction numbers for homology and persistence computaion respectively.
- Experiment 4 - `*_flagser_results.csv` (4 such files) will contain the flagser runtimes of homology and persistence computation both pre and post collapse. `collapser_outputs/` contains txt files for each flag file in the pre-collapse collapse folders upon running the algorithm, details of edge reduction, memory used and iteration count would be present in these.