Readme for reproducibility submission of Paper314

Source code info

Repository: codes

Programming Language: C++

Additional Programming Language Info:

Compiler Info: gcc version 7.5.0 (Ubuntu 7.5.0-3ubuntu1~18.04)

Packages/Libraries Needed: Boost 1.66.0, CMake 2.8, makefile

Datasets info

All datasets used in our experiments are from the KONECT website. But, it seems the website is no longer available. Thus, we summarise all the datasets used into a zip file.

Repository: datasets

Hardware Info

Here you should include any details and comments about the used hardware in order to be able to accommodate the reproducibility effort. Any information about non-standard hardware should also be included. You should also include at least the following info:

- Processor: 2 Intel(R) Xeon(R) Silver 4110 CPU @ 2.10GHz processors
- Caches: 3 level caches (512KiB L1 cache, 8MiB L2 cache, 11MiB L3 cache) for each processor

- Memory: `256GiB System Memory` (8 * `32GiB DIMM DDR4 Synchronous 2666 MHz (0.4 ns)`)
- Secondary Storage: HDD, `6001GB TOSHIBA MG04ACA6`, (interface speed: `6.0 Gbit/s Max.`, rotation speed: `7,200 rpm` , average latency time: `4.17 ms` , buffer size: `128 MiB`, data transfer speed: `205 MiB/s`) write speed: `210-280 MiB/s`, read speed: `320-350 MiB/s`
- Network: there is no network usage in our experiments

Experimentation Info

Scripts and how-tos to generate all necessary data or locate datasets

After downloading the `datasets.tar.gz` file, use the following command to unzip the datasets files.

```
tar -xzf datasets.tar.gz
```

Scripts and how-tos to prepare the software for system

After cloning the codes from GitHub, use the following command to compile the codes in the repository:

```
cmake CMakeList.txt
```

```
makefile
```

Scripts and how-tos for all experiments executed for the paper

Assuming their is a parent folder containing the datasets folder and the code repository:

```
parent_folder/
|-- SIGMOD2020DDS/
|-- datasets/
```

To get the results in Figure 9 in the paper, use the following commands under `SIGMOD2020DDS` folder.

Run `Exact` on `MO`:

```
./DirectedDensestSubgraph -g ../datasets/MO.txt -a e -m b
```

Run `Core-Exact` on `MO`:

```
./DirectedDensestSubgraph -g ../datasets/MO.txt -a e -m c
```

Run `DC-Exact` on `MO`:

```
./DirectedDensestSubgraph -g ../datasets/MO.txt -a e -m a
```

Similarly, we can get the command for other datasets, i.e., `TC`, `OF`, `AD`, `AM`.

From the results for `DC-Exact` on each datasets, we can also get the `k` values in the Table 5, and statistics in Table 7.

To get the results in Figure 10, set the `size_reported` in the `Graph` class as `true` and re-compile the codes. Then, use the following commands.

Run `Exact` on `AD`:

```
./DirectedDensestSubgraph -g ../datasets/AD.txt -a e -m b
```

Run `Core-Exact` on `AD`:

```
./DirectedDensestSubgraph -g ../datasets/AD.txt -a e -m c
```

Run `DC-Exact` on `AD`:

```
./DirectedDensestSubgraph -g ../datasets/AD.txt -a e -m a
```

Similarly, we can get the command for `AM`.

Note, the trend plotted in Figure 10 is picked for a specific `a` value for each algorithm. The users can find similar trends on other `a` values.

To get the results in Figure 11 and Figure 12, use the following commands.

Run `Core-Approx` on `AD`:

```
./DirectedDensestSubgraph -g ../datasets/AD.txt -a a -m a
```

Run `PM-Approx` on `AD`:

```
./DirectedDensestSubgraph -g ../datasets/AD.txt -a a -m p -d 2 -e 1
```

Run `BS-Approx` on `AD`:

```
./DirectedDensestSubgraph -g ../datasets/AD.txt -a a -m b
```

Run `KS-Approx` on `AD`:

```
./DirectedDensestSubgraph -g ../datasets/AD.txt -a a -m i
```

Run `FKS-Approx` on `AD`:

```
./DirectedDensestSubgraph -g ../datasets/AD.txt -a a -m f
```

Similarly, we can get the command for `TC`, `OF`, `AD`, `AM`, `AR`, `BA`, `TW`.

The approximation ratio in Figure 12 equals the density returned by the exact algorithm (`DC-Exact`) divided by the density returned by the corresponding approximation algorithm.

Note, the `\delta` value in Table 6 can also be obtained through the results here.

The results in Figure 13 can be obtained by combining the results in exact algorithms and approximation algorithms.