HW5: K-way Graph Partitioning Using JaBeJa

Authors: Andrei Capastru, Alex Porciani | Group 11

The goal of this assignment is to understand distributed graph partitioning using gossip-based peer-to-peer techniques, such as, JaBeJa described in [F. Rahimian, et al., JA-BE-JA: A Distributed Algorithm for Balanced Graph Partitioning, SASO2013.

1.1 How to run the code

0. Install java and maven on your machine. Edit in `src/main/java/se.kth.jabeja/config/CLI` line 55 - by inserting the name of the wanted graph.
1. Run in unix terminal:

```
cd ~
git clone https://github.com/andreicap/data-mining.git
cd data-mining/HW5
./compile.sh
./run.sh
./plot.sh [output_file]
```

2. Open the `graph.png` file to see the plots of the convergences.

1.2 Task 1

The task is to implement the Ja-Be-Ja algorithm. The implemented classes are: `findPartner` and `SampleAndSwap`, according to the paper.

1.3 Task 2

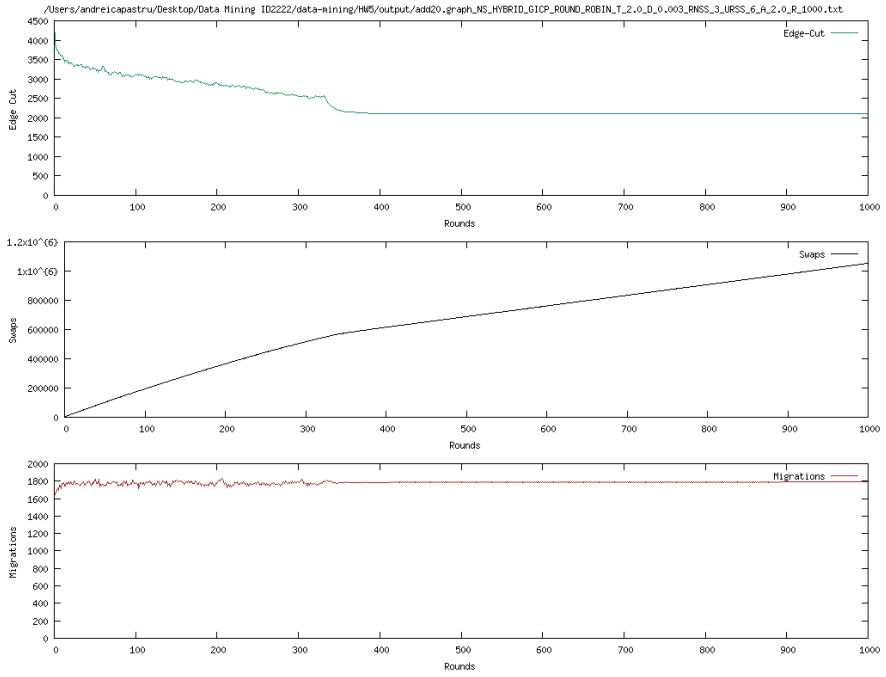
The 2nd task is to tweak different JaBeJa configurations in order to find the smallest edge cuts for the given graphs. We implemented Sampled Annealing.

1.4 Task3 (extra bonus)

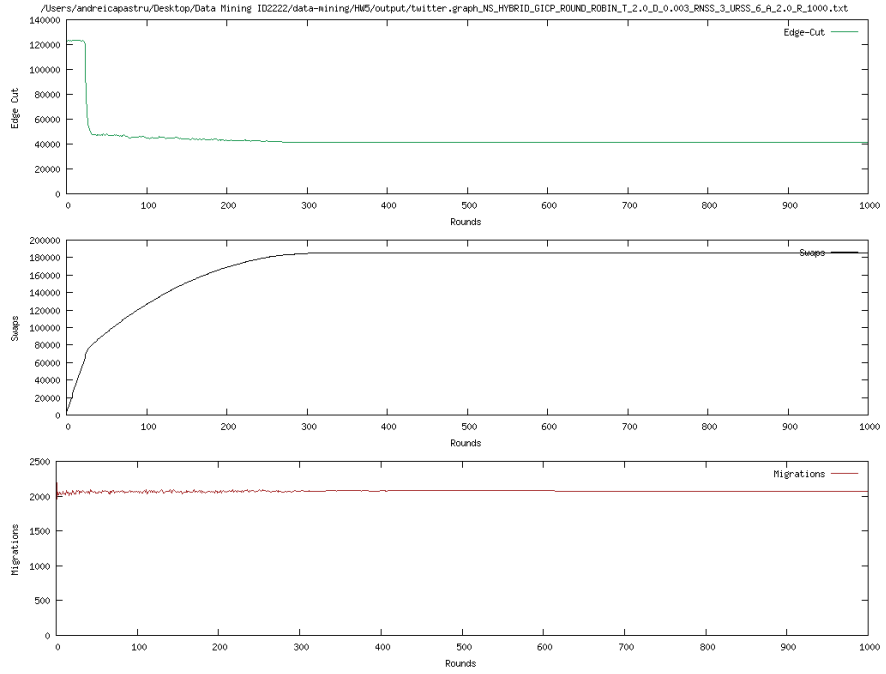
We implemented momentum from Keikha et al. paper named Improved Simulated Annealing using Momentum Terms. We tweaked the updates of T and the updates of acceptance probability to get better results.

TASK 1 – basic implementation without SA acceptance probability and restart:

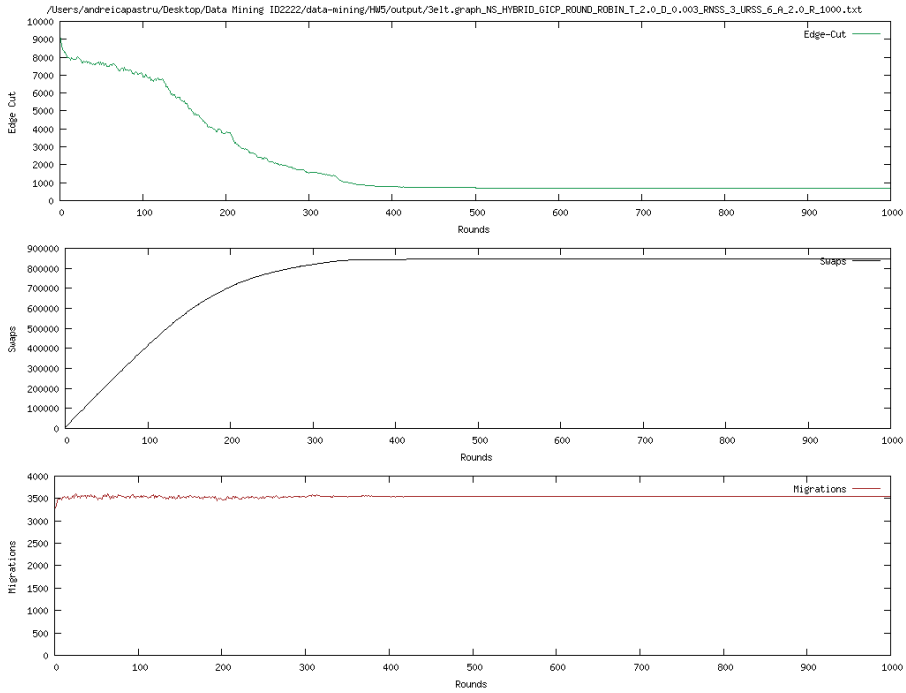
Graph	delta	T_init	alpha	rounds	edge-cut	swaps	migrations
3elt	0.003	2	2	1000	711	845631	3544
Add20	0.003	2	2	1000	2099	1183275	1851
Twitter	0.003	2	2	1000	41312	185499	2075



Add20 – Task 1



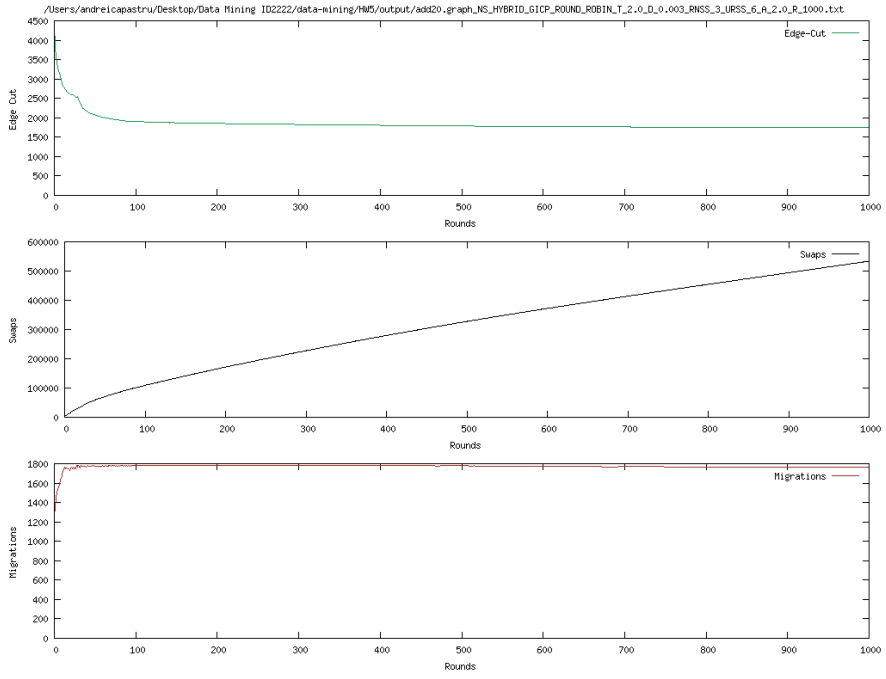
Twitter – Task 1



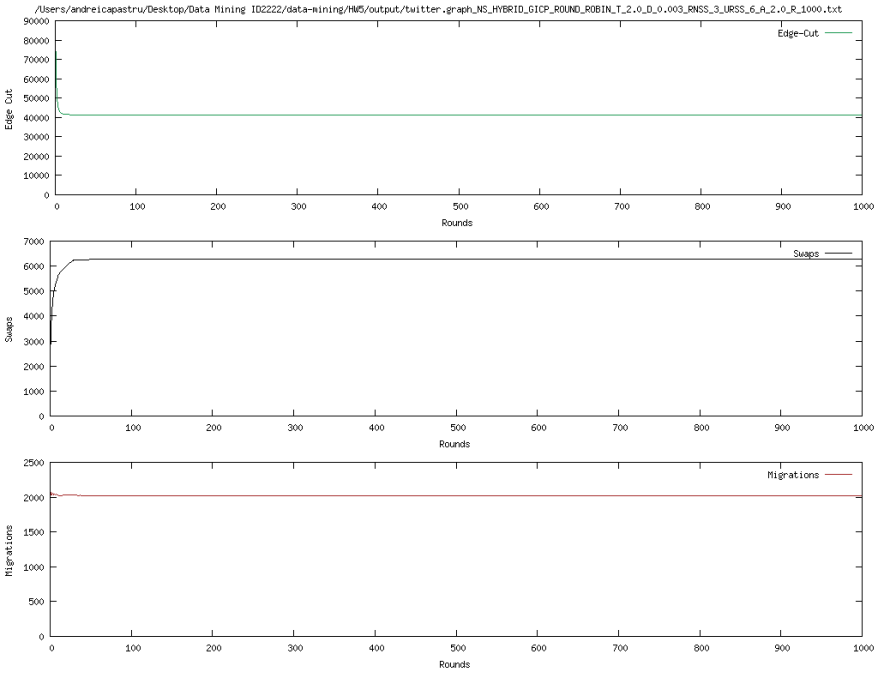
3elt – Task 1

TASK 2 - SA with acceptance probability without restart:

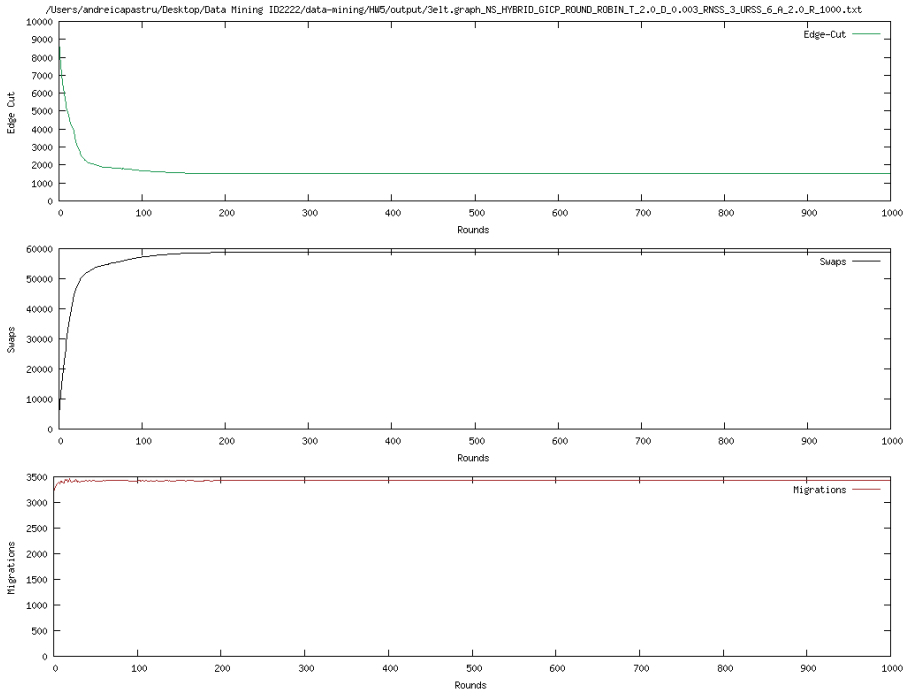
Graph	delta	T_init	alpha	rounds	edge-cut	swaps	migrations
3elt	0.003	1	2	1000	1525	58990	3446
Add20	0.003	1	2	1000	1765	534866	1790
Twitter	0.003	1	2	1000	42135	6238	2032



Add20 – SA



Twitter – SA

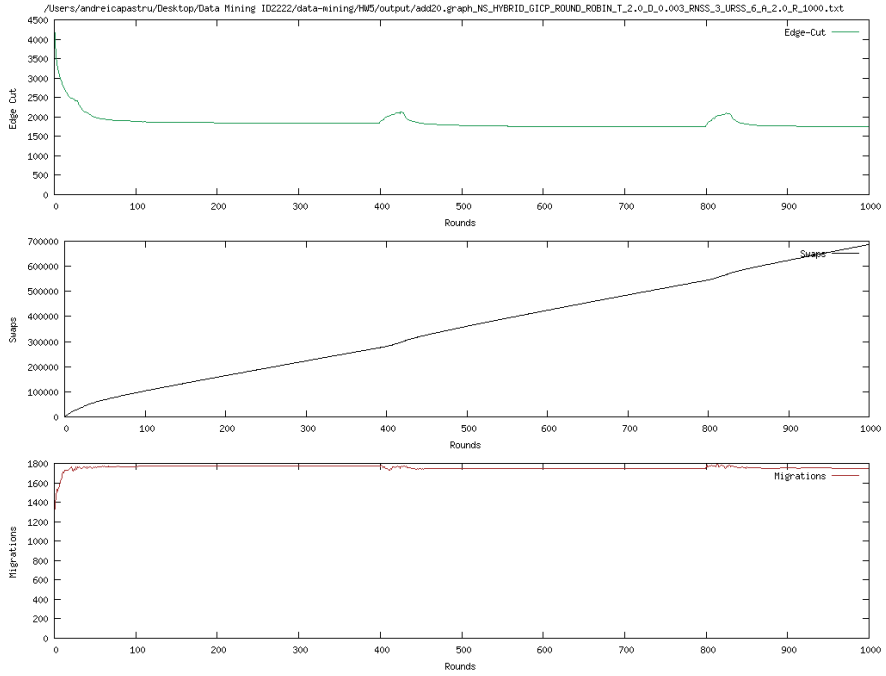


3elt – SA

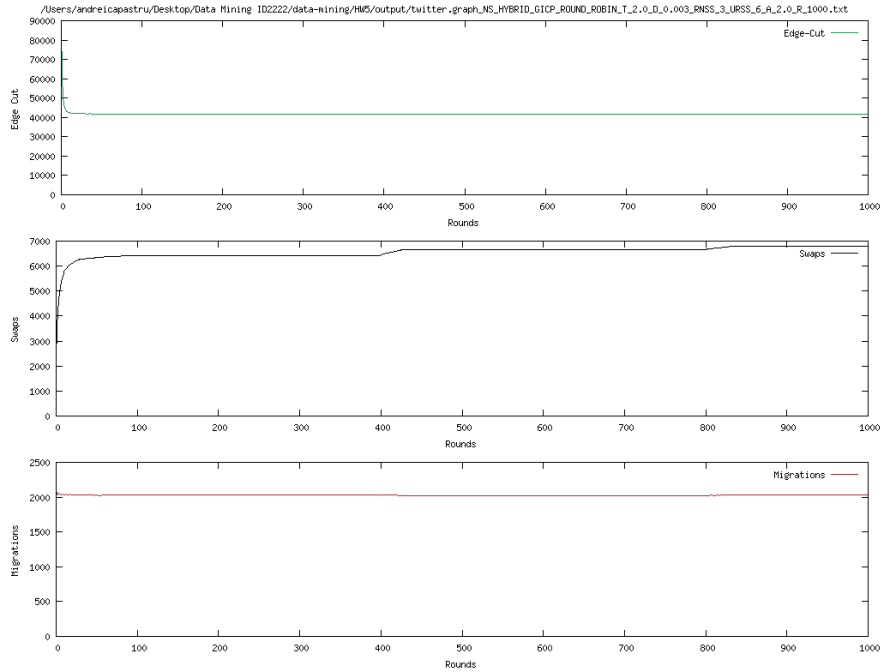
Is it possible to notice that once the parameter T reaches its final value (that is, no more bad swaps are allowed) then Ja-Be-Ja converges to an **edge cut and number of swaps faster** and they not change over time.

TASK 2 - SA with acceptance probability with restart:

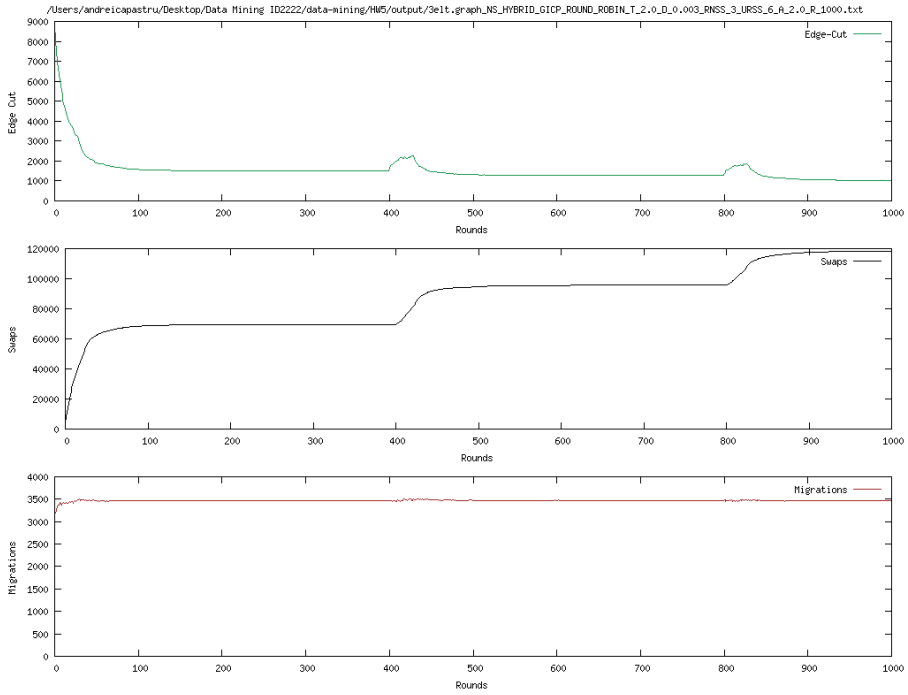
Graph	delta	T_init	alpha	rounds	edge-cut	swaps	migrations
3elt	0.03	1	2	1000	1004	119749	3490
Add20	0.03	1	2	1000	1752	697453	1764
Twitter	0.03	1	2	1000	41065	6856	2095



Add20 – SA & restart



Twitter – SA & restart

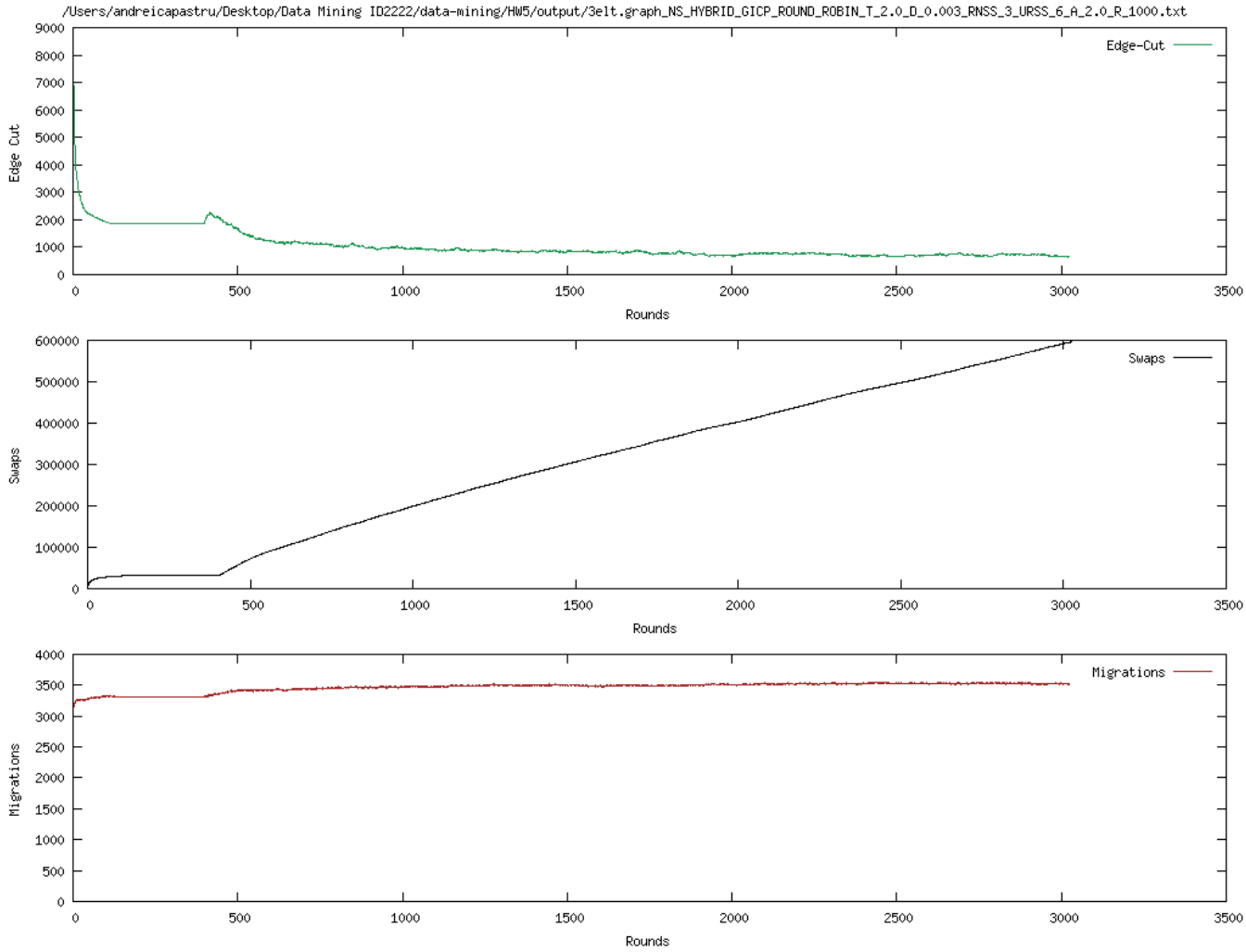


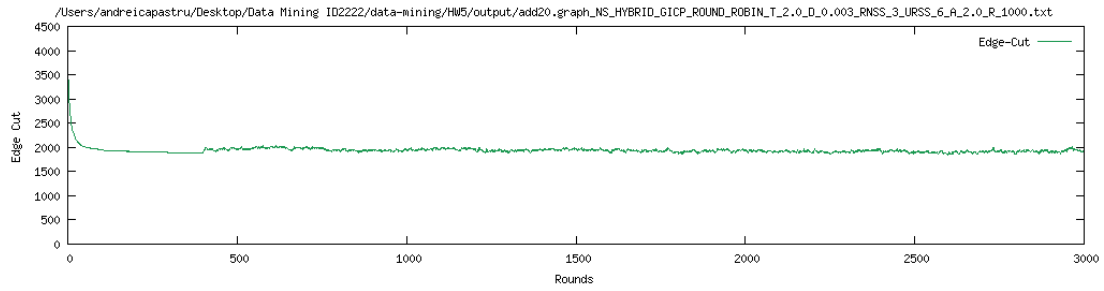
3elt – SA & restart

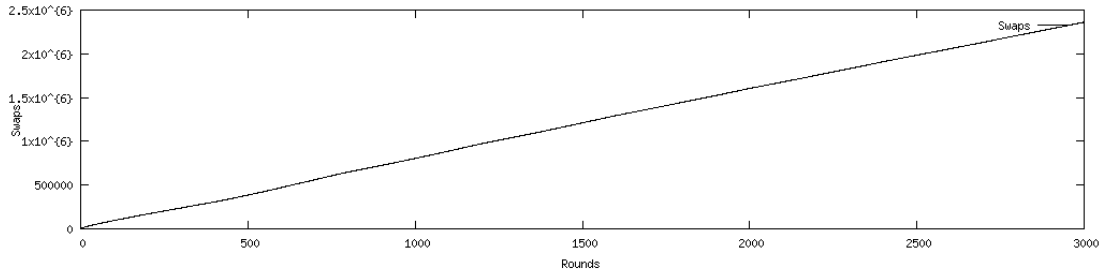
We testested Simultaed annealing techniques with restarts to tune the parameters in order to decrease the overall number of edge cuts. We found out that the solution that enables to have the lowest edge cuts is with restarting after every 400 rounds, using initial T=1 and delta=0.03 (that is 10 times the original delta, because the temperature was decreasing too fast).

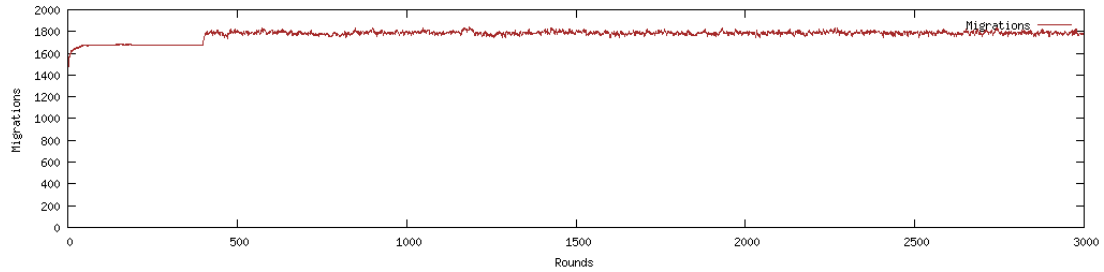
OPTIONAL TASK – using SA with acceptance probability, restarts and momentum:

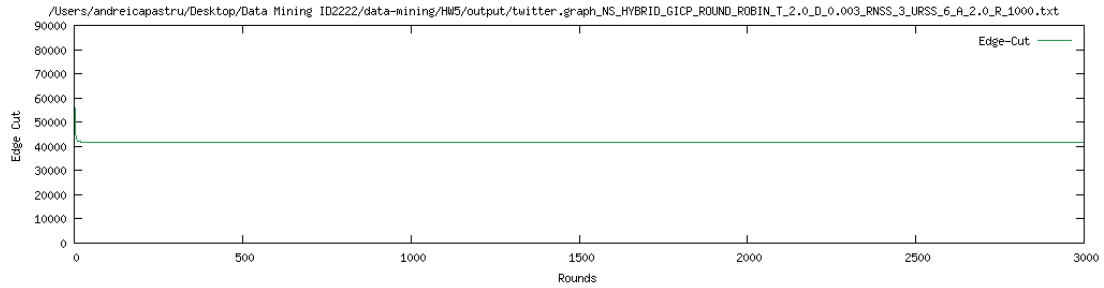
Graph	delta	T_init	T_min	alpha	rounds	edge-cut	swaps	migrations
3elt	0.003	2	0.001	2	3000	654	596142	3528
Add20	0.003	2	0.001	2	3000	1911	2360130	1784
Twitter	0.003	2	0.001	2	3000	41664	28697	2022

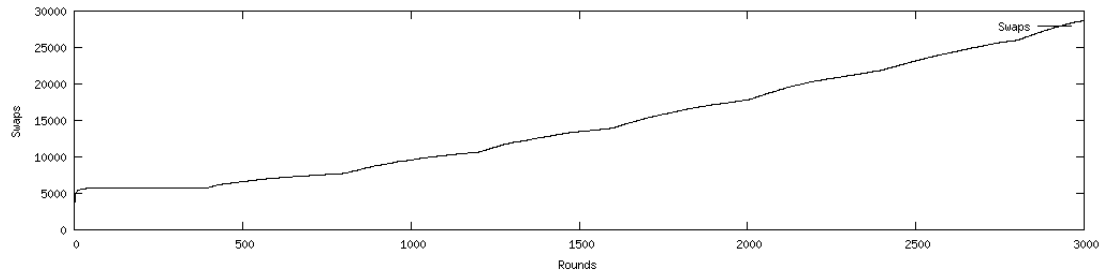


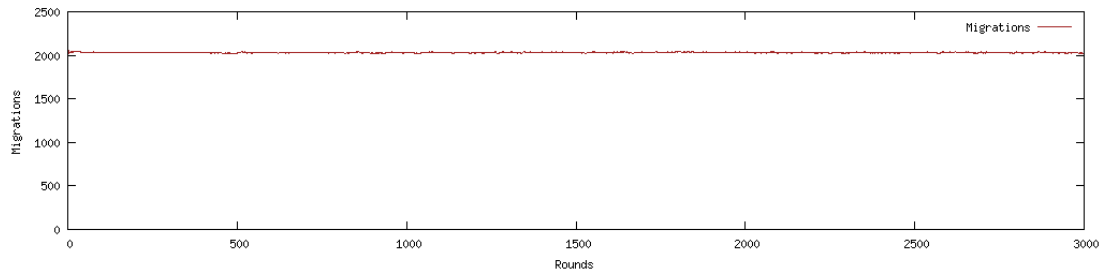

/Users/andreicapastru/Desktop/Data Mining ID2222/data-mining/HW5/output/add20.graph_NS_HYBRID_GICP_ROUND_ROBIN_T_2.0_D_0.003_RNSS_3_URSS_6_A_2.0_R_1000.txt
Edge-Cut
Edge Cut
Rounds


Swaps
Swaps
Rounds


Migrations
Migrations
Rounds


/Users/andreicapastru/Desktop/Data Mining ID2222/data-mining/HW5/output/twitter.graph_NS_HYBRID_GICP_ROUND_ROBIN_T_2.0_D_0.003_RNSS_3_URSS_6_A_2.0_R_1000.txt
Edge-Cut
Edge Cut
Rounds


Swaps
Swaps
Rounds


Migrations
Migrations
Rounds