HOMEWORK 5

K-way Graph Partitioning Using JaBeJa

Fabio Camerota and Gianluca Ruberto

INTRODUCTION

In this Homework we used JaBeJa that is a distributed graph partitioning algorithm described in F. Rahimian, et al., *JA-BE-JA: A Distributed Algorithm for Balanced Graph Partitioning*, SASO2013

In the **first task**, we have implemented the JaBeJa algorithm, completing the already provided functions *sampleAndSwap* and *findPartner*. The default number of rounds was 1000 and the default temperature was 2. In order to decrease the temperature, we used a linear acceptance probability function.

In the **second task**, we used a restart mechanism where we set T=2 (i.e. the initial value) after JaBeJa algorithm converges in order to see if it was able to find a lower number of edge cuts. We also changed the simulated annealing mechanism by using an exponential acceptance probability function as described in simulated-annealing. Moreover, we made multiple tests with alpha=0.8 and alpha=0.5 (default was alpha=0.9) but the results were all similar.

It's important to note that the Simulated Annealing mechanism is necessary, otherwise the algorithm will get stuck in a local optima.

In the **optional task**, we defined a new acceptance probability function based on the sigmoid function and we saw a relevant improvement (i.e. lower number of edge cuts).

The **source code** is also available here.

RESULTS

This is the table that summarizes the results:

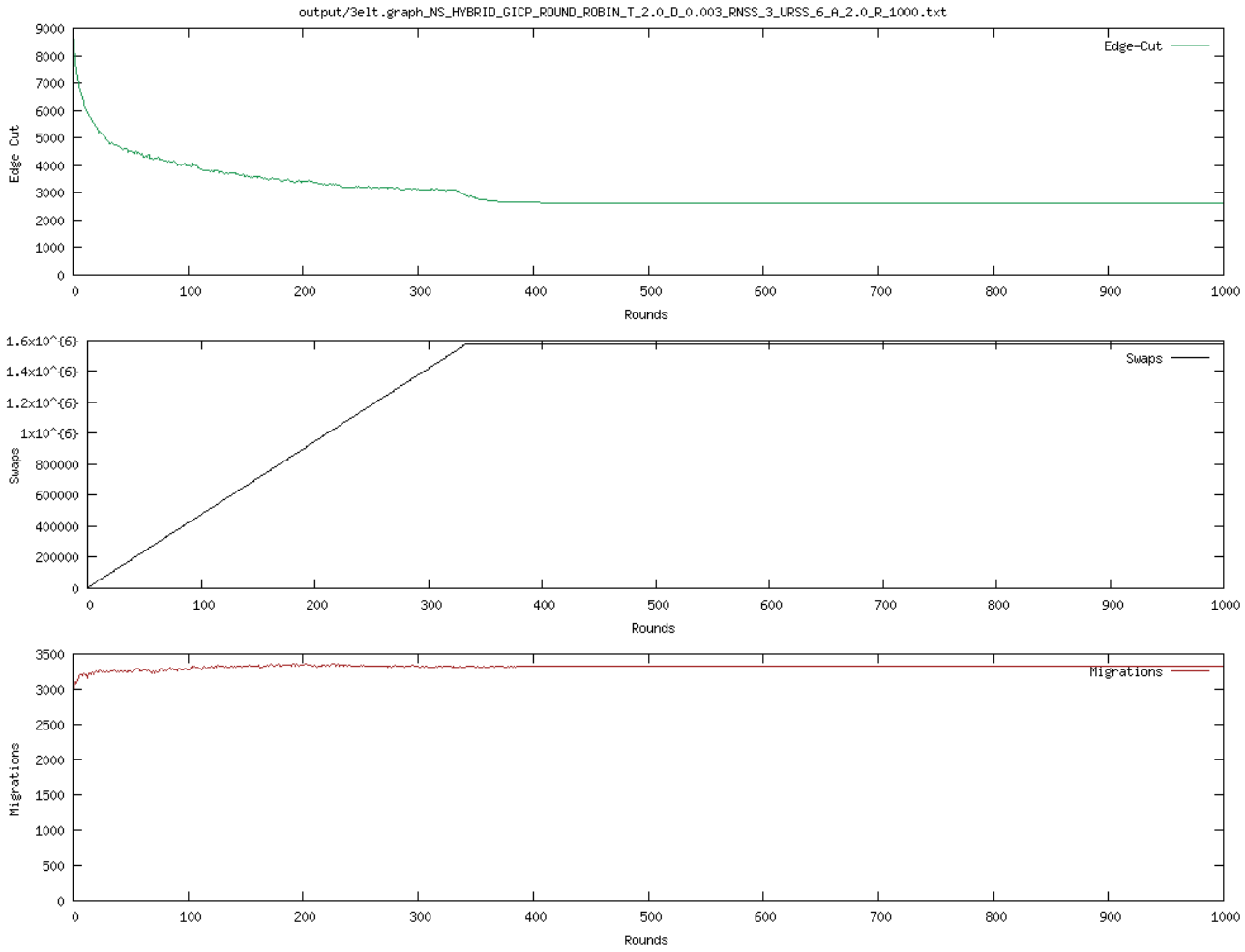
Graph	Algorithm	Number of swaps	Time to converge	Minimum edge cut observed
3elt	JaBeJa	1580209	472	2604
3elt	JaBeJa with T set to 2 at round 400	3158508	805	2269
3elt	Simulated Annealing	4718934	-	3686
3elt	Simulated Annealing with Sigmoid	4427382	-	1311
add20	JaBeJa	1226671	1382	2083
add20	JaBeJa with T set to 2 at round 1500	2310472	2402	2041
add20	Simulated Annealing	11966472	-	2294
add20	Simulated Annealing with Sigmoid	11201760	-	2230
twitter	JaBeJa	899515	813	41156
twitter	JaBeJa with T set to 2 at round 900	1787894	1816	41098
twitter	Simulated Annealing	2240120	-	41255
twitter	Simulated Annealing with Sigmoid	2242149	-	41152

We applied 4 different algorithms:

- JaBeJa: the algorithm described in the above mentioned paper
- JaBeJa with T=2 after x rounds. The number of round was found by looking at where the original JaBeJa algorithm's chart converges.
- Simulated annealing, namely the JaBeJa algorithm with an exponential acceptance probability function.
- Simulated annealing with Sigmoid, namely the JaBeJa algorithm with a sigmoid acceptance probability function.

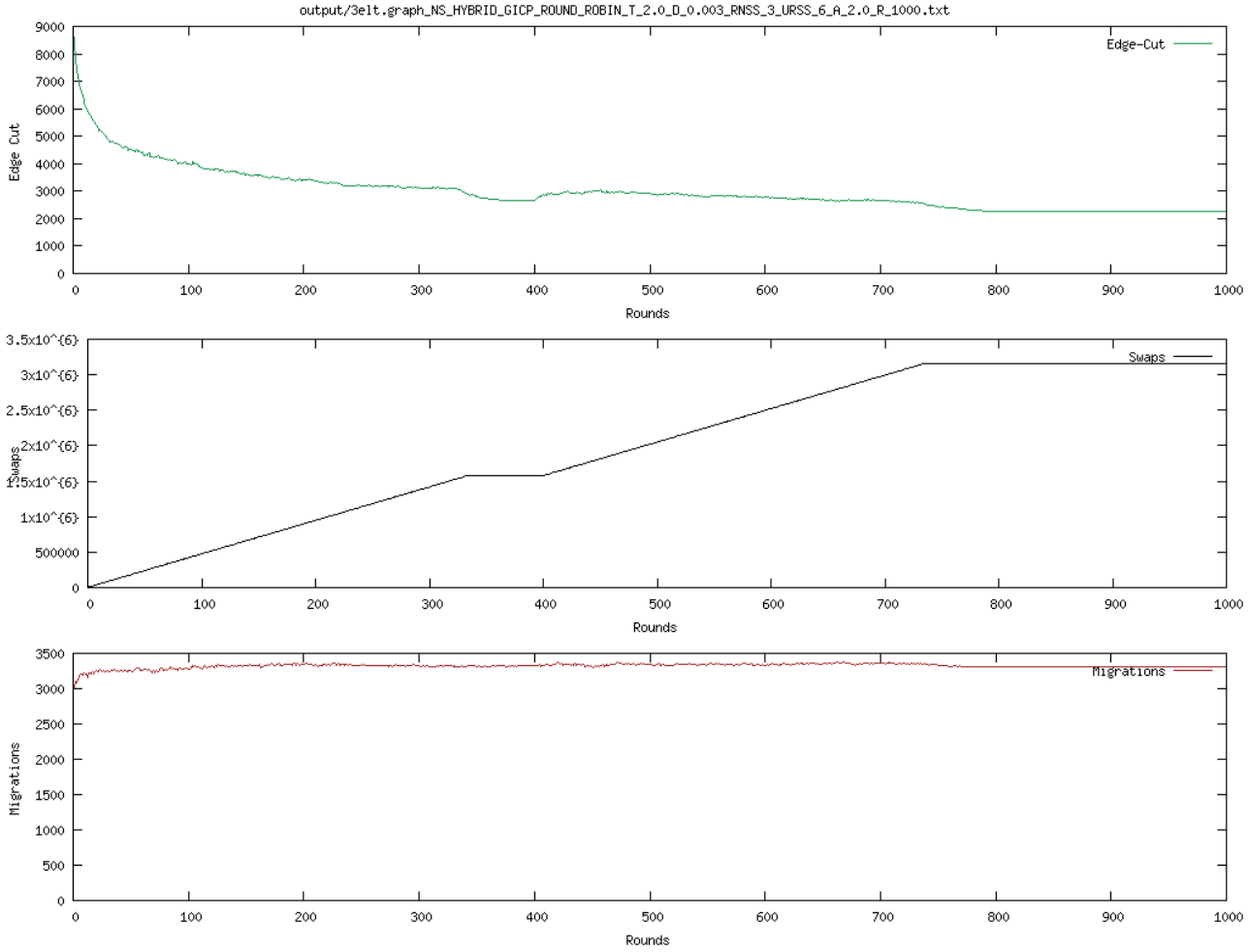
It's important to note that when using simulated annealing with exponential or sigmoid acceptance function, it doesn't converge (as we can see from the charts below). In fact, Simulated Annealing is a method for finding a good (not necessarily perfect) solution.
Regarding the 3elt graph dataset, these are the charts obtained by applying those 4 algorithms.

JaBeJa



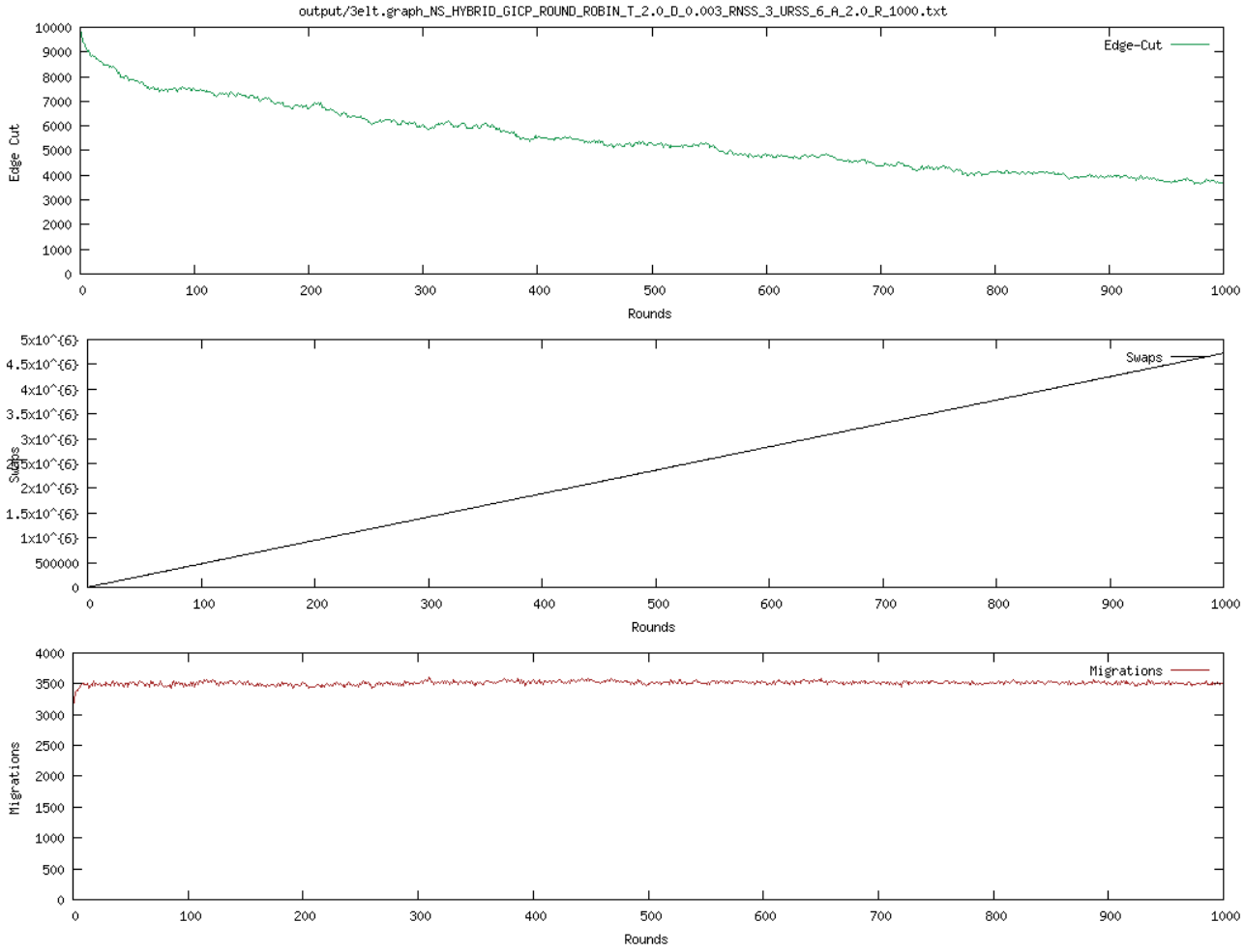
We see that the algorithm converges at round 400, since the number of edge cuts stop decreasing. The same reasoning is valid for the number of swaps.

JaBeJa with T=2 at round 400



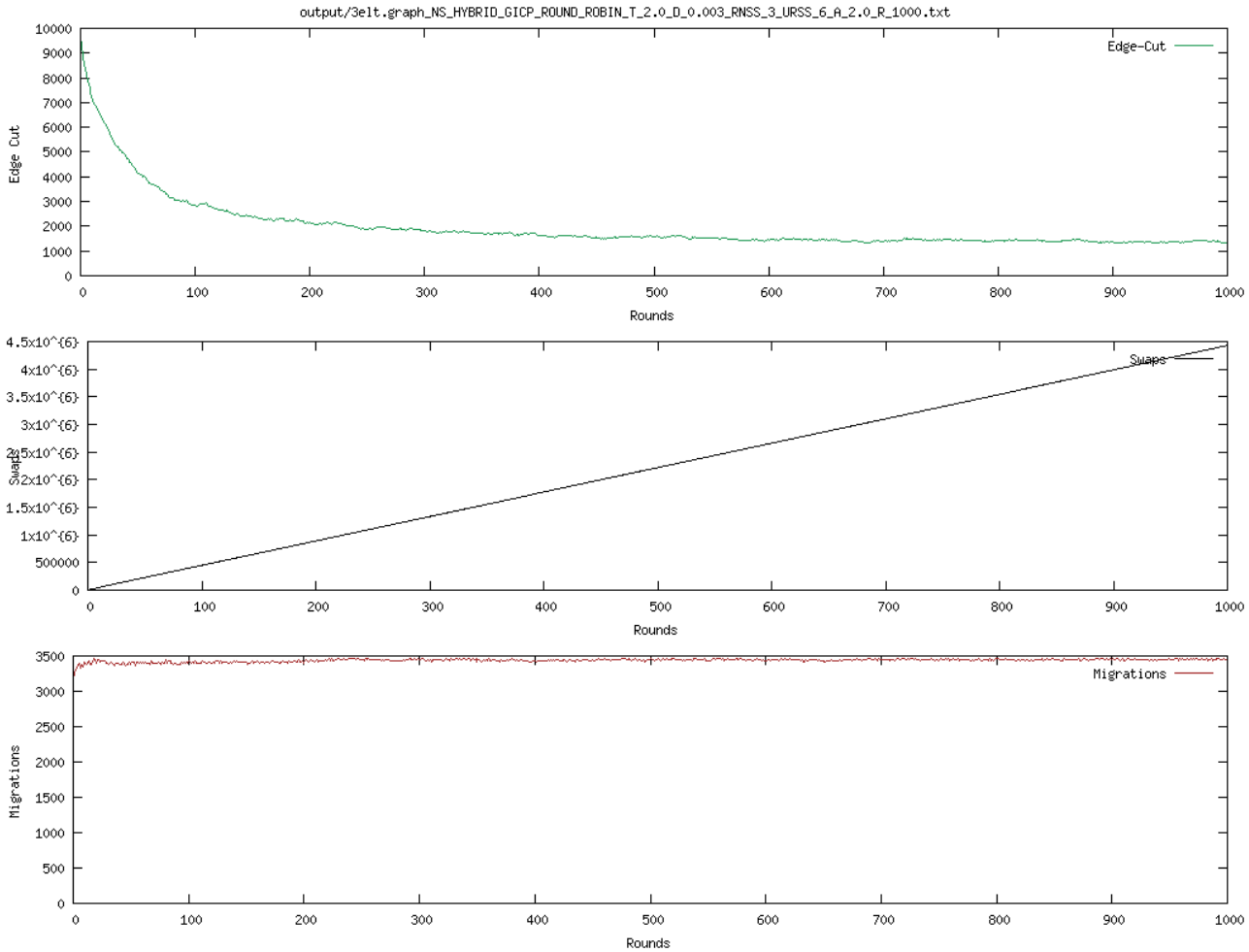
This algorithm shows better performance than the previous one, since we obtain a lower number of edge cuts. So restarting the algorithm when it converges is a good idea, because we're able to find values that are better than the local optimal.

JaBeJa algorithm with an exponential acceptance probability function



Performance is worse than the previous algorithms, in fact at round 1000 it shows an higher number of edge cuts.
The algorithm doesn't converge, in fact we tried also with 5000 and 10000 rounds but still the algorithm didn't converge, in fact also the number of swaps keeps increasing.

JaBeJa algorithm with a sigmoid acceptance probability function



We can see a relevant improvement in the performance since we obtain a lower number of edge cuts compared to the one with an exponential acceptance function.