ID2222 Data Mining Homework 5

Graph Partitioning Using JaBeJa

Luís Santos
lmpss@kth.se

Kevin Dalla Torre
kevindt@kth.se

December 2020

1 Introduction

As instructed in this report we detail our implementation of the JaBeJa algorithm from the paper Rahimian et al. (2013) featuring the K-way graph partitioning algorithm. We have used our implementation to analyze three graph datasets: 3elt, add20 and Twitter. For task 1 we implemented the JaBeJa algorithm, while for task 2 we implemented an exponentially-decreasing simulated annealing policy, and also a mechanism for restarting the temperature. We omit the Twitter plots to conserve space (contact authors for details).

2 Task 1

To implement the JaBeJa algorithm, it was required to modify three methods of the Jabeja class: saCoolDown(), sampleAndSwap() and findPartner(). The implementation follows closely the pseudo-code provided in the original paper. We extended saCoolDown() and findPartner() methods to be called with different annealing policies, which we will detail further in the section for task 2.1. We present the results from our JaBeJa implementation using two different node policies (LOCAL and HYBRID) in Table 1. The annealing policy is linear-decreasing over time as described in the paper. We set $\delta = 0.005$, which resulted in a better edge cut overall for the hybrid policies, and keep the default values for the initial temperature T and alpha α parameters.

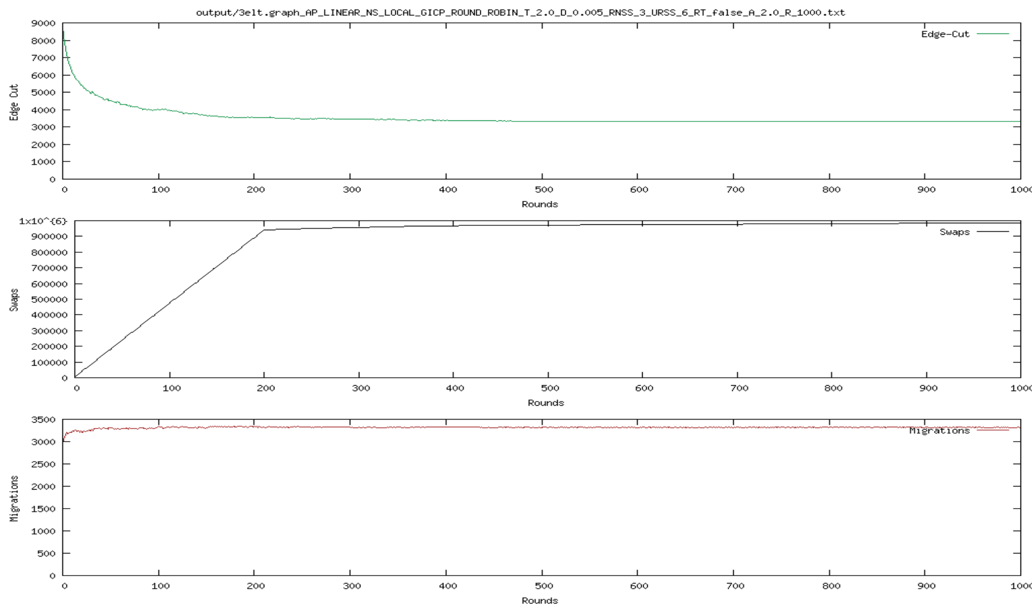

Figure 1: 3elt graph dataset with a LOCAL node policy - edge cut: 3325; swaps: 985412; migrations: 3317.

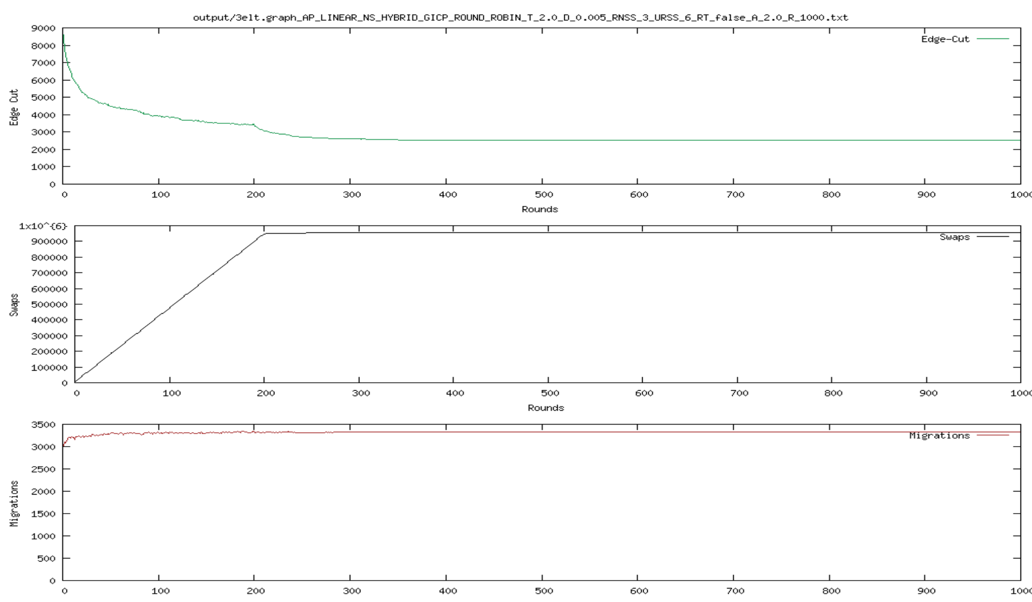

Figure 2: 3elt graph dataset with a HYBRID node policy - edge cut: 2549; swaps: 955211; migrations: 3326.

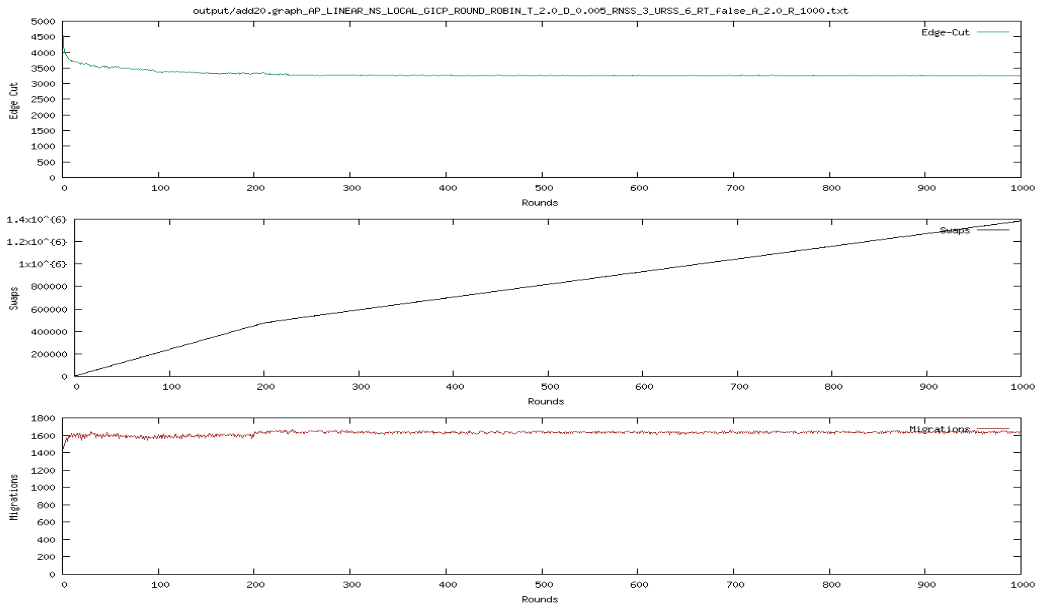

Figure 3: add20 graph dataset with a LOCAL node policy - edge cut: 3239; swaps: 1080338; migrations: 1637.

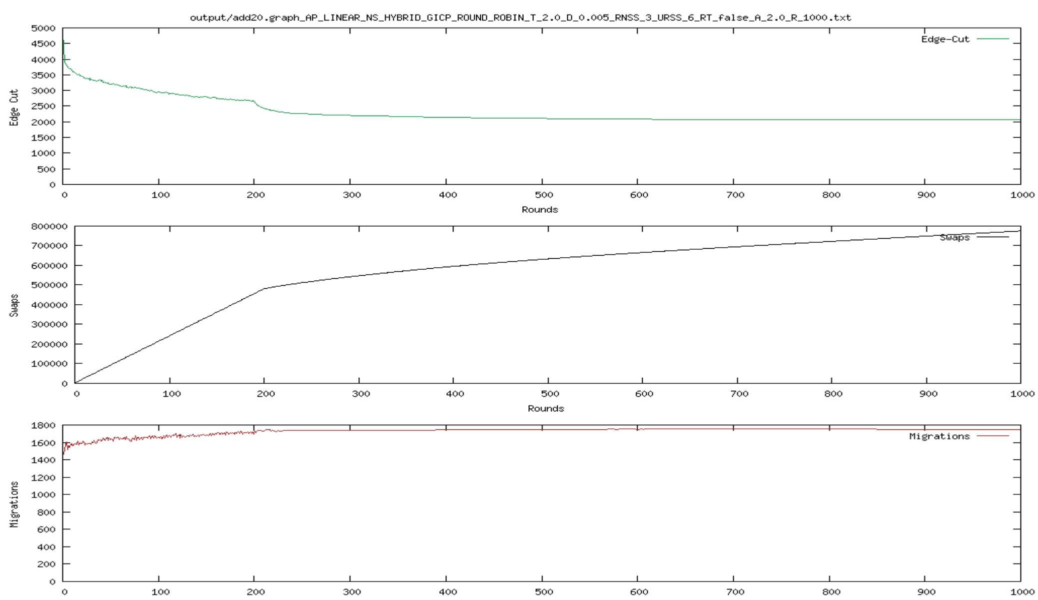

Figure 4: add20 graph dataset with a HYBRID node policy - edge cut: 2063; swaps: 775024; migrations: 1750.

Graph	Node Policy	Edge-cut	Swaps	Migrations
3elt	LOCAL	3325	985412	3317
3elt	HYBRID	2549	955211	3326
add20	LOCAL	3239	1080338	1637
add20	HYBRID	2063	775024	1750
twitter	LOCAL	44131	464239	2049
twitter	HYBRID	41202	536130	2055

Table 1: Different node selection policies using a linear-decreasing annealing policy with $\delta = 0.005$, $T = 2$, $\alpha = 2$.

3 Task 2.1

We implemented an exponentially-decreasing annealing policy[1] for task 2.1. The temperature is decreased by an exponential factor of δ starting at 1 until a minimum value of 0.0001 is reached. This behaviour corresponds to the saCoolDown() method which accepts either a linear or exponential annealing policy. We compute an acceptance probability a_p in the method findPartner() using an old value old_s and a new value new_s as follows: $a_p = e^{(new_s - old_s)/T}$. By analysis of this formula, we conclude that the acceptance probability a_p decreases as the new value gets smaller than the old value, or when the temperature becomes larger and $new_s > old_s$. We present results for different delta values in Table 2 using the exponential annealing policy just described.

Graph	Delta	Edge-cut	Swaps	Migrations
3elt	0.8	1986	28314	3328
3elt	0.9	1665	34197	3365
add20	0.8	2134	571283	1699
add20	0.9	2380	764523	1751
twitter	0.8	41574	6113	2034
twitter	0.9	41566	6736	2027

Table 2: Different δ values using an exponentially-decreasing annealing policy with $T = 1$, $\alpha = 2$ and HYBRID node selection.

[1] http://katrinaeg.com/simulated-annealing.html

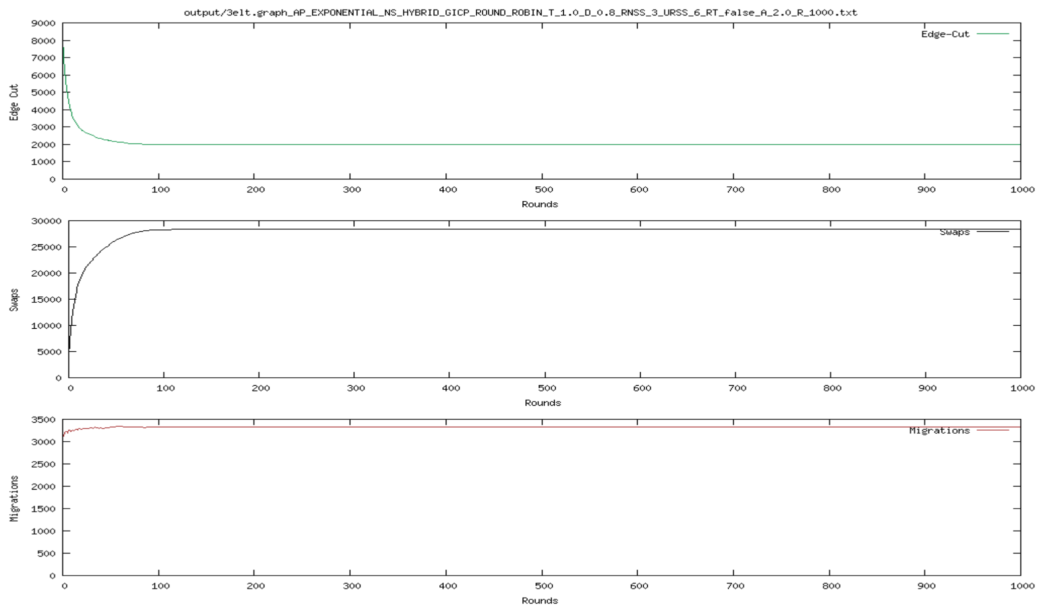

Figure 5: 3elt graph dataset with $\delta = 0.8$ and the exponential annealing policy - edge cut: 1986; swaps: 28314; migrations: 3328.

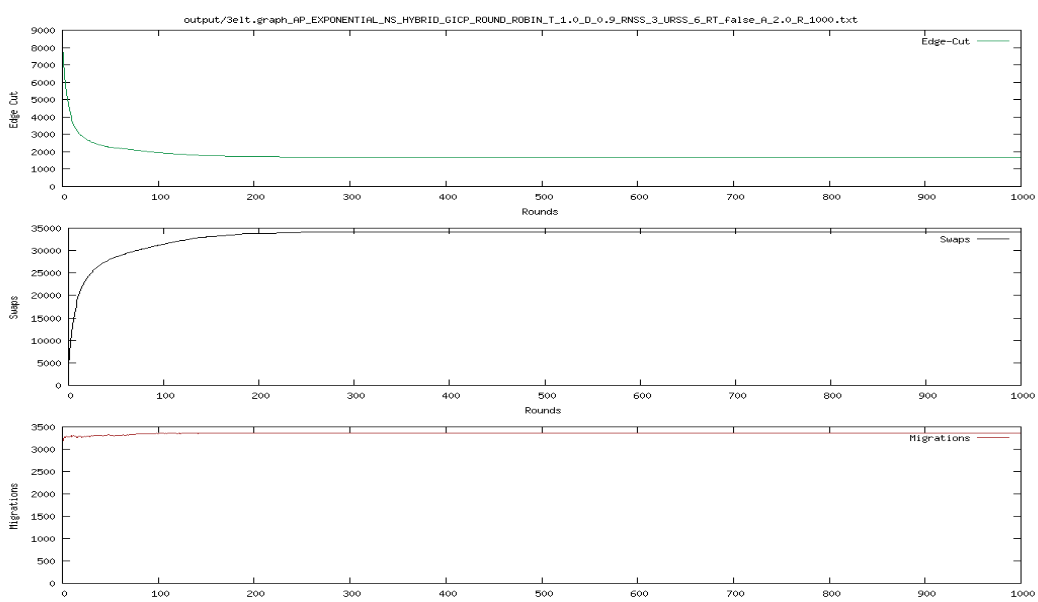

Figure 6: 3elt graph dataset with $\delta = 0.9$ and the exponential annealing policy - edge cut: 1665; swaps: 34197; migrations: 3365.

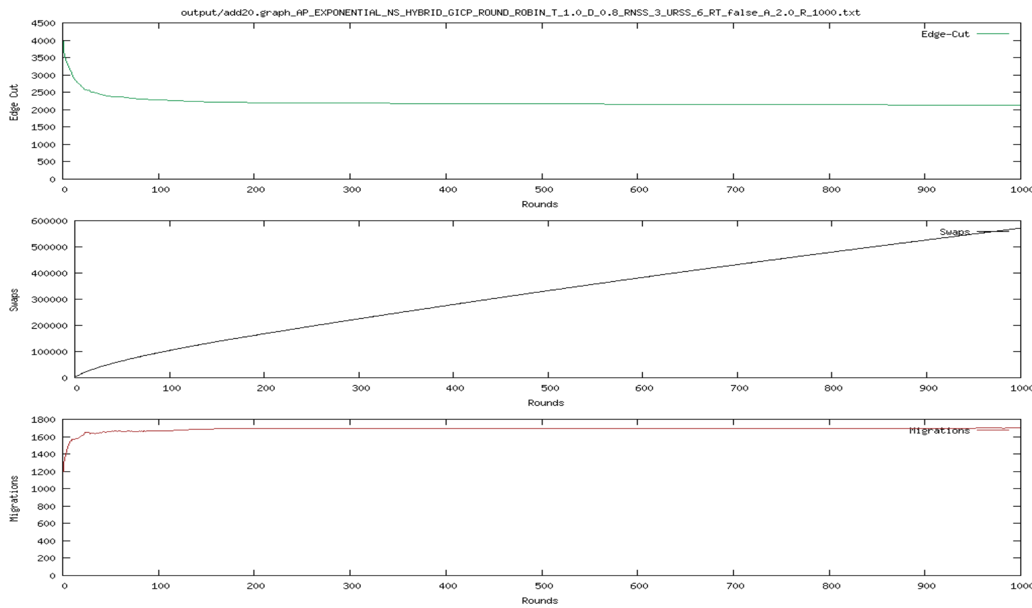

Figure 7: add20 graph dataset with $\delta = 0.8$ and the exponential annealing policy - edge cut: 2134; swaps: 571283; migrations: 1699.

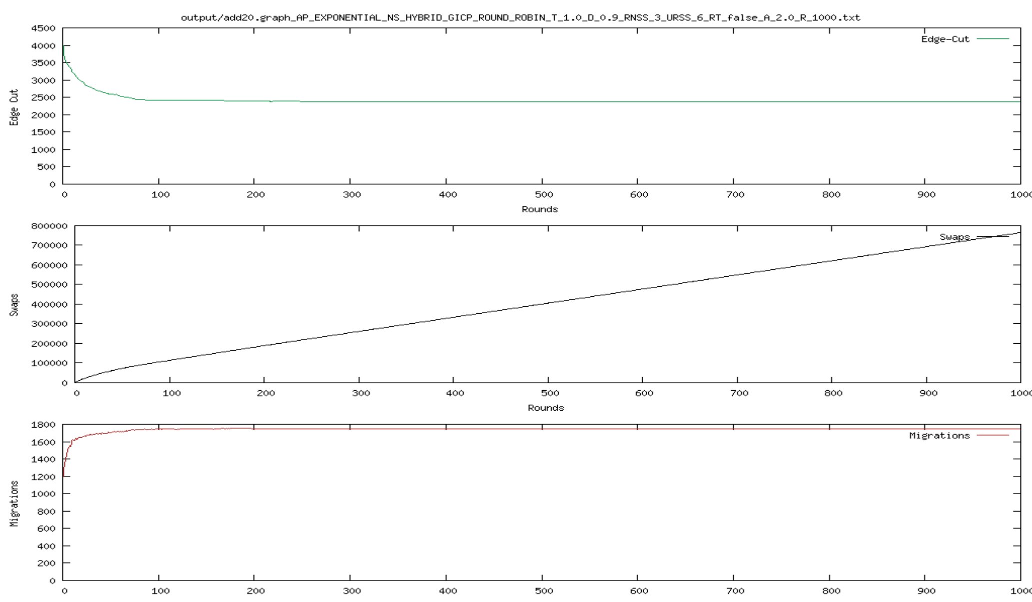

Figure 8: add20 graph dataset with $\delta = 0.9$ and the exponential annealing policy - edge cut: 2380; swaps: 764523; migrations: 1751.

4 Task 2.2

We observed via analysis of metrics that the temperature T can get stuck at a local minimum. Indeed, since the problem at hand is NP-complete, a global minimum may be prohibitive to discover for real-world problems. In order to alleviate this issue, we optionally restart the temperature T after convergence has been detected. To detect convergence, the method restartTemperature() checks if the edge cut has remained constant for a predefined number of rounds, and in that case restarts the temperature to the initial value. The results are shown at Table 3 both for the linear and exponential annealing policies. We use values $\delta = 0.005$ and $\delta = 0.9$ for the linear and exponential policies respectively with 100 rounds of constant edge cut before temperature is restarted. We report only the best solution since the final solution may end up at a sub-optimal value. All other parameters retain their default values.

Graph	Annealing	Delta	Edge-cut	Swaps	Migrations
3elt	Linear	0.005	2016	1908796	3351
3elt	Exponential	0.9	1597	35177	3378
add20	Linear	0.005	2063	775024	1750
add20	Exponential	0.9	2380	345483	1751
twitter	Linear	0.005	41202	536130	2055
twitter	Exponential	0.9	41555	7885	2032

Table 3: Different annealing policies and δ values with temperature restarting after 100 rounds of constant edge cut. Here we report the best found solution.

5 Task 3

For the final task, we implemented two modifications for enhacing the performance. Firstly, we changed the acceptance probability function as follows:

$$a_p = e^{\frac{\frac{1}{old_s} - \frac{1}{new_s}}{T}}$$

This equation for a_p resulted in better edge-cuts overall. We hypothesise that might be related with the fact this formula is able to get closer to zero than the exponential annealing function from task 2.1. As new_s tends to zero this

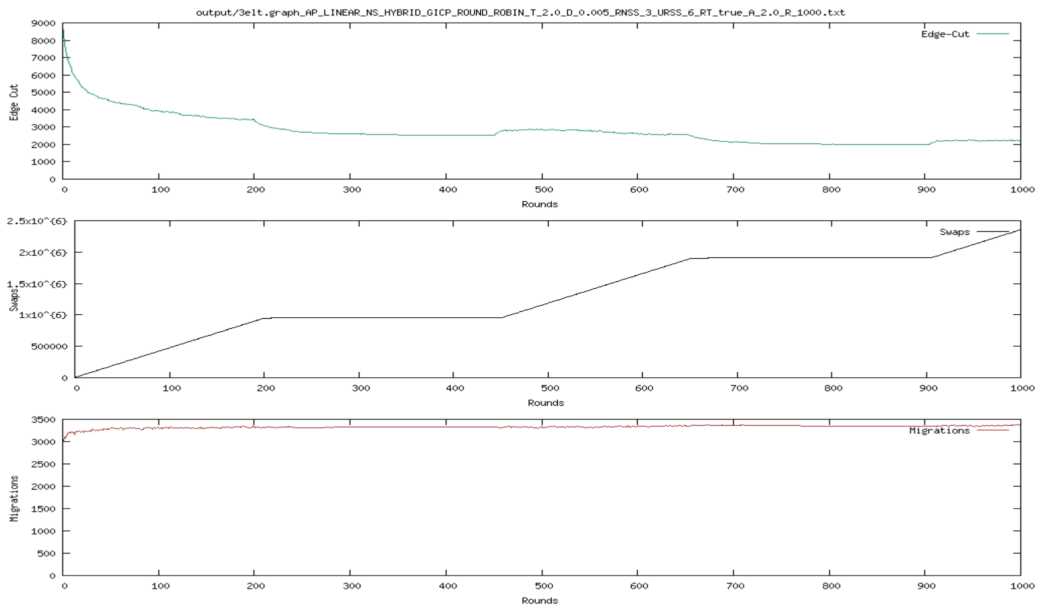

Figure 9: 3elt graph dataset with $\delta = 0.005$, temperature restart and linear annealing policy - edge cut: 2016; swaps: 1908796; migrations: 3351.

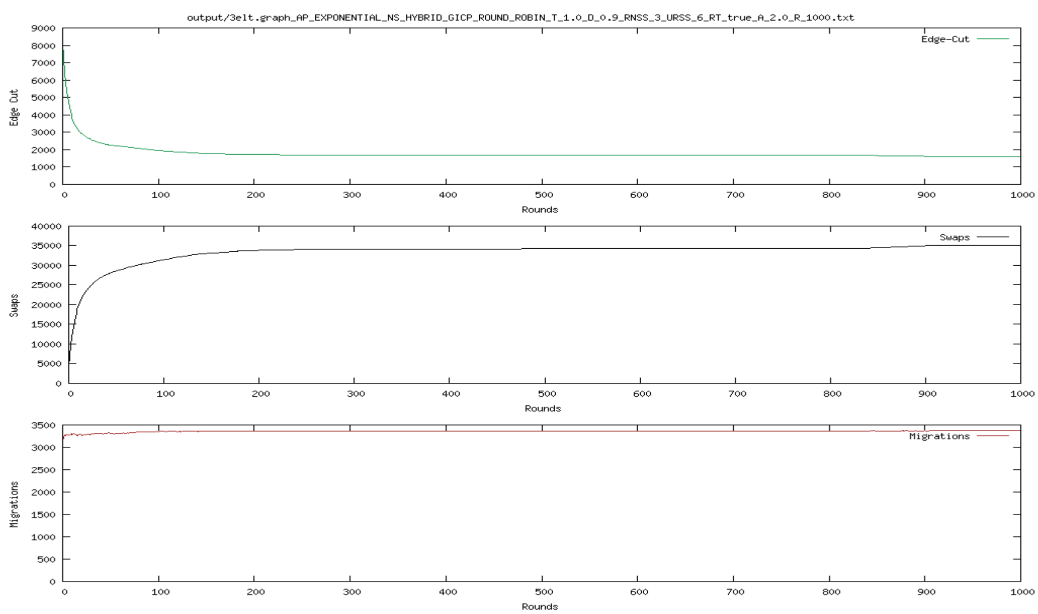

Figure 10: 3elt graph dataset with $\delta = 0.9$, temperature restart and exponential annealing policy - edge cut: 1597; swaps: 35177; migrations: 3378.

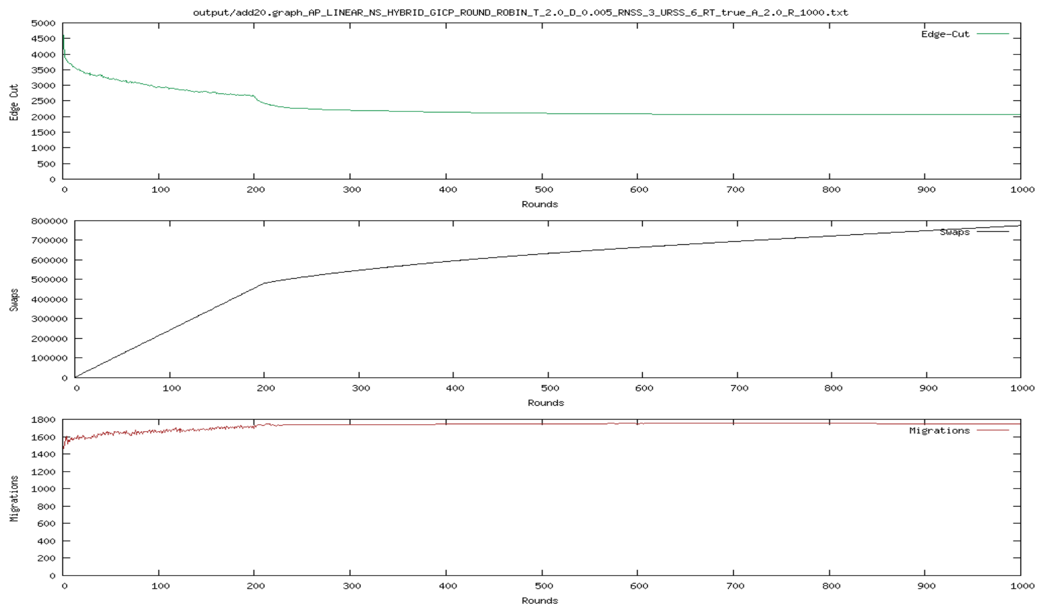

Figure 11: add20 graph dataset with $\delta = 0.005$, temperature restart and linear annealing policy - edge cut: 2063; swaps: 775024; migrations: 1750.

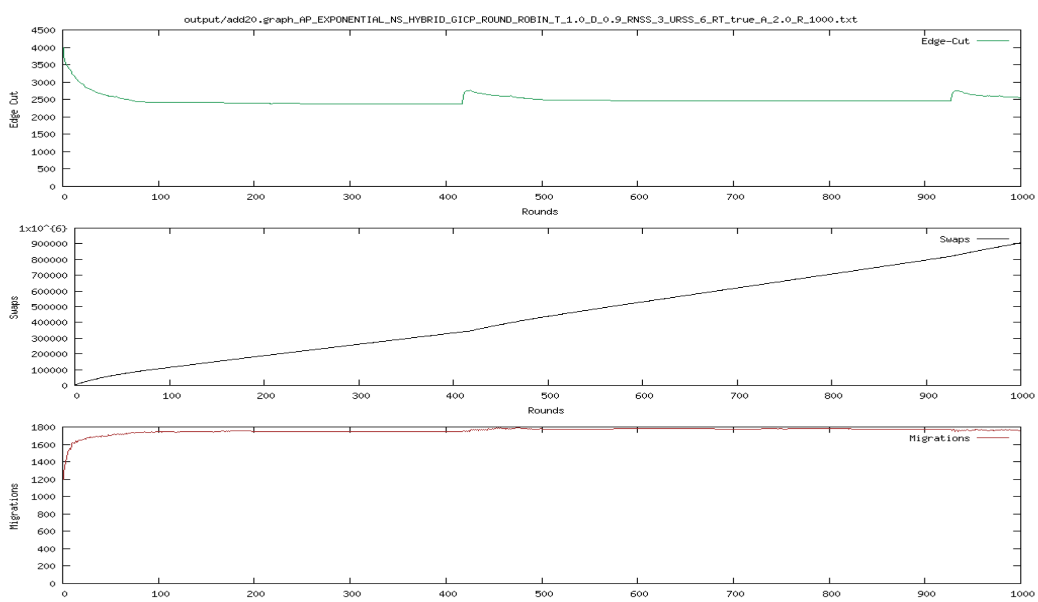

Figure 12: add20 graph dataset with $\delta = 0.9$, temperature restart and exponential annealing policy - edge cut: 2380; swaps: 345483; migrations: 1751.

equation becomes $e^{-\infty/T} = 0$, whereas the previous equation tends instead to $e^{-old_s/T}$. The second improvement we made was to decay delta each time the temperature is restarted. We couple this with the temperature restarting behaviour from task 2.2. We thus decay delta as follows, where δ^{t+1} is the post-restart δ value and δ_d is the decay factor.

$$\delta^{t+1} = \frac{\delta^t}{1 + \delta_d}$$

We observed the metrics changing drastically right after restarting the temperature, as can be seen in Figure 13 for the edge-cuts metric. Decaying delta led to faster updates and thus faster convergence properties after a restart. In Table 4 we show the results for the improved annealing policy with delta decaying on restarts and initial $\delta = 0.9$. Delta decaying was important for the 3elt dataset, but not for the other two graphs. We managed to improve the edge-cuts for 3elt and add20 graphs using these modifications, but not for the Twitter dataset. As the authors state in the paper, Jabeja performs especially well for social network graphs, so this behaviour is to be expected.

Graph	Delta Decay	Edge-cut	Swaps	Migrations
3elt	1	1363	327530	3519
3elt	2.5	1170	310325	3463
add20	1	1789	583455	1796
add20	2.5	1789	583455	1796
twitter	1	87373	312396	2020
twitter	2.5	87373	312396	2020

Table 4: Different delta decays with initial $\delta = 0.9$ with temperature restarting after 100 rounds of constant edge cut and improved annealing policy.

References

F. Rahimian, A. H. Payberah, S. Girdzijauskas, M. Jelasity, and S. Haridi. Ja-be-ja: A distributed algorithm for balanced graph partitioning. In *International Conference on Self-Adaptive and Self-Organizing Systems*, pages 51–60. IEEE, 2013.

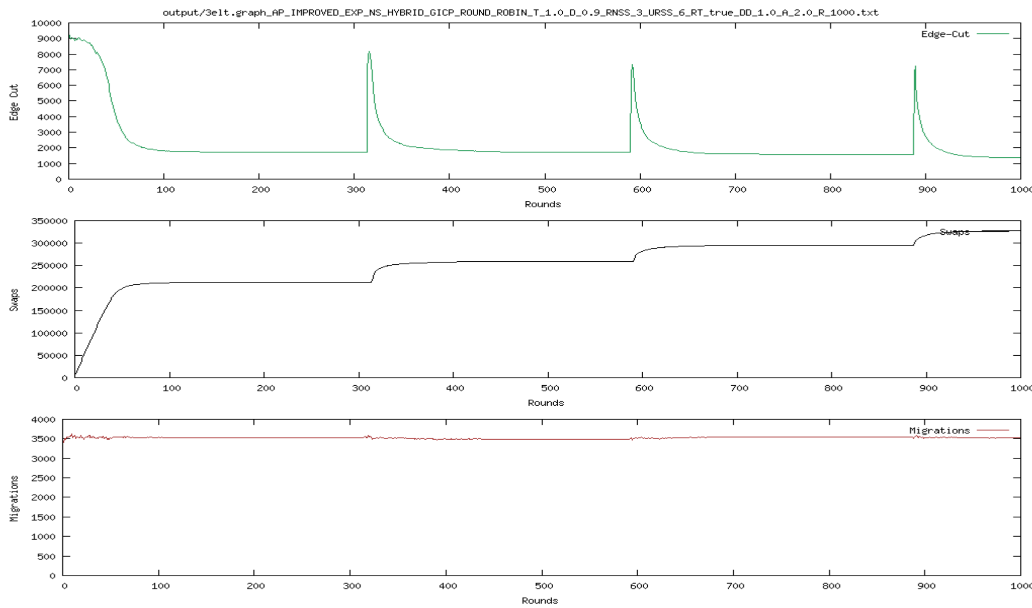

Figure 13: 3elt graph dataset with $\delta_d = 1$, temperature restart and improved annealing policy - edge cut: 1363; swaps: 327530; migrations: 3519.

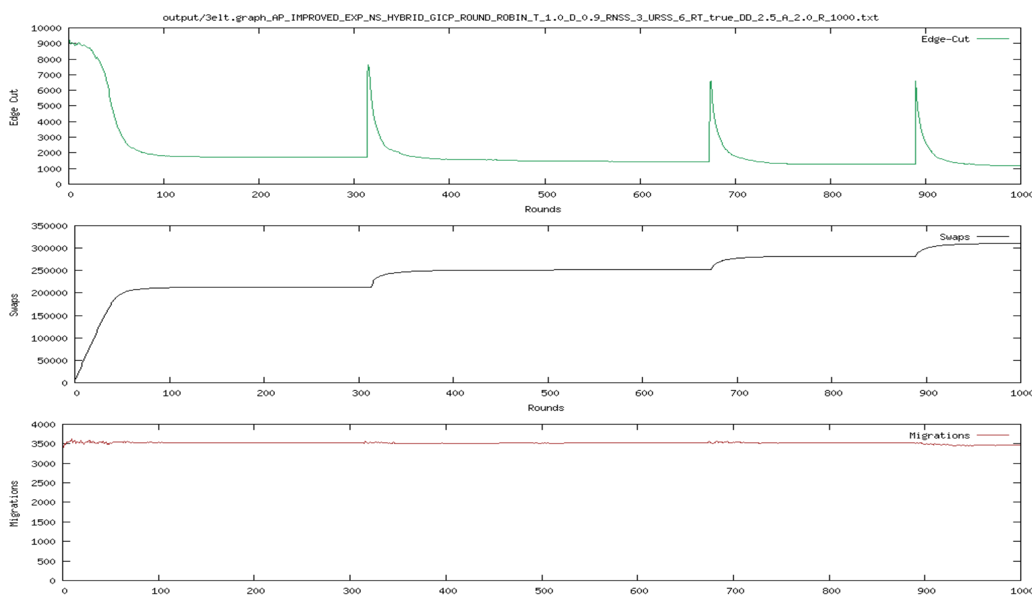

Figure 14: 3elt graph dataset with $\delta_d = 2.5$, temperature restart and improved annealing policy - edge cut: 1170; swaps: 310325; migrations: 3463.

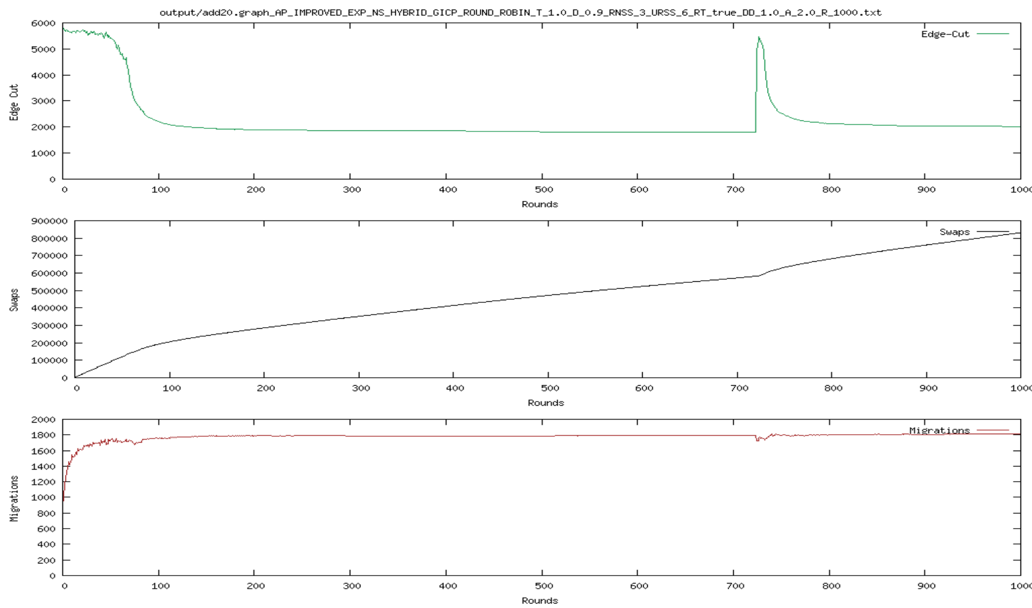

Figure 15: add20 graph dataset with $\delta_d = 1$, temperature restart and improved annealing policy - edge cut: 1789; swaps: 583455; migrations: 1796.

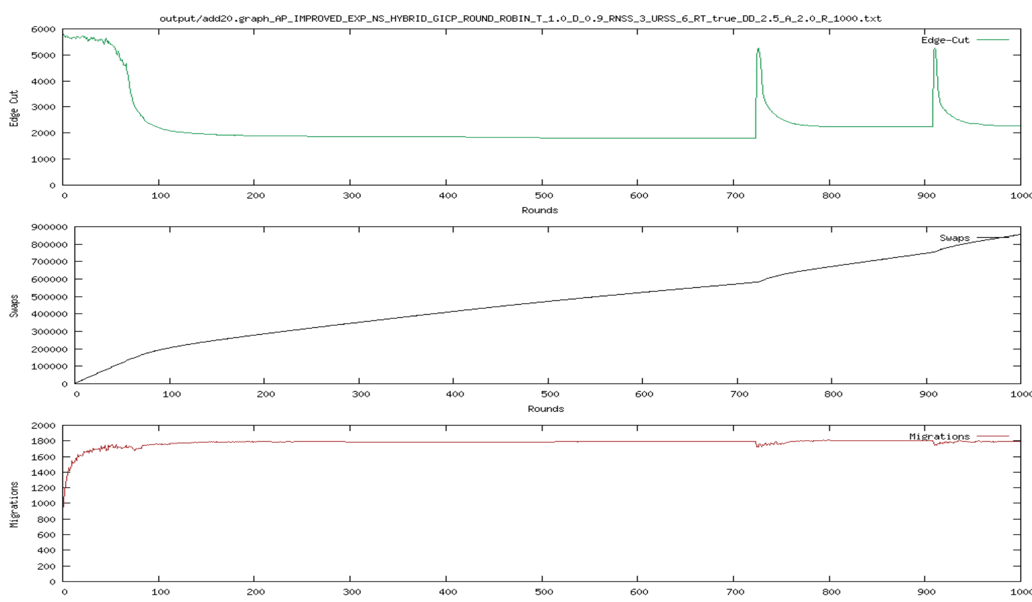

Figure 16: add20 graph dataset with $\delta_d = 2.5$, temperature restart and improved anealing policy - edge cut: 1789; swaps: 583455; migrations: 1796.