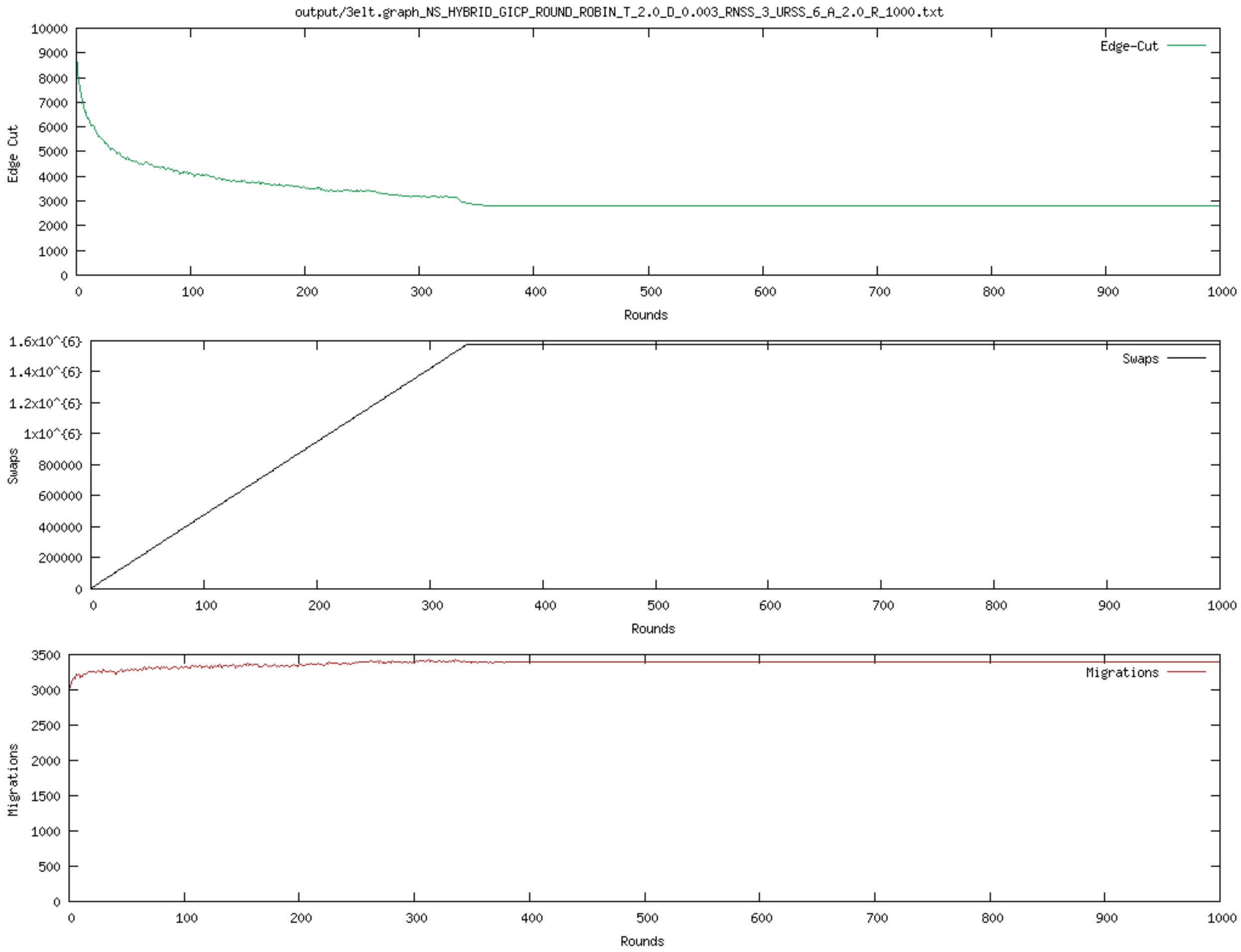
Homework 5

Default parameters

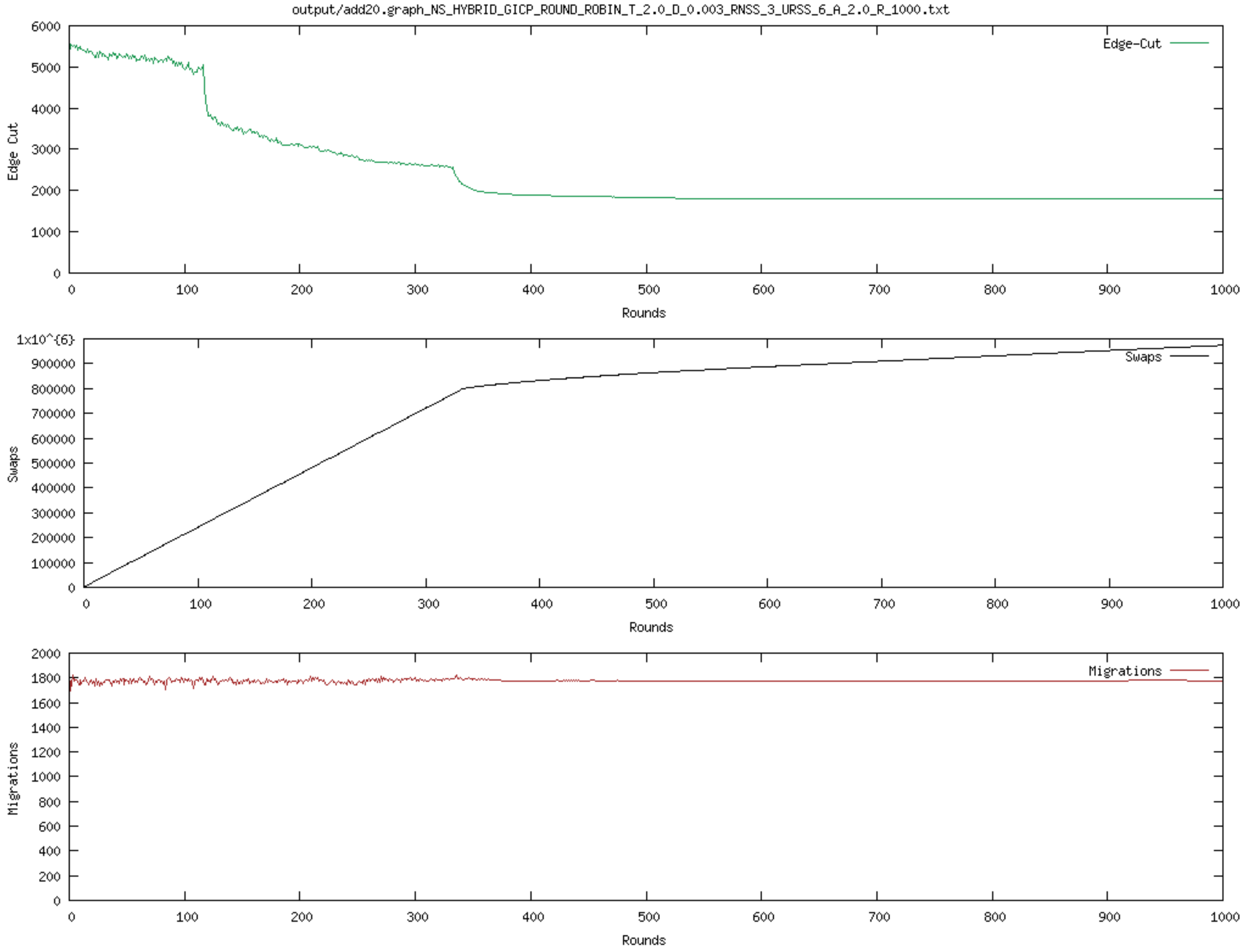
- TEMPERATURE = 2;
- DELTA = 0.003;
- ALPHA = 2;

Task 1 (default parameters) - LINEAR annealer

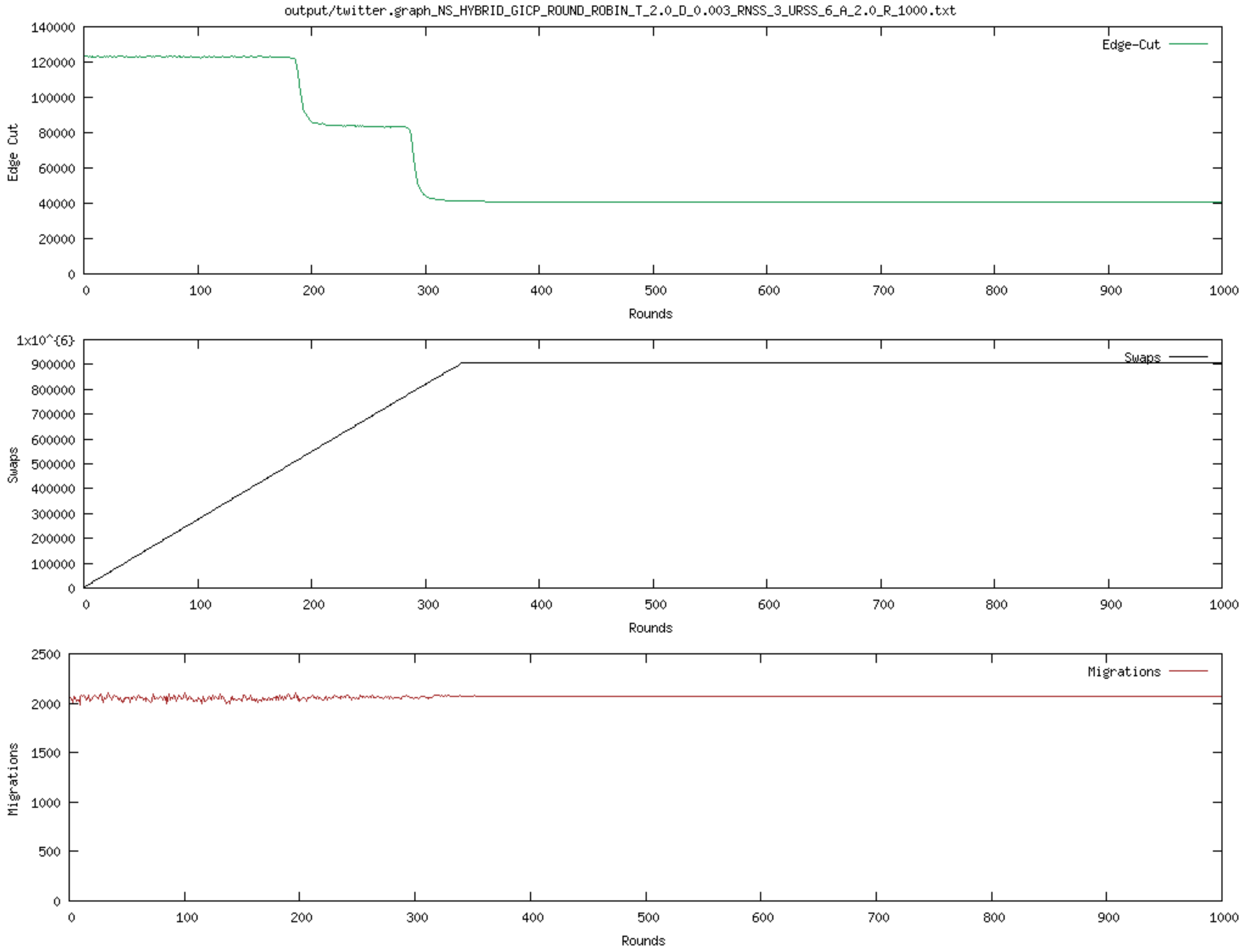
3elt ~ 8 seconds, edge_cut=2788



add20 ~ 6 seconds, edge_cut=1797



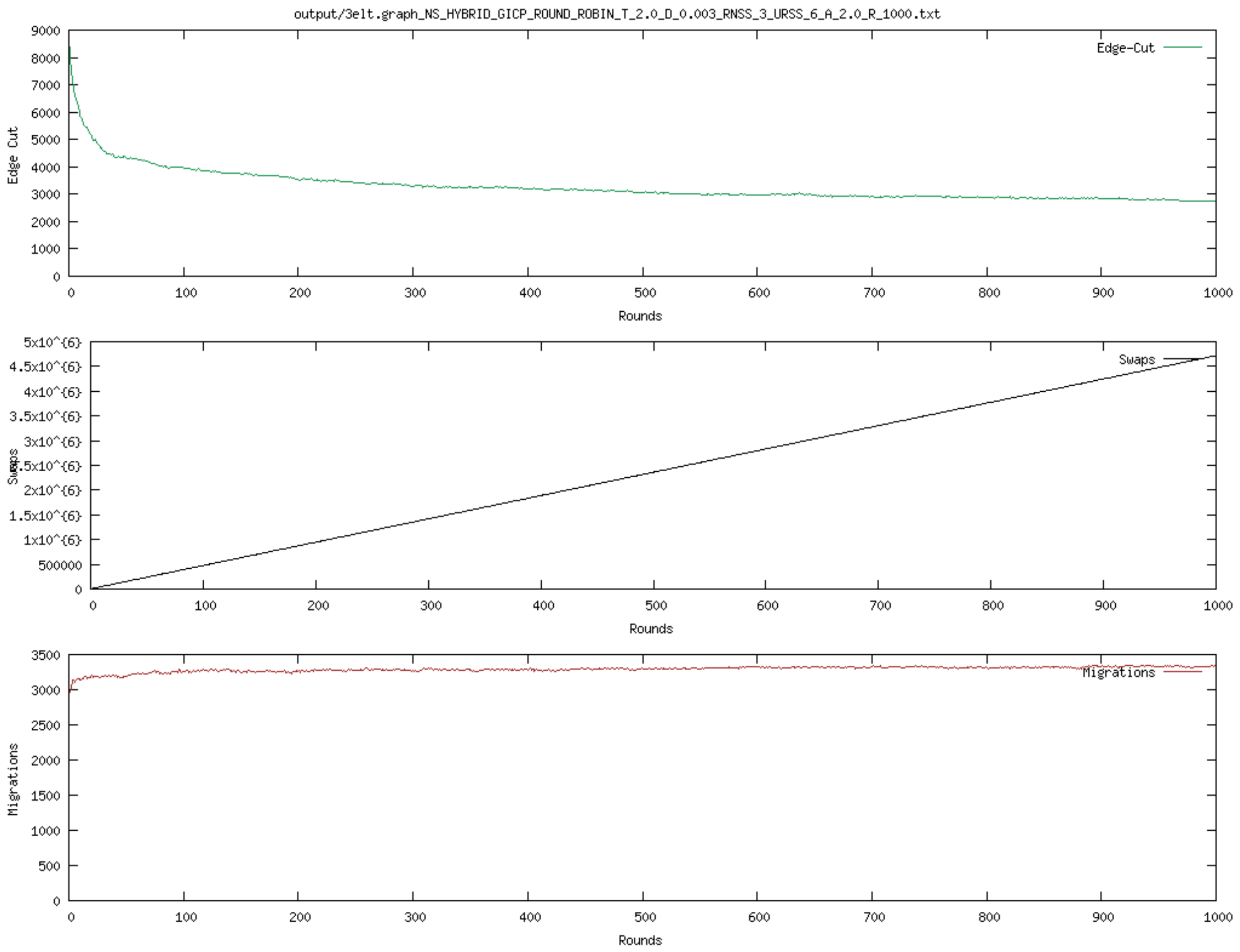
Twitter ~ 116 seconds, edge_cut=40964



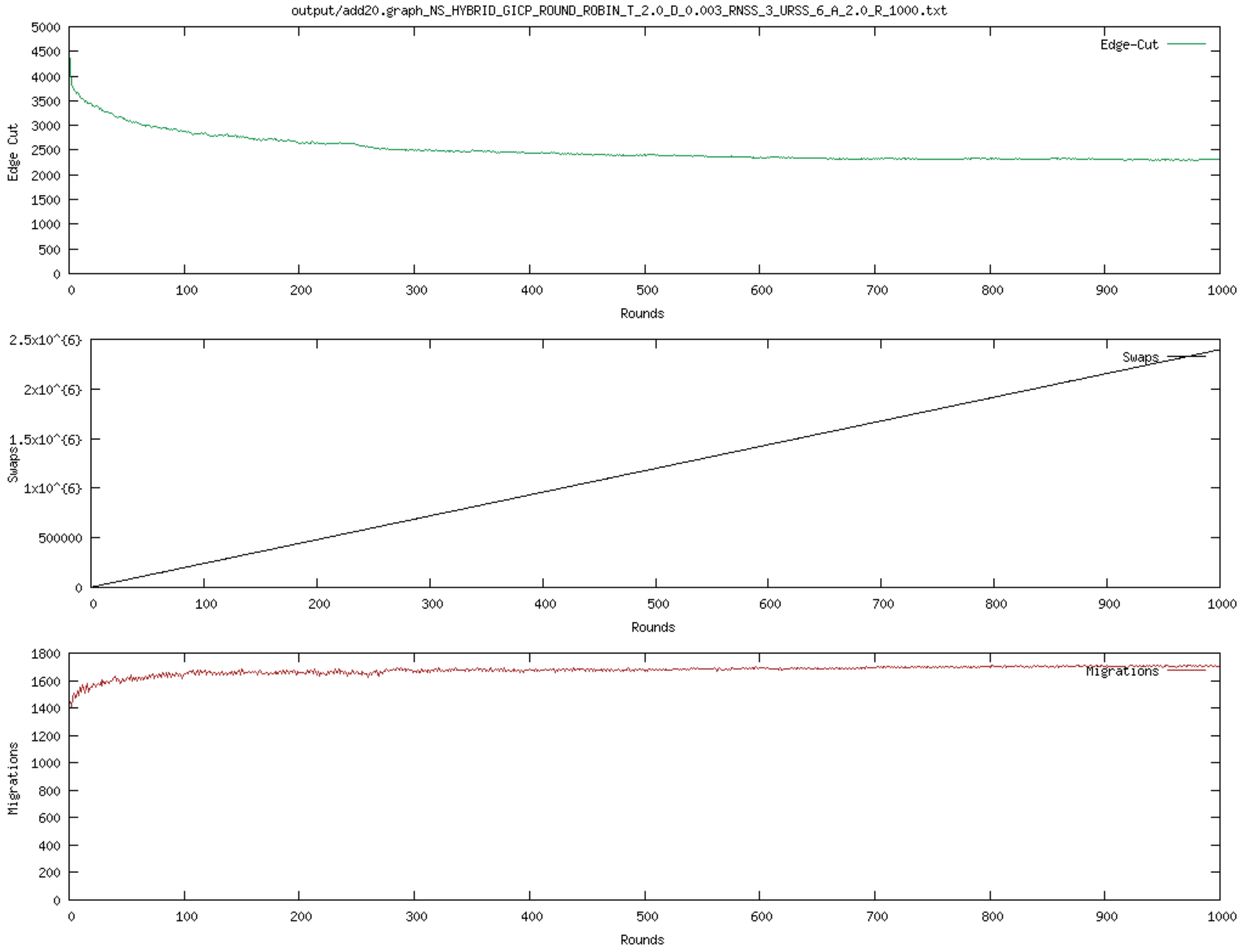
Task 2.1 (default parameters) - EXPONENTIAL annealer

Implemented exp annealer from http://katrinaeg.com/simulated-annealing.html.

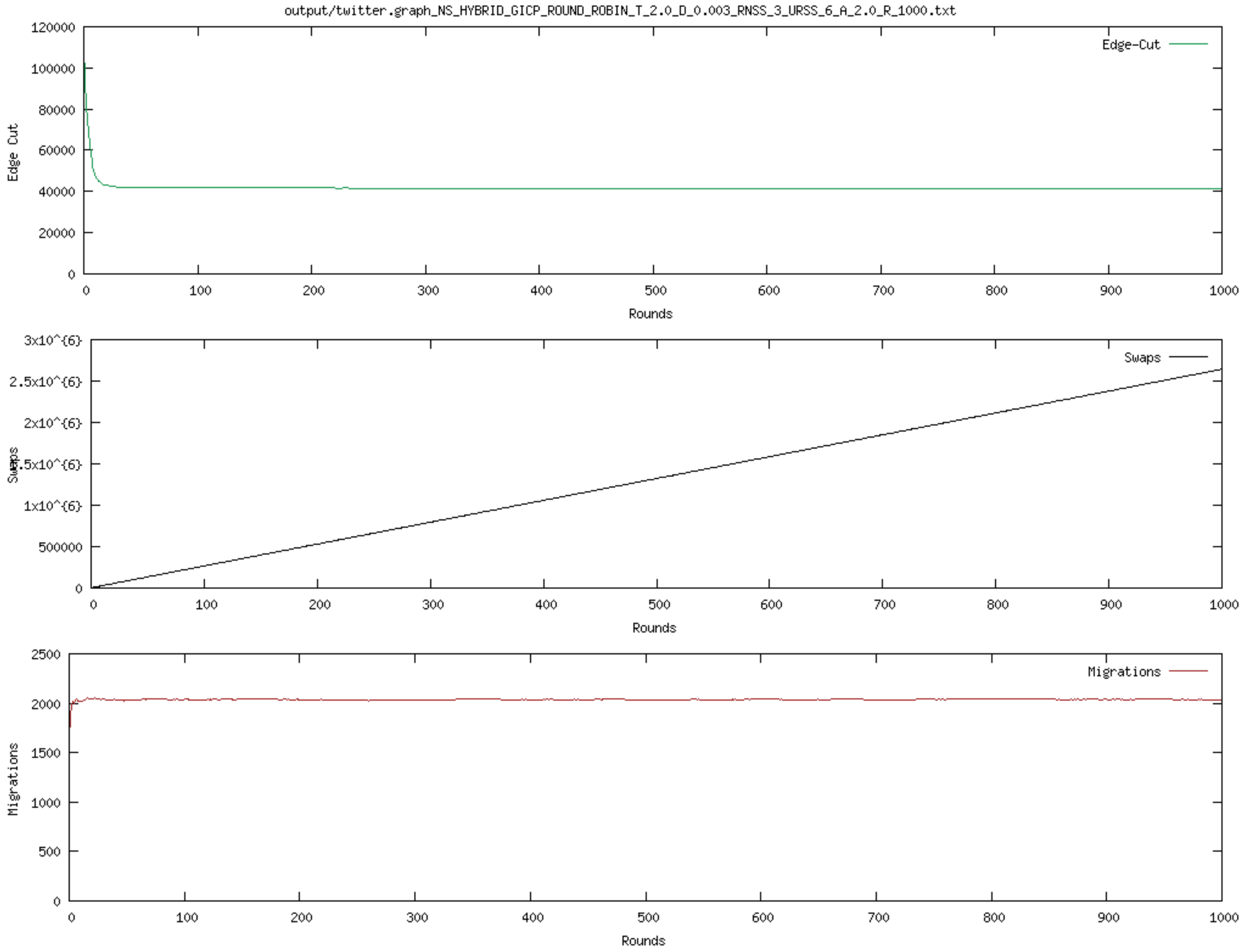
3elt ~ 9 seconds, edge_cut=2753



add20 ~ 6 seconds, edge_cut=2310



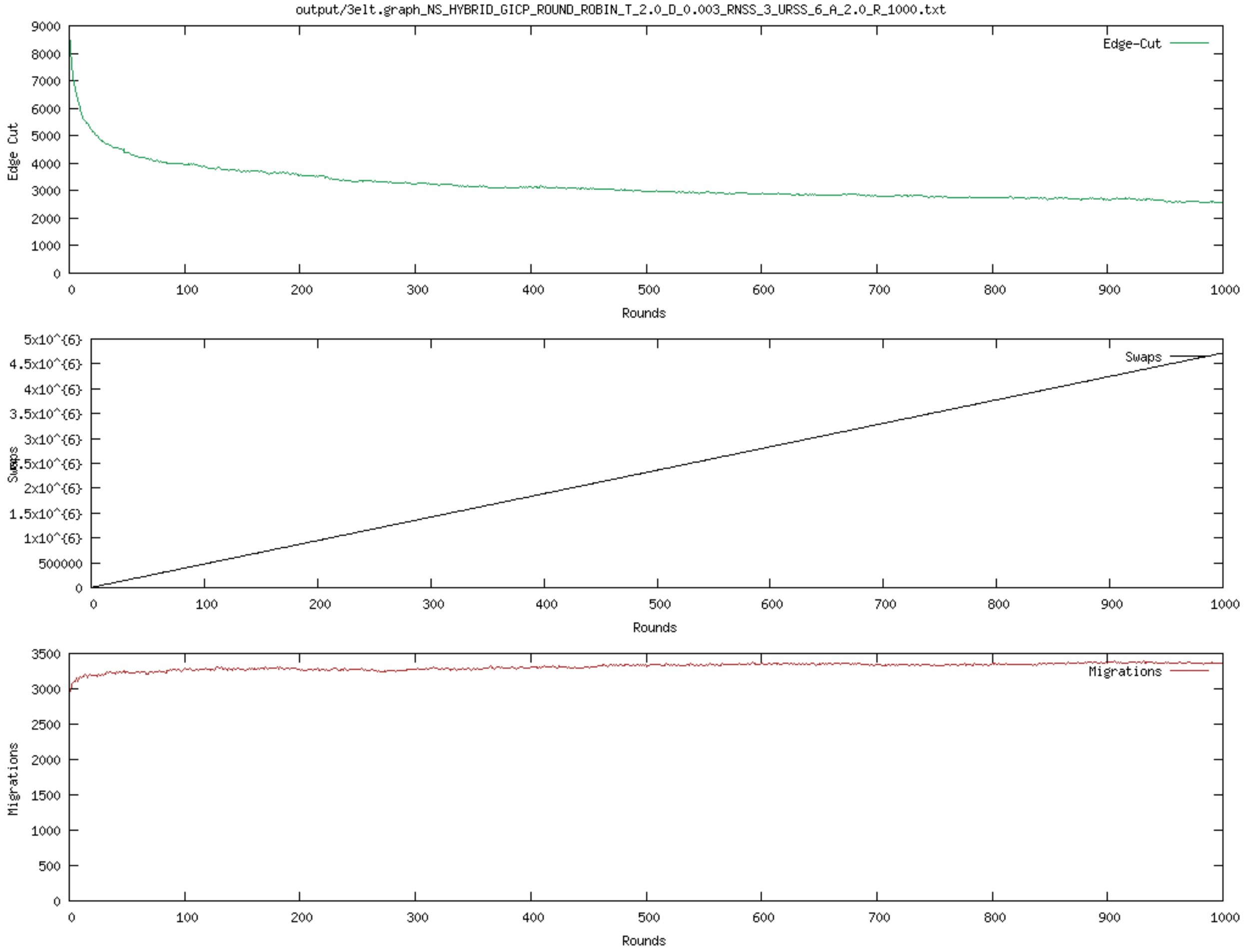
Twitter ~ 71 seconds, edge_cut=41627



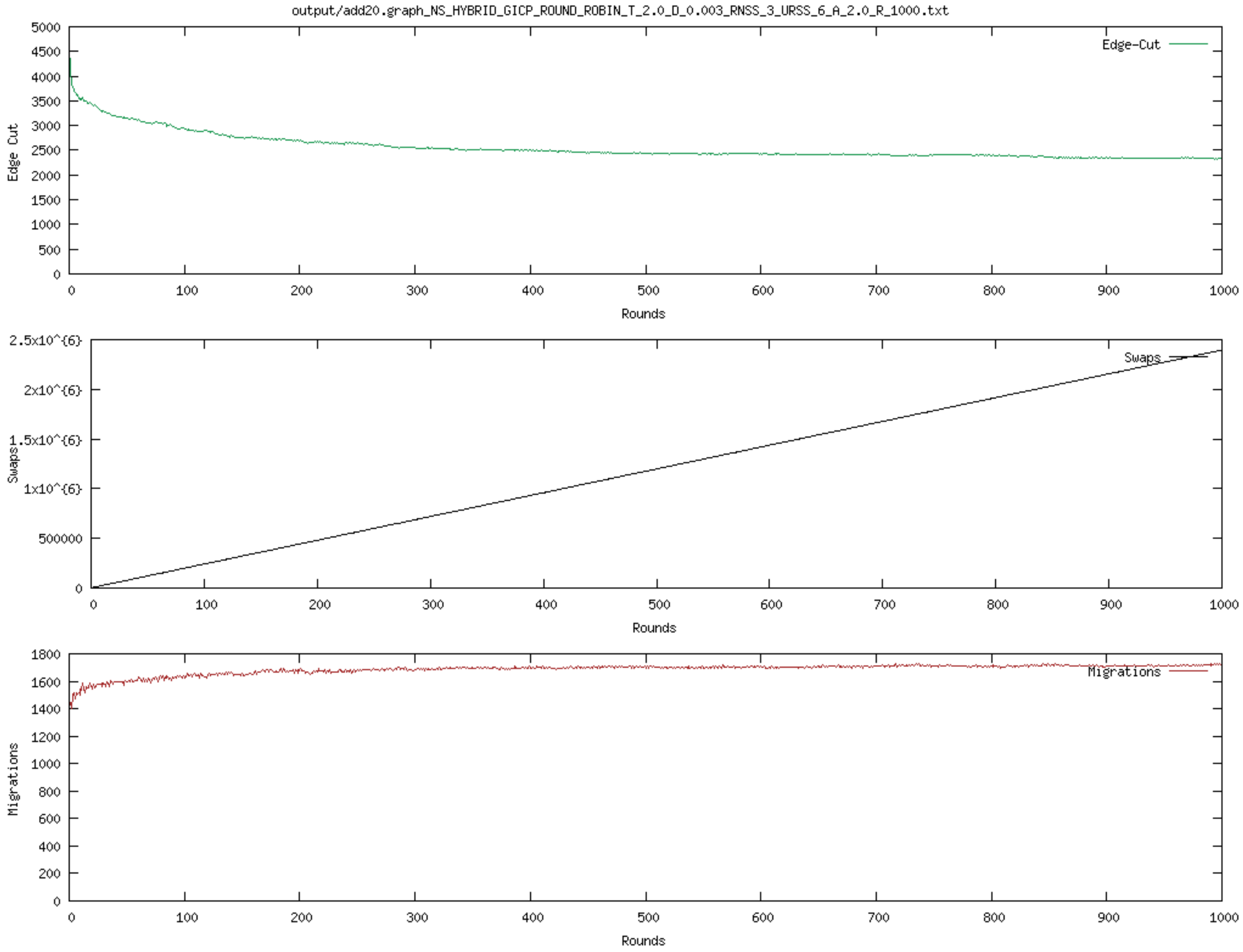
Task 2.2 (default parameters) - EXPONENTIAL annealer with restart=yes

We reset the temperature every T/delta rounds and wait for T/delta/2 rounds. This is to give the Jabeja algorithm time to converge before starting over again. Another possible solution is restarting immediately by resetting the temperature every T/delta/2 rounds.

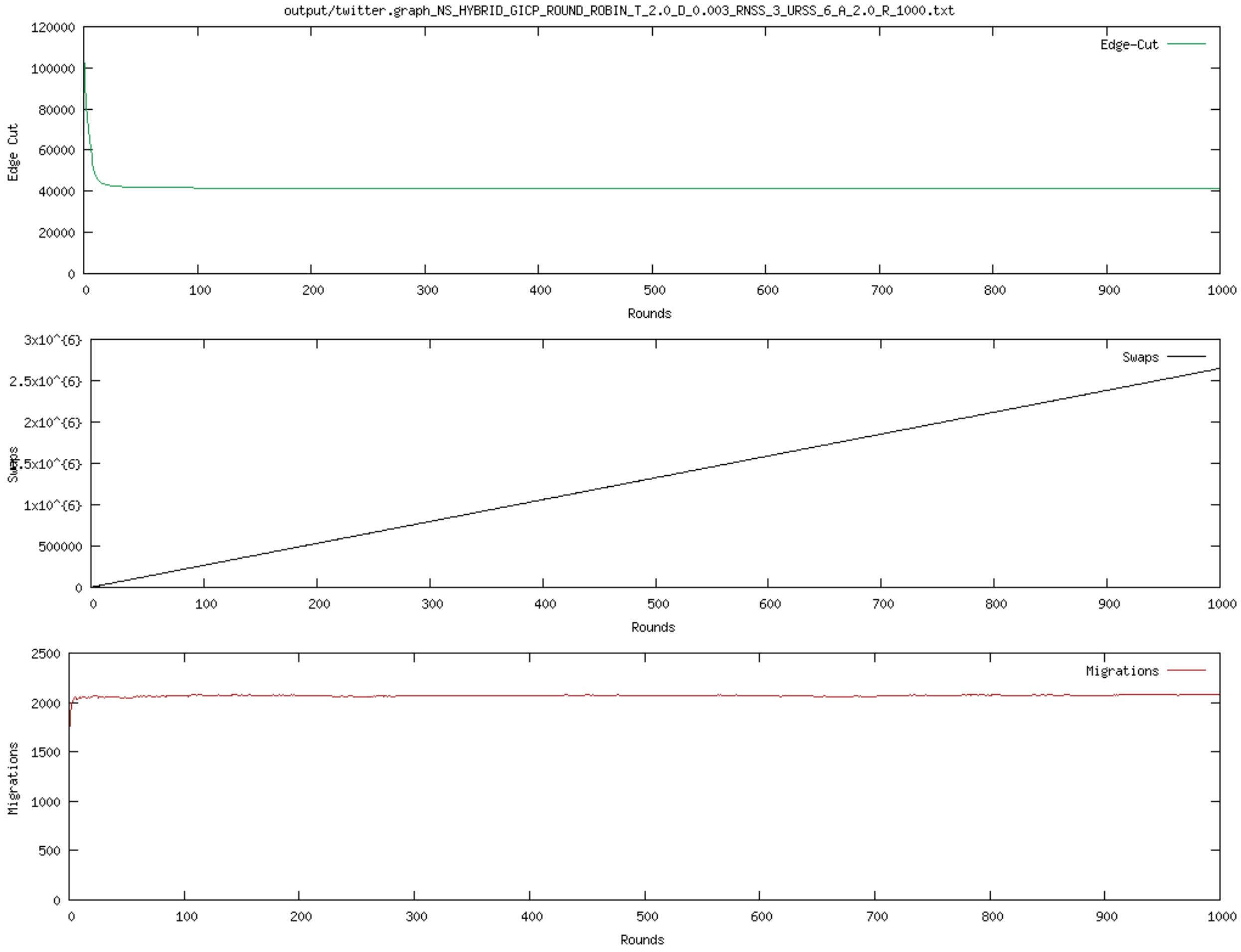
3elt ~ 9 seconds, edge_cut=2588



add20 ~ 6 seconds, edge_cut=2328



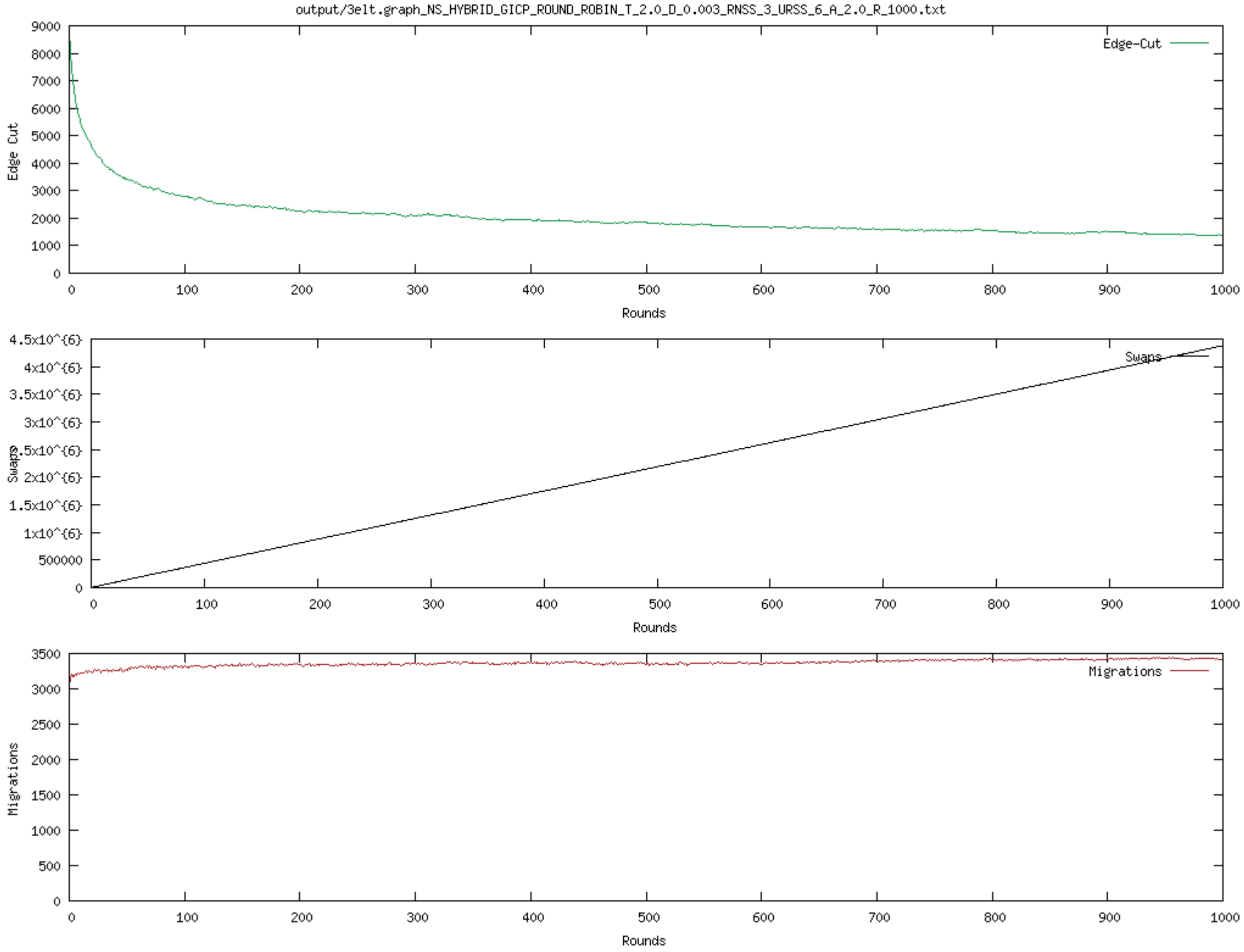
Twitter ~ 79 seconds, edge_cut=41516



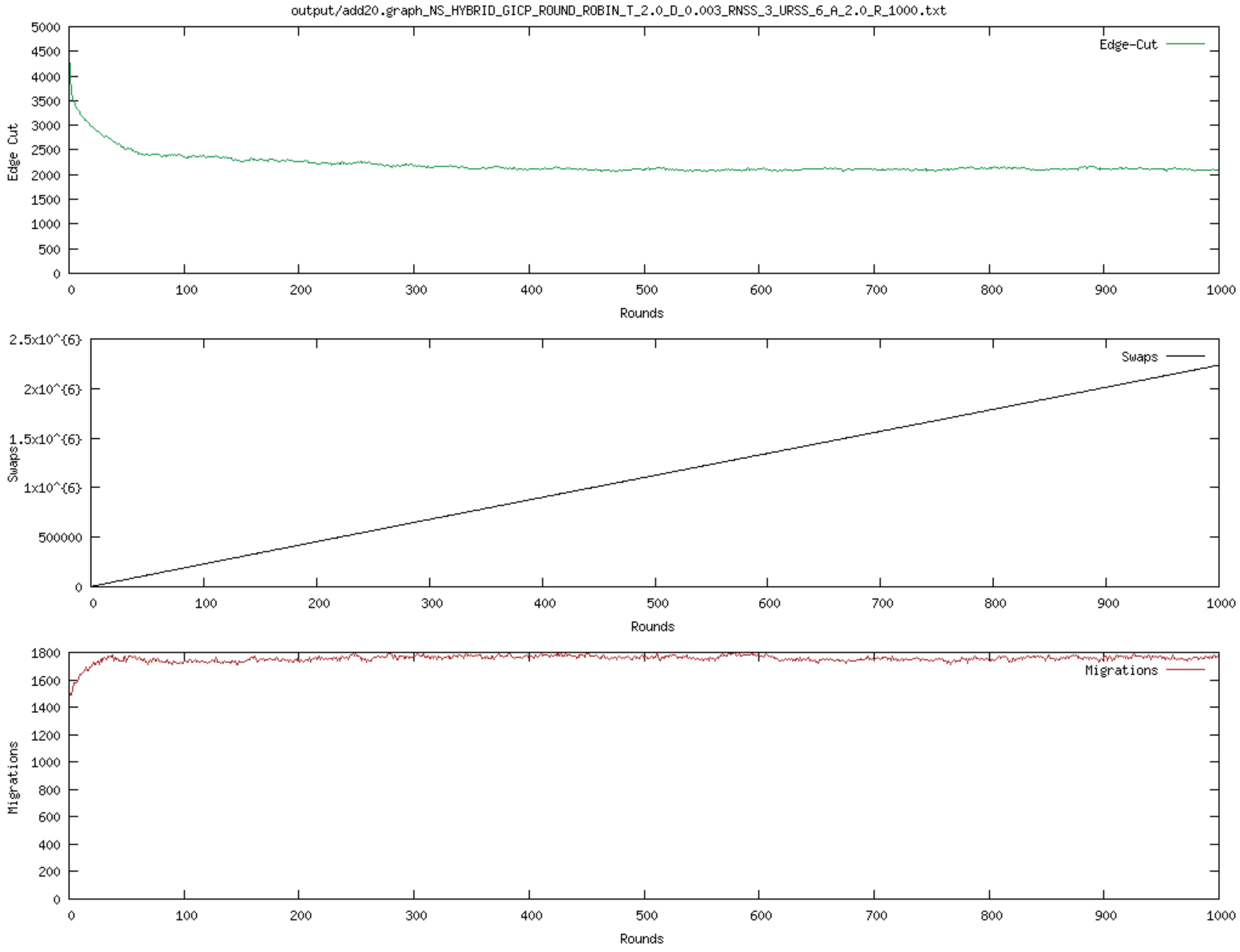
Optional task (default parameters) - CUSTOM (LOGISTIC) annealer

Formula: *1 + exp(-(newCost - oldCost) / temperature)*
This Formula is the logistic function, often used in logistic regression and neural networks to map any real-valued number into the range (0, 1), making it useful for interpreting the output as a probability.
Both the formulas used in the Exponential case and the one here return 0 when oldCost is way higher than newCost, and 1 when newCost is way higher than oldCost.
The difference is that the Logistic function is smoother when stepping from 0 to 1, thus it has a nicer behavior.

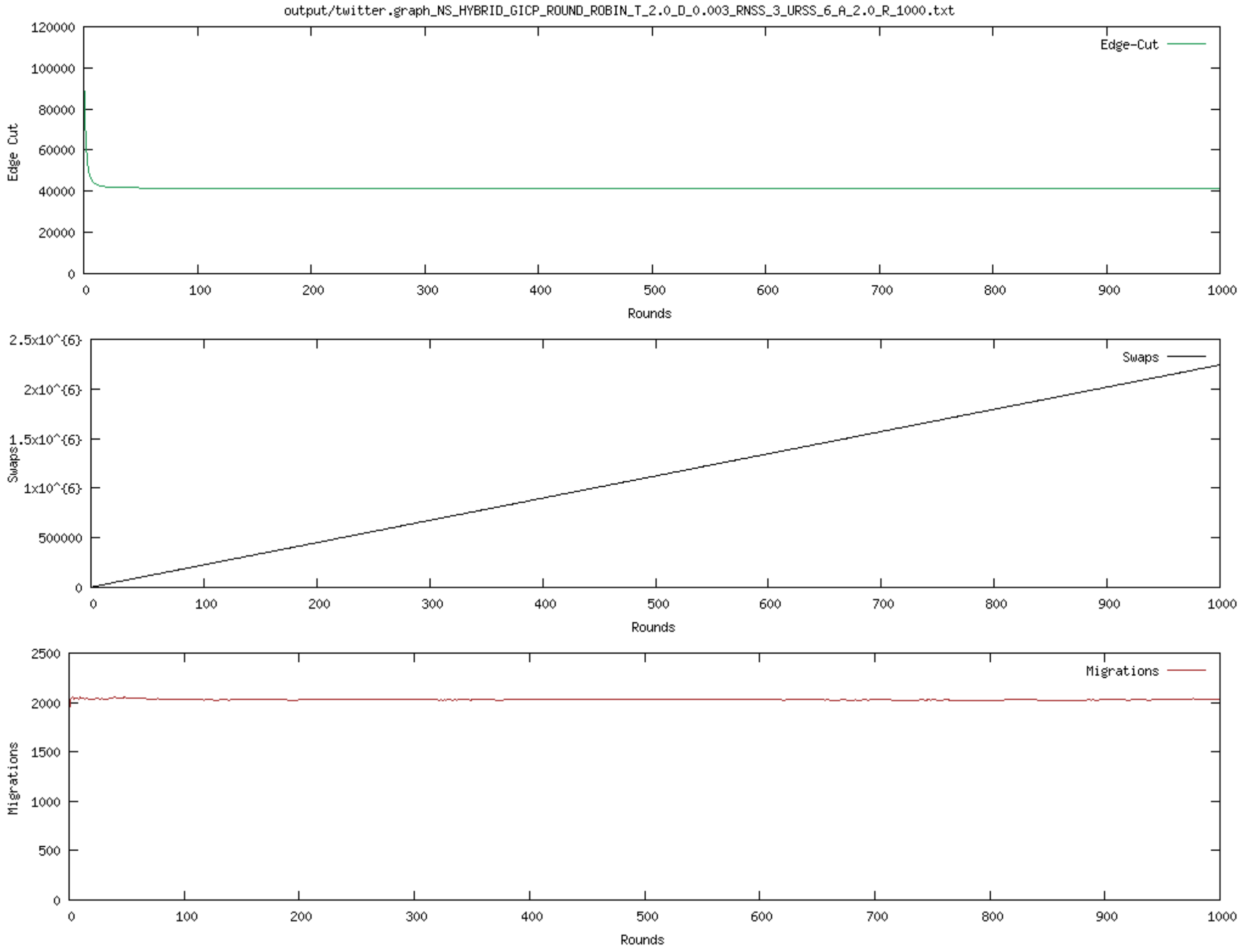
3elt ~ 13 seconds, edge_cut=1381



add20 ~ 8 seconds, edge_cut=2098



Twitter ~ 91 seconds, edge_cut=41462

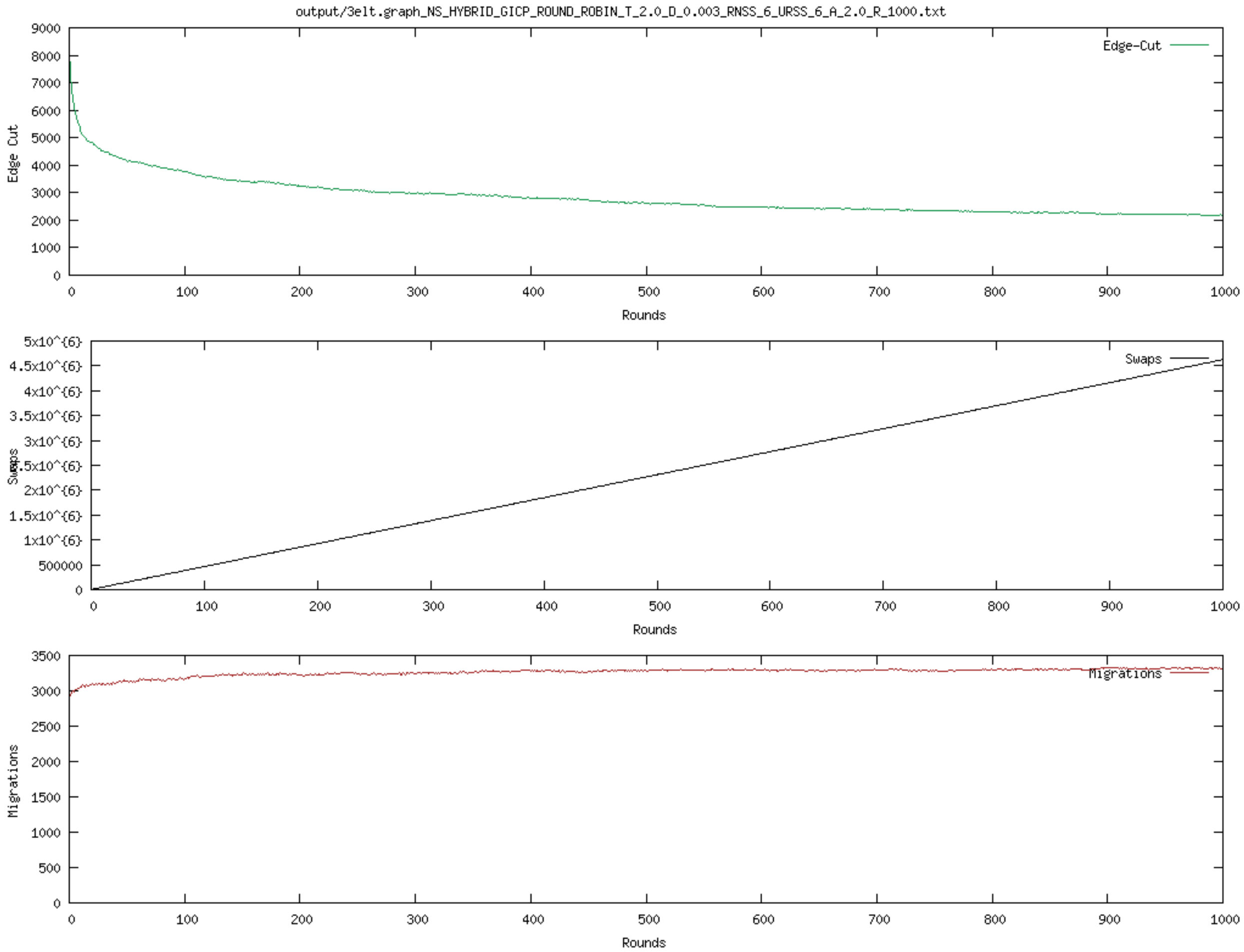


Playing with parameters - BEST annealer

```
./run.sh -graph ./graphs/add20.graph -randNeighborsSampleSize 6
```

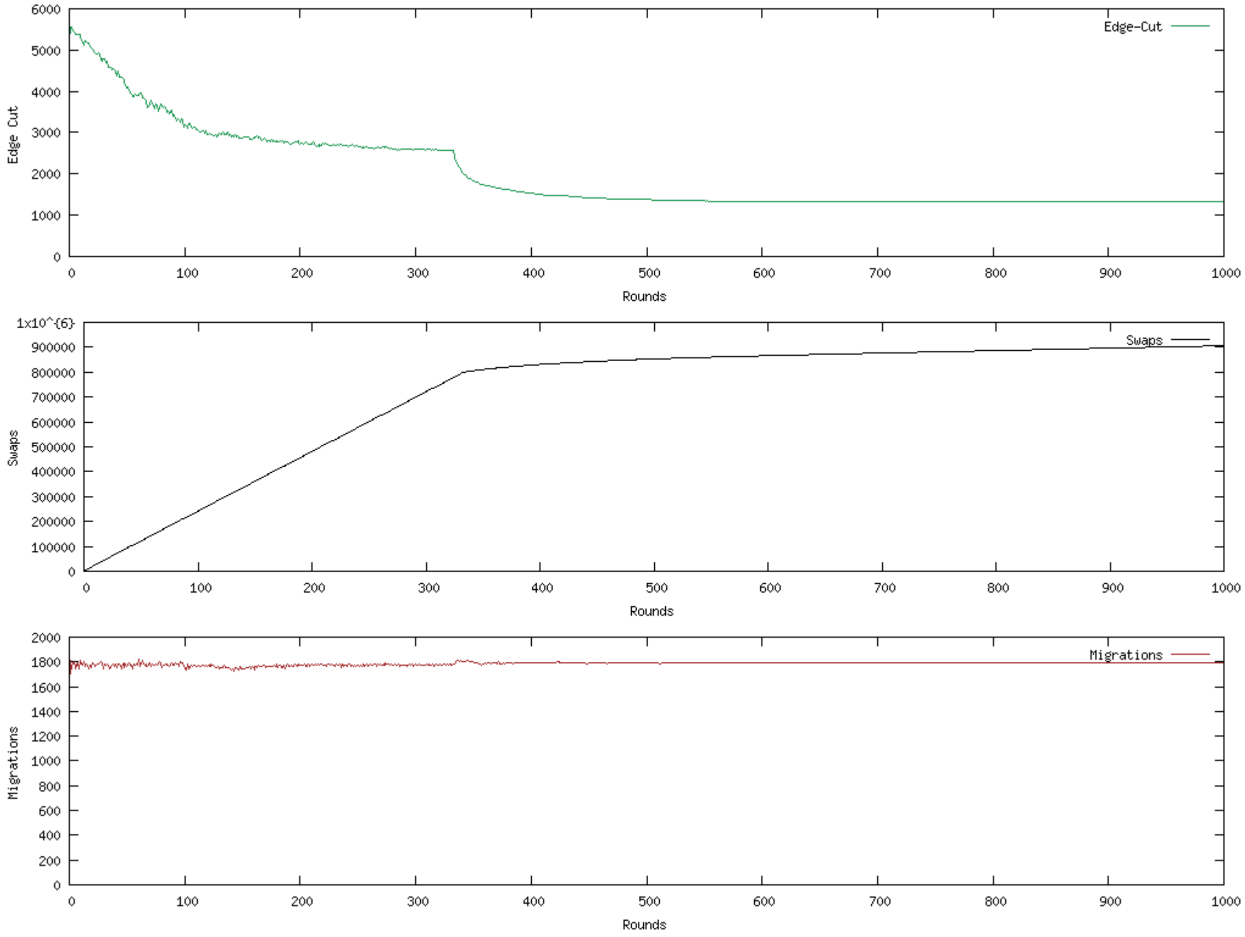
Increasing the randNeighborsSampleSize parameter yields a more precise edge_cut, because visiting more neighbors one can perform a more informed, thus better, swap.
It is slightly slower, as we need to check more neighbors for each swap.

3elt ~ 8 seconds, edge_cut=1320

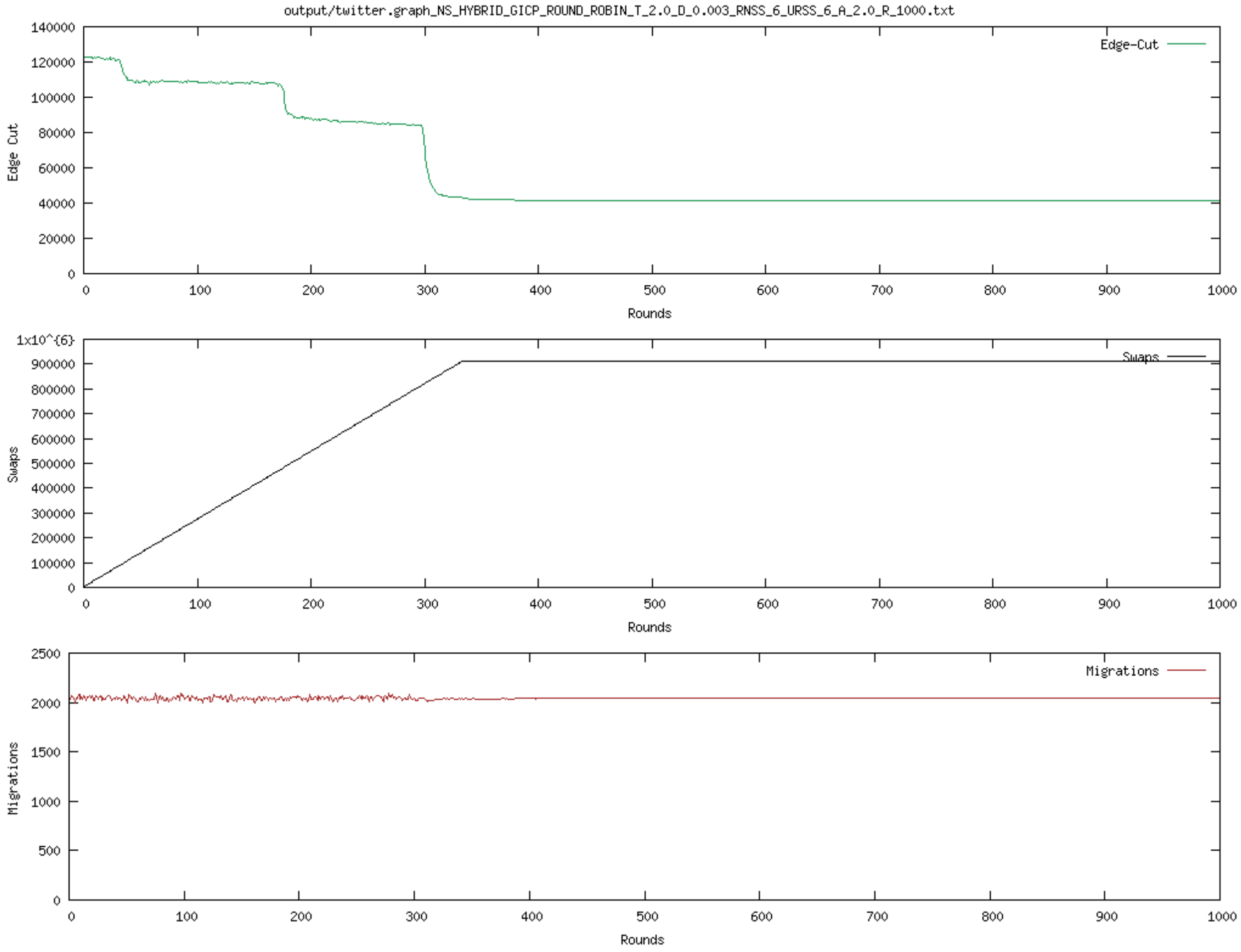


add20 ~ 17 seconds, edge_cut=2181

output/add20.graph_NS_HYBRID_GICP_ROUND_ROBIN_T_2.0_D_0.003_RNSS_6_URSS_6_A_2.0_R_1000.txt



Twitter ~ 160 seconds, edge_cut=41067



Summarizing table

dataset	Annealer	Restart	Performance (edge_cut)	Execution time (seconds)	-randNeighborsSampleSize param
3elt	LIN	no	2788	8	3
3elt	EXP	no	2753	9	3
3elt	EXP	yes	2588	9	3
3elt	CUSTOM	no	**1381**	13	3
3elt	CUSTOM	no	2181	17	6
add20	LIN	no	1797	6	3
add20	EXP	no	2310	6	3
add20	EXP	yes	2328	6	3
add20	CUSTOM	no	2098	8	3
add20	LIN	no	**1320**	8	6
Twitter	LIN	no	**40964**	116	3
Twitter	EXP	no	41627	71	3
Twitter	EXP	yes	41516	79	3
Twitter	CUSTOM	no	41462	91	3
Twitter	LIN	no	41067	160	6

Considerations

The table above analyzes the 3elt, add20, Twitter graphs. It discusses how our changes affect the performance of the algorithm in terms of:
a) Time to converge;
b) Minimum edge cut observed.
The Number of swaps are observable in the relative graphs.

For 3elt the best configuration for the minimum edge cut is CUSTOM annealear with no restart and randNeighborsSampleSize as default (3).
For add20 the best configuration for the minimum edge cut is LINEAR annealear with no restart and randNeighborsSampleSize as 6.
For Twitter the best configuration for the minimum edge cut is LINEAR annealear with no restart and randNeighborsSampleSize as default (3).

Surprisingly, the original configuration with a linear annealer, the one that is also mentioned in the paper, performed better than the others, except for the 3elt graph, where the logistic annealer definitely improved the performances. Furthermore, the linear annealer tweaked with a different value for the randNeighborsSampleSize parameter, further improved the performances.
The exponential annealer with restart set as true performs better than the same but with restart set as false.
Finally, the custom annealer outperforms the exponential one, meaning we further improved the performances by changing the acceptance probability function.