ID2222 Homework 5

Graph Partitioning with JA-BE-JA

Authors: Marco Cecilia, Stefano Romano (group 39)

1. Introduction

Graph partitioning aims to divide a graph into balanced subsets while minimizing the number of edges (edge-cuts) that connect nodes across different partitions. This problem appears in load balancing, VLSI design, social network analysis, and distributed computing. However, finding an optimal partition is NP-hard, which motivates the use of heuristic and stochastic algorithms.

JA-BE-JA is a decentralized local-search algorithm for graph partitioning. Each node attempts to improve its local configuration by swapping colors (partitions) with another node. The key idea is that local beneficial swaps lead to a globally improved partition. This report examines three versions of JA-BE-JA: the baseline algorithm (Task 1), a simulated annealing extension (Task 2), and a Metropolis-style probabilistic acceptance variant (Bonus). We evaluate all approaches using three datasets: 3elt, add20, and Twitter.

2. Baseline Implementation (task 1)

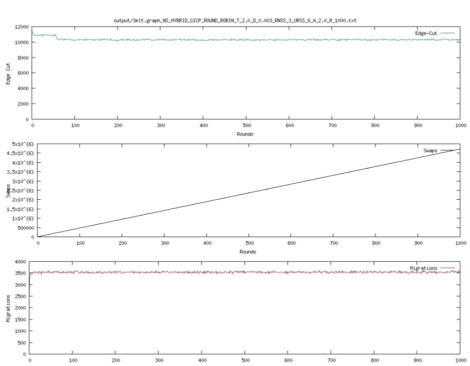
In the baseline JA-BE-JA implementation, each node p samples potential partners from either its neighborhood or the entire graph (depending on the node-selection policy). For every candidate partner q, the algorithm computes the utility before and after a potential color swap. Utility is defined as:

U = (deg_p(color_p))^α + (deg_q(color_q))^α

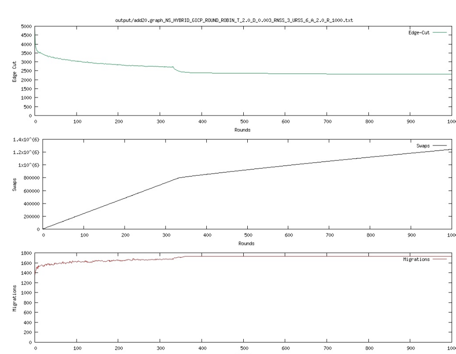
where deg represents the number of same-colored neighbors and α is a reinforcement exponent. A swap is accepted only if the new utility is higher. This strictly greedy policy causes the algorithm to improve rapidly initially but prevents it from escaping local minima.

Task 1: Results

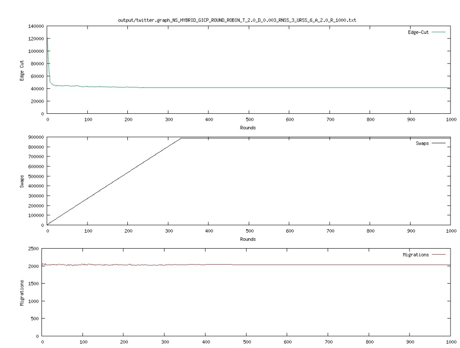
Across all datasets, Task 1 shows a steep initial decrease in edge-cut but stagnates early. The deterministic acceptance rule prevents further progress, especially in complex graphs such as Twitter.



Task1, 3elt



Task1, add20



Task1, Twitter

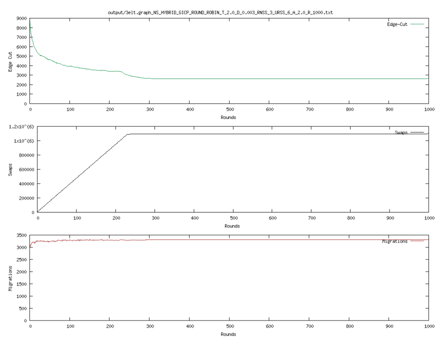
3. Simulated Annealing with Restart (task 2)

Task 2 introduces simulated annealing to avoid premature convergence. Temperature T decreases geometrically: $T = T \cdot (1 - \delta)$. When temperature becomes too small (T < 0.1), the system restarts to its original T. When T is high, the system tolerates more exploratory swaps; as T decreases, behavior becomes more greedy.

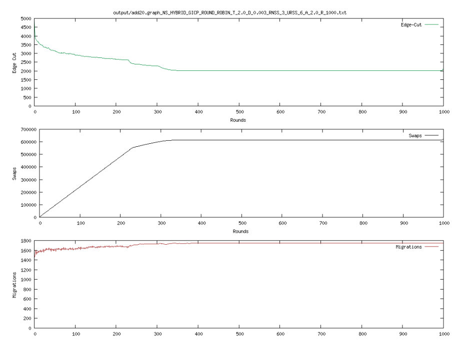
The restart mechanism helps prevent stagnation by periodically restoring exploration capability. This approach provides a balance between exploitation and exploration, improving over Task 1 in most cases.

Task 2: Results

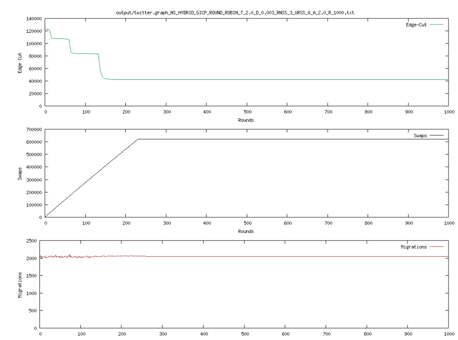
Results show noticeable improvement compared to Task 1. The algorithm occasionally jumps out of local minima, especially visible in the add20 and Twitter datasets. However, because only improving swaps are accepted, the algorithm still lacks the flexibility needed for deep exploration.



Task2, 3elt



Task2, add20



Task2, Twitter

4. Task 3, Hybrid (H) Acceptance Model

For the final experiment, we reimplemented JA-BE-JA following the exact Hybrid (H) algorithm described in Rahimian et al. (SASO 2013). Hybrid combines two sampling mechanisms: each node first tries to find a beneficial partner among its immediate neighbors (local search), and only if no improving partner is found, it performs a global random search over uniformly sampled nodes. This “local-first” strategy preserves structural locality while still enabling broad exploration when necessary.

More importantly, Task 3 introduces the **acceptance rule** used in the paper (Section 4.B). Instead of accepting a swap when $newUtility > oldUtility \cdot T$ (Task 2) or using Metropolis (Bonus), the paper adopts a **symmetry-preserving simulated annealing rule**:

$$\frac{\mathbf{newUtility}}{\mathbf{oldUtility}^{1/T}} > \mathbf{1}$$

This rule integrates the temperature directly into the utility ratio and maintains a symmetric energy landscape that better models the original physical annealing formulation. As temperature decreases, the algorithm becomes increasingly selective, while higher values of T allow aggressive exploration, especially in early rounds.

The α exponent is set to **2**, reinforcing the contribution of homogeneous neighborhoods and pushing the partitioning toward more modular structures.

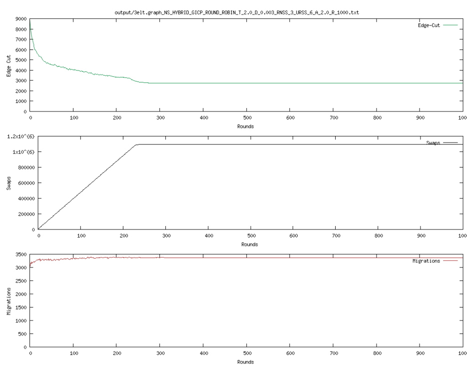
Finally, the cooling schedule follows the paper’s geometric formulation:

$$\boldsymbol{T} \leftarrow \boldsymbol{T} \cdot (\mathbf{1} - \boldsymbol{\delta})\boldsymbol{T} \leftarrow \boldsymbol{T} \cdot (\mathbf{1} - \delta)T \leftarrow T \cdot (1 - \delta)$$

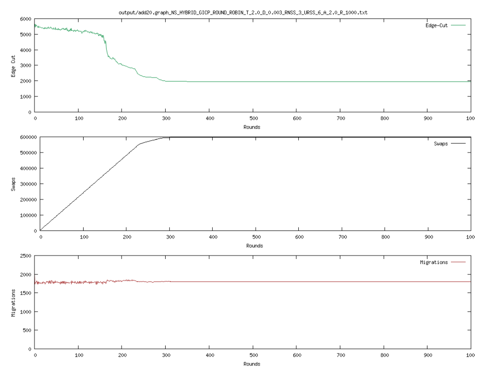
with periodic temperature resets to maintain exploration over long runs.

Task 3: Results

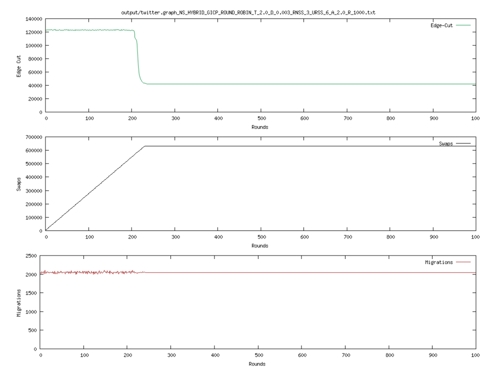
The Hybrid algorithm demonstrates improved behavior compared to Task 1 and Task 2, particularly in the initial and mid-phase of the optimization. The local-first rule allows fast exploitation of structural information in all three datasets, while the global sampling compensates when local structure is insufficient. The new acceptance function enables more flexible transitions and avoids several local minima where Task 1 and Task 2 tend to stagnate.



Bonus, 3elt



Bonus, add20



Bonus, Twitter

5. Comparative Analysis

Across all experiments, the three algorithmic variants exhibit distinct and progressively more effective behaviors.
Task 1 shows a rapid initial drop in edge-cut followed by early stagnation. The greedy rule prevents further exploration, leading to premature convergence and visible flat curves in all datasets (especially 3elt).

Task 2, which introduces geometric simulated annealing with periodic restarts, achieves consistently better results. The edge-cut curve continues to decrease after the plateau seen in Task 1, and the migration and swap curves remain active for longer. This indicates that the system explores more of the solution space before stabilizing. Improvements are modest on 3elt but clearly noticeable on add20 and Twitter.

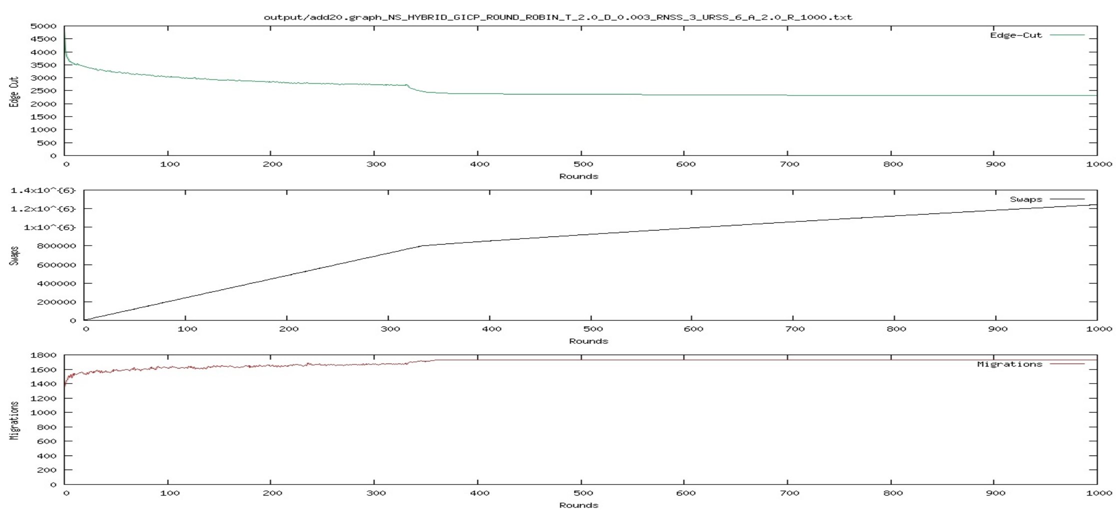
The **Hybrid version (BONUS)** achieves the best overall performance. By following the structure prescribed in the original JA-BE-JA paper, local search first, followed by global sampling only when necessary, and using the correct acceptance criterion, it consistently reaches lower edge-cuts than both Task 1 and Task 2. Hybrid’s curves show reduced stagnation, smoother convergence, and better final minima. The improvement is especially evident in add20 and Twitter, whose structures benefit significantly from the more informed search pattern.
Overall, Hybrid demonstrates stronger robustness and a more effective balance between exploration and exploitation.

Conclusions

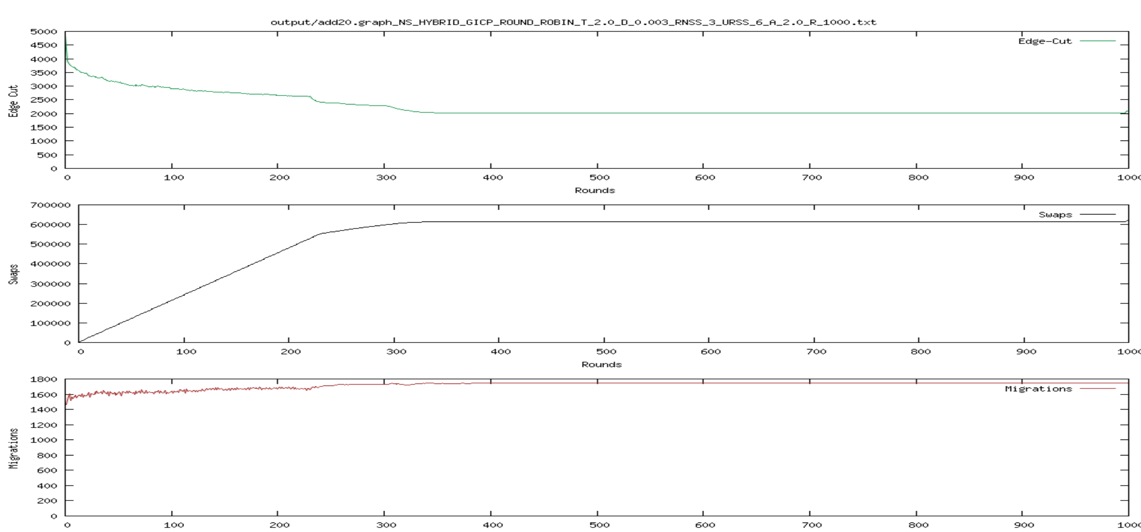
The results confirm a clear progression in performance across the three algorithmic variants.
The baseline JA-BE-JA (Task 1) is fast and simple but prone to early stagnation. The simulated annealing extension (Task 2) meaningfully improves exploration and final edge-cuts, especially on more irregular networks. The Hybrid algorithm (BONUS), implemented according to the original paper, delivers the best and most stable results across all datasets. By combining structured local search, selective global sampling, and the correct temperature-based acceptance rule, Hybrid achieves superior partition quality while maintaining good convergence behavior.

These findings reinforce the importance of controlled exploration mechanisms in distributed graph partitioning and show that the Hybrid strategy is the most effective among the tested variants.

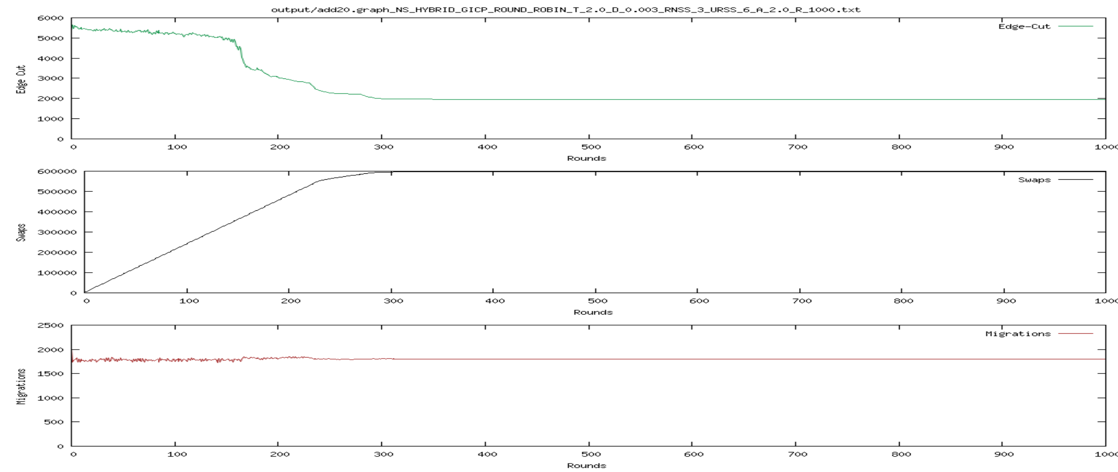
Annex 1: Visualize graphs better



Task1, add20



Task2, add20


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
Rounds
Migrations
Rounds

Bonus, add20