Efficient Stochastic Local Search for Modularity Maximization

Rafael Santiago
Universidade do Vale do Itajaí
Itajaí, Brazil
rsantiago@univali.br

Luís C. Lamb
Federal University of Rio Grande do Sul
Porto Alegre, Brazil
lamb@inf.ufrgs.br

ABSTRACT

In this paper, we analyze stochastic local searches and neighborhood strategies for the Modularity Maximization problem. Modularity Maximization was shown relevant in the identification of clusters in complex and social networks. Our experimental analysis shows that *1-neighborhood* is a better strategy for the problem as it quickly locates suboptimal partitions. We find 99% near best-known partitions with 94% of frequency in time $|V|^{1.44}$ and 95% near best known Q values in sublinear time.

Keywords

Clustering, Modularity Maximization, Stochastic Local Search

1. INTRODUCTION

Modularity Maximization is a graph clustering problem defined in [3], where an objective function measures the difference between the internal edges of each cluster and the expected number of connections. The expected number of connections is the probability of nodes of the same cluster to be connected with an edge.

For a given graph $G = (V, E)$, where V is the set of nodes, E is the set of edges, d_i is the degree of node $i \in V$, Equation 1 describes the value of the objective function of Modularity Maximization for a partition s, where a_{ij} is 1 when nodes i and j are connected by an edge, and 0 otherwise. The partition s is a set of disjoint clusters.

$$Q(s) = \frac{1}{2|E|} \sum_{c \in s} \sum_{i,j \in c} \left(a_{ij} - \frac{d_i d_j}{2|E|} \right) \quad (1)$$

The aim of this paper is to show that how neighborhood strategy can be used to improve solutions generated by Modularity Maximization heuristics. We identified which combination between Stochastic Local Search (SLS) and neighborhood strategy groups is better for Modularity Maximization. In doing so, we experimentally show that *Tabu search* using the *1-neighborhood* strategy finds better partitions than the combinations among *Randomized Iterative Improvement*, *Simmulated Annealing*, and *Tabu search* methods, when using the *merge*, *split*, and *merge+split* neighborhood strategies.

GECCO '16 July 20-24, 2016, Denver, CO, USA

ACM ISBN 978-1-4503-4323-7/16/07.

DOI: http://dx.doi.org/10.1145/2908961.2909003

2. STOCHASTIC LOCAL SEARCHES

SLS are local searches that have some random behavior such that each execution performs an exploration in a different area of the search space. They use an initial solution to improve it over successive iterations. At each iteration, a solution with a small modification (determined by a neighborhood strategy) is selected by priority criteria.

In this paper, we described results for three SLS and four neighborhood strategies for Modularity Maximization, where a solution is a partition of disjoint clusters. The first SLS reported is the Randomized Iterative Improvement (RII), which selects a random neighbor solution with probability α and the best neighbor with $1 - \alpha$ [1]. The second SLS is the Simulated Annealing (SA), in which a random neighborhood s' from the current solution s is selected at each iteration, and replaces s if s' is better than s; otherwise, s' replaces s with probability $e^{-\frac{\Delta(s,s')}{kT}}$, where k is the Boltzmann constant (frequently $k = 1$) and T is the temperature. The heuristic starts with a high temperature that is decreased at each iteration. In the first iterations, a random walk is performed, and only the best solutions are selected to replace s at the last iteration [2]. Finally, our last SLS is Tabu Search (TS) which has a short memory structure: the feature f of the selected solution s' is inserted into a tabu list for a particular time window at each iteration.

In our experiments, we use the following neighborhood strategies: *1-neighborhood*: given a partition s, s' is a neighbor of s if and only if, there are clusters C' and C'' in s, where, if we change a node from C' to C'' then $s = s'$. In our experiments, the change of a node to another cluster C'' occurs if, and only if, this node has an edge to another node in C''; *Merge*: s' is neighbor of s if, and only, all its clusters are the same, except for cluster $C \in s'$ and both C' and $C'' \in s$, where $C' \cup C'' = C$; and *Split*: s' is neighbor of s if, and only if, there is a cluster C in s that is split by half in s'; *Merge+split*: all neighbors are defined as in strategies *merge* and *split*.

3. RESULTS AND CONCLUSIONS

The experiments used an Intel Core i3 64 bits with 3.10GHz with 3072KB of cache memory and 4GB of RAM over Linux Ubuntu 12.04.3 LTS operating system. The language used was Python, with "Pypy" processor.

Figures 1 (a), (b), and (c) show the Q value results of algorithms RII, TS and SA, respectively. In each figure, we show the average approximation to the best known Q value by parameter and neighborhood. Each vertical bar represents the average over all

Table 1: Parameter setting analysis for near best solutions. Number of visited partitions needed, frequency of achievement, and time complexity to reach approximation of 90%, 95%, 99% of the best known modularity value. C_1=(RII, *1-neighborhood*, 0.5), C_2=(RII, *merge*, 0.1), C_3=(RII ,*merge+split*, 0.1); C_4=(SA, *1-neighborhood*, 0.99), C_5=(TS, *1-neighborhood*, R), and C_6=(TS, *merge+split*, 2.0).

	#visited solutions			%frequency			k for $\|V\|^k$		
C_i*	0.9	0.95	0.99	0.9	0.95	0.99	0.9	0.95	0.99
C_1	122.2±9.4	224.1±23.4	1064.5±320.7	100.0	100.0	79.3	1.03±0.63	1.29±0.51	2.41±0.94
C_2	100.0±0.0	100.0±0.0	none	95.3	54.0	0.0	1.79±1.88	5.35±9.25	none
C_3	108.7±16.9	none	none	42.7	0.0	0.0	0.58±0.53	none	none
C_4	2368.5±126.7	3435.3±551.6	4419.3±1424.1	99.3	90.7	16.0	0.1±0.3	0.18±0.31	0.54±0.59
C_5	100.0±0.0	100.0±0.0	492.4±138.6	100.0	100.0	94.7	1.29±0.22	1.29±0.22	1.44±0.77
C_6	166.7±268.8	none	none	44.7	0.0	0.0	none	none	none

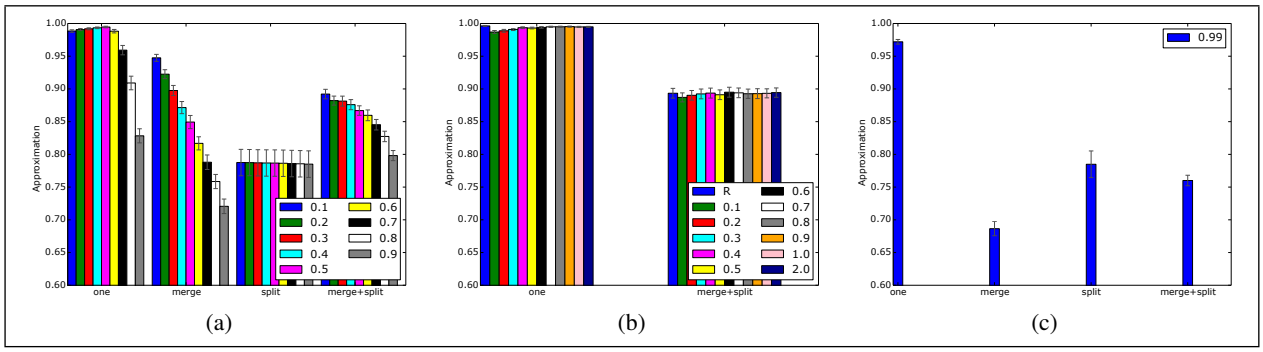

Figure 1: Average approximation obtained by tested neighborhoods and parameters of the RII (a), TS (b), and SA (c) search.

best results from classical instances for 30 trials for a specific SLS, neighborhood, and parameter.

The results are executed for $10,000$ iterations for RII and TS, and SA started with the temperature equal to $10,000$. The most frequent, highest Q values were achieved by TS, with *1-neighborhood* and $\alpha = R$ (random parameter between 0.1 and 0.9).

Table 1 describes the best parameters for the best combinations of SLS heuristics and neighborhood strategies (in column C_i*). The results of RII with *split*, SA with *merge*, *split* and *merge+split* are omitted because no run achieved the approximation. The column "#visited partitions" shows the average number of partitions visited in each combination to achieve a near best known Q value. The "%frequency" reports the percentage that achieved an approximate partition over all tests. The column "k for $|V|^k$" shows the constant exponent over the polynomial hypothesis and the error. Thus, to achieve a result close to the best-known solution for TS using *1-neighborhood* the empirical complexity is approximate $|V|^{1.44}$ operations with error ± 0.77. The empirical complexity is calculated using polynomial hypothesis in linear regression $\log(time) \sim \log(|V|)$ with the R statistical computing system (http://www.r-project.org/).

In our experiments, finding partitions with 0.99 evaluation function near the best-known partition was possible with *1-neighborhood*. TS with *1-neighborhood* strategy resulted in the most frequent near best solutions: for $\sim 94\%$ of samples, using a number of operations $\sim |V|^{1.44}$. The number of visited partitions increases to find the best-known partitions. TS also obtained the smallest number of visits. We can note that to find partitions, at least, 90% near to best known, the *1-neighborhood* is the best choice over all other tested neighborhoods.

Acknowledgement

This work is partly supported by the Brazilian Research Council CNPq and the *Universidade do Vale do Itajaí*.

4. REFERENCES

[1] H. Hoos and T. Stützle. *Stochastic Local Search: foundations and applications*. Morgan Kaufmann, 2004.

[2] S. Kirkpatrick, C. Gelatt Jr, and M. Vecchi. Optimality conditions and exact neighborhoods for sequencing problems. *Science*, 220(4598):671–680, 1983.

[3] M. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, Feb. 2004.