One-mode projection-based multilevel approach for community detection in bipartite networks

Alan Valejo, Vinícius Ferreira, Geraldo P. R. Filho
Maria C. F. de Oliveira and Alneu A. Lopes

Institute of Mathematical and Computer Sciences (ICMC), University of São Paulo (USP)
P.O. Box 668, 14560-970, São Carlos, SP, Brazil

{alanvalejo,viniciusferreira97,geraldoprfilho}@gmail.com
{alneu,cristina}@icmc.usp.br

Abstract

Interest in algorithms for community detection in networked systems has increased over the last decade, mostly motivated by a search for scalable solutions capable of handling large-scale networks. Multilevel approaches provide a potential solution to scalability, as they reduce the cost of a community detection algorithm by applying it to a coarsened version of the original network. The small-scale solution thus obtained is then projected back to the original large-scale model to obtain the desired solution. However, standard multilevel methods are not directly applicable to bipartite network models and the literature lacks studies on multilevel optimization applied to such networks. This article addresses this gap and introduces a novel multilevel method based on one-mode projection that allows executing traditional multilevel methods in bipartite network models. The approach has been validated with an algorithm that solves the Barber's modularity problem. It attained improved runtime performance, whilst solution accuracy is shown to be statistically equivalent to that of the standard method.

1 Introduction

Complex networks are relational structures that represent many real-world systems composed by a large number of highly interconnected dynamical units. Many such systems exhibit a natural bipartite (or two-layer) structure, in which the set of units (known as vertices) is split into two disjoint subsets (layers) and connections (known as edges) are established between units placed in different layers. Document-word (Rossi et al., 2016), protein-ligand (Jeong et al., 2000) and actor-movie (Watts and Strogatz, 1998) networks are a few examples of real-world bipartite networks.

Community structures, defined as groups of vertices densely connected to each other within a group, but sparsely connected to other groups, are an important and frequent property of many such networks. Vertices that belong to the same community usually share common properties and play similar roles in a network system. Therefore, the identification of a community structure in networked systems contributes to a better understanding of their topological structure and dynamical processes (Fortunato, 2010). For instance, in biological domains, communities in a protein network typically correspond to proteins that share a single specific function (Mahmoud et al., 2014). Furthermore, the increasing interest in identifying community structures in bipartite networks (Dormann and Strauss, 2013; Thébault, 2013; Larremore et al., 2014; Dormann and Strauss, 2014; Alzahrani and Horadam, 2015; Beckett, 2016) is a strong indicator that is a promising research topic.

Community detection algorithms aim at subdividing a set of vertices into k communities for minimizing the number of edges connecting vertices placed in different communities. This is a hard combinatorial optimization problem, in which the goal is to optimize a given cost function, such as modularity (Girvan and Newman, 2002). As the number of possible network states can be exponential, it becomes unfeasible to search for an optimal solution on large-scale networks.

To overcome this problem, researchers have resorted to multilevel approaches, in which: i. an original network is continuously reduced through a collapsing of vertices and edges (coarsening phase); ii. an initial community structure is obtained on the coarsest network (solution phase); and iii. the initial solution is successively projected back over the inverse sequence of coarsened networks, until the original network (projection and refinement phase).

Many multilevel community detection algorithms have been developed for handling unipartite or one-mode networks. Some studies introduced multilevel community detection methods for specific types of networks; for instance, Abou-Rjeili and Karypis (Abou-Rjeili and Karypis, 2006) considered networks that exhibit a power-law degree distribution and Valejo et al. (Valejo et al., 2014c,b,a) explored properties of social networks, as high transitivity and assortativity. Other contributions focused on the application of multilevel optimization for improving the modularity measure (Djidjev, 2008; Schuetz and Caflisch, 2008; Ye et al., 2008; Noack and Rotta, 2009; Rotta and Noack, 2011; Djidjev and Onus, 2013; Lasalle and Karypis, 2015). Furthermore, many authors investigated parallel paradigms to improve the performance of coarsening and refinement phases (Baños et al., 2004; Banos et al., 2004; Trifunovic and Knottenbelt, 2004b,a; Erciye et al., 2005; Schweitz and Agrawal, 2007; Walshaw and Cross, 2007; LaSalle and Karypis, 2013; Lasalle and Karypis, 2015).

However, the above-mentioned approaches are not directly applicable to bipartite networks, since standard coarsening methods rely on collapsing pairs of connected vertices, assuming that all vertices are of the same type. In bipartite networks, vertices in different layers are not connected and should not be collapsed. Coarsening bipartite networks requires collapsing pairs that belong to the same layer (represent entities of the same type) and are not connected by edges.

This article addresses this gap and introduces a novel one-mode projection-based multilevel method that enables applying standard coarsening algorithms to bipartite networks. Tests conducted on a large set of synthetic network models have shown that it can be combined with a community detection method, yielding good speedup with no significant loss in solution quality.

The remainder of the paper is organized as follows: Section 2 reviews some basic concepts on networks and provides a brief overview of standard multilevel approaches; Section 3 introduces the proposed multilevel formulation for bipartite networks and its implementation; Section 4 presents results from an empirical study on a large synthetic test suite; finally, Section 5 summarizes the results and discusses potential applications and future work.

2 Fundamentals

This section describes the terminology and fundamental concepts required to understand the proposed solution.

2.1 Basic definitions

A unipartite network is given by $G(V, E, \omega)$, where $V = \{v_1, v_2, ..., v_n\}$ is the set of vertices, $E = \{e_1, e_2, ..., e_k\}$ is the set of edges connecting vertices, such that $e_i = (v, u) = \{(u, v) = (v, u) \mid u, v \in V\}$ and $\omega = \{w_1, w_2, ..., w_k\}$ is the set of weights, so that each $w_i \in \mathbb{R}$ is associated with a corresponding edge e_i. Two vertices are said to be neighbors if they are connected by at least one edge.

A bipartite network is given by $G(V, E, \omega)$, V is partitioned into two sets V_1 and V_2 so that $V_1 \cap V_2 = \emptyset$, $V_1 = \{u_1, u_2, \dots, u_n\}$ is a set (or layer) of vertices, $V_2 = \{v_1, v_2, \dots, v_m\}$ is another set of vertices and $E = \{e_1, e_2, ..., e_k\}$ is the set of edges connecting vertices from different layers, i.e. for all $(u, v) \in E$, $u \in V_1$ and $v \in V_2$ and $E \subseteq V_1$ x V_2. Similarly, $\omega = \{w_1, w_2, ..., w_k\}$ is the set of edge weights. Figure 1 illustrates a bipartite network.

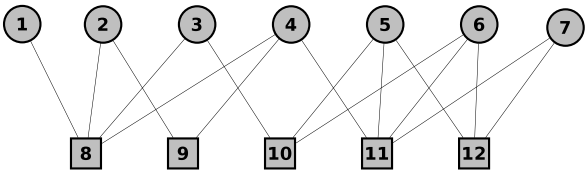
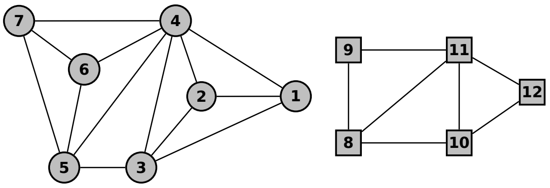


Figure 1: A bipartite network.

Bipartite networks can be transformed into unipartite networks through one-mode projections (Newman, 2001; Opsahl, 2010; Padrón et al., 2011). Application of a one-mode projection to a bipartite network generates two unipartite networks, one for each layer, G^1 and G^2, so that vertices with common neighbors are connected by edges in their respective projection. Figure 2 illustrates the result of applying a one-mode projection to the simple bipartite network shown in Figure 1. Figures 2(a) and 2(b) show, respectively, the one-mode projections of V (i.e., G^1) and U (i.e., G^2).

If two vertices share more than a single common neighbor, their connection in the unipartite projec-



(a) G^1 (b) G^2

Figure 2: Unipartite networks G^1 and G^2 resulting from a one-mode projection of the bipartite network in Figure 1.

tion should reflect this topology. In weighted unipartite projections, the number of common neighbors between two vertices is assigned as their edge weight, as illustrated in Figure 3 for a particular pair of projected vertices.

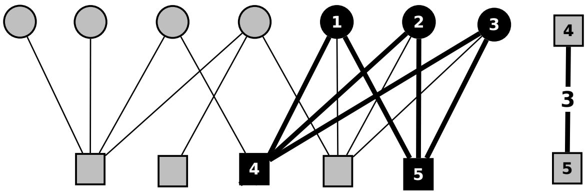


Figure 3: Example of a weighted projection: vertices 4 and 5 are connected by an edge of weight 3, as they share three common neighbors.

2.2 Multilevel optimization

Multilevel optimization reduces the number of operations required to solve a combinatorial problem. Common applications in the literature include partitioning and community detection algorithms in networks. The rationale behind the multilevel strategy is to execute a complex optimization algorithm, that can not be executed on a very large network, on a reduced version of this network which requires a much smaller number of operations. The results obtained in the smaller network are then projected back to get the solution relative to the original network.

Let us consider a unipartite network $G_0(V_0, E_0, \omega_0)$ and assume its size (in terms of edges and vertices) prevents the execution of a target algorithm. A multilevel approach could be applied as follows (Karypis and Kumar, 1998):

Coarsening phase. Network G_0 is transformed into a sequence of smaller networks $G_1, G_2, ..., G_m$. The size of the vertex set is reduced in each subsequent network, i.e., $|V_0| > |V_1| > |V_2| > ... > |V_m|$.

Initial solution phase. The target algorithm is applied to network G_m. As $|V_m|$ is sufficiently small, the target algorithm can be run in feasible time. In the present study, the target is a community detection algorithm.

Uncoarsening phase. The solution obtained in the coarsest network G_m is projected back, through the intermediate levels $G_{m-1}, G_{m-2}, \cdots, G_1$, until it is obtained in the network G_0.

The coarsening phase is an iterative process that constructs a sequence of reduced versions of the initial network G_0. The vertices of a network G_i are collapsed into super-vertices to obtain a network G_{i+1}. Edges incident to the original vertices are joined to obtain the edges incident to a super-vertex. The coarsening process is split into two phases, namely matching and coarsening.

In the matching phase, edges, or vertex pairs, are selected to collapse. Once an edge in G_i has been collapsed, its incident vertices are joined into a super-vertex. Any vertex from G_i with no incident edge selected is inherited by G_{i+1}. In the present study, we employed two matching methods introduced by Karypis and Kumar (1998), namely:

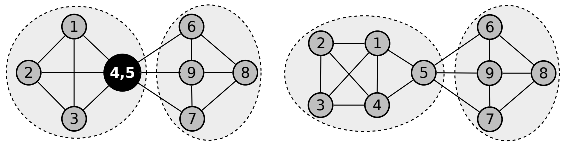
Random Matching (RM). In this approach vertices are visited in a random order. If a vertex v has not been matched yet, one of its unmatched neighbors is selected. If such a vertex u exists, the pair (v, u) is included in the matching set, otherwise v remains unmatched. Although it may yield poor results, RM has complexity $O(|E|)$.

Heavy edge matching (HEM). This approach minimizes the edge-cut by selecting a maximal matching formed by the edges with heavier weights. Similarly to RM, vertices are also visited in random order. However, unlike RM, vertices v and u are matched if edge (v, u) has maximum weight over all valid edges incident to v. Although HEM does not guarantee that the matching obtained has maximum weight, it yields better results than RM with equivalent asymptotic complexity.

Next, the coarsening phase starts and a coarser network G_{i+1} can be created directly from the matching by joining each pair of matched vertices into a single *super-vertex* (sV). Edges incident to sV, called super-edges, are obtained by joining the edges incident to vertices $\{u, v\} \in V_i$. The weight of the resulting super-edge is given by the sum of the weights of all edges incident to $\{u, v\} \in V_i$.

The target algorithm (community detection, in our case) is then evaluated in the coarsest network G_m to obtain an initial solution. As $|V_M| < |V_0|$, the algorithm converges faster and generates an initial solution in feasible time.

In uncoarsening phase, the initial solution is successively projected back to G_0. At each level, each super-vertex $s_v = \{u, v\} \in V_{i+1}$ is expanded to its original vertices in V_i, i.e. u and v, and the solution is projected through the intermediate levels $G_{m-1}, G_{m-2}, ..., G_0$. For each decomposed $s_v \in V_{i+}$, its original vertices $\{u, v\} \in V_i$ are assigned to the same community of their parent $s_v \in G_i$. Figure 4 illustrates this process: super-vertex $s_v = \{4, 5\}$ (Figure 4(a)) is expanded to its original vertices 4 and 5, which are assigned to the same community of s_v.



(a) (b)

Figure 4: Super-vertex s_v formed by vertices 4 and 5 is expanded and both vertices are assigned to the same community of s_v.

3 Proposal

This section introduces a multilevel community detection method that handles bipartite networks. Standard methods do not consider vertices of different types, whereas in bipartite networks, layers usually represent different types of entities that should be handled independently. Therefore, typical coarsening methods, such as RM or HEM, are not directly applicable. Nonetheless, they can be applied to a projection of G, $P = G_V, G_U$, since in a one-mode projection all vertices are of the same type. Weighted one-mode projection methods enable applying any standard coarsening algorithm to bipartite networks after a transformation process. We rely on this concept to introduce a multilevel community detection method applicable to bipartite networks.

Algorithm 3.1 summarizes the implementation of the proposed ***o**ne-mode **p**rojection-based **m**ultilevel community detection* (OPM). It comprises the phases of coarsening (lines 1-6), community detection (line 7) and uncoarsening (lines 8-10). The inputs are the initial bipartite network $G = (V, E, \sigma, \omega)$, a maximal number of levels $L = \{L_i \mid L_i \in [0, n] \subset \mathbb{Z}\} \therefore |L| = 2$ and a reduction factor for each layer $rf = \{rf_i \mid rf_i \in (0, 0.5] \subset \mathbb{R}\} \therefore |rf| = 2$.

The bipartite network initially undergoes a one-mode projection transformation, being split into two unipartite networks G^1 and G^2. The coarsening process is then applied to each unipartite network (line 3), level by level, until each one has been reduced by the desired factor. The process comprises a matching step (line 4) and a coarsening step (line 5). In this study, we have adopted the aforementioned coarsening and matching methods HEM and RM (Karypis and Kumar, 1998).

An initial community structure S_l is then obtained on the coarsest bipartite network G_l, at level l (line 7). As G_l and S_l are, respectively, the input and output (S_l representing the community structure of network G_l), different algorithms for community detection can be considered. Depending on the settings of the coarsening phase, the coarsest bipartite network can be very small, so that computationally expensive algorithms can be employed with limited impact on overall performance. Finally, in the subsequent uncoarsening phase (lines 8-10), solution S_l is projected back to G_0 through the space of intermediate solutions $S_{l-1}, S_{l-2}, ..., S_1, S_0$ (line 9). Following the guidelines proposed by Karypis and Kumar (1998) for the uncoarsening process, solution S_l is constructed from S_{l+1} simply by assigning vertices $\{u, v\} \in V_l$ to the same community of their parent super-vertex $sV \in V_{l+1}$.

4 Experimental Results and Analysis

In order to evaluate the proposed solution, we implemented OPM and investigated whether it could yield solutions of quality statistically equivalent to that of a standard community detection approach, whilst increasing its scalability to larger networks.

Beckett (Beckett, 2016) recently introduced the $LPAwb+$ algorithm, which maximizes Barber's modularity through label propagation in weighted bipartite networks and has competitive perfor-

Algorithm 3.1: OPM: One-mode projection-based multilevel community detection

```
Input:
    bipartite network                    : $\mathsf{G} = (\mathsf{V}, \mathsf{E}, \sigma, \omega)$
    maximal number of levels             : array $\mathsf{L} = \{L_i \mid L_i \in [0, n] \subset \mathbb{Z}\} \therefore |\mathsf{L}| = 2$
    reduction factor for each layer      : array $\mathsf{rf} = \{rf_i \mid rf_i \in (0, 0.5] \subset \mathbb{R}\} \therefore |\mathsf{rf}| = 2$
Output:
    solution $\mathsf{S}$

for $i \in \{1, 2\}$ do
    while ($l \leqslant \mathsf{L}_i$) or (layer is as small as desired) do
        $\mathsf{G}_l^i \leftarrow$ projection($\mathsf{G}_l$, $i$);
        $\mathsf{M} \leftarrow$ matching($\mathsf{G}_l^i$, $\mathsf{rf}_i$);
        $\mathsf{G}_{l+1}^i \leftarrow$ coarsening($\mathsf{G}_l^i$, $\mathsf{M}$);
        increase $l$;
$\mathsf{S}_l \leftarrow$ community detection in $\mathsf{G}_l$;
while $l \neq 0$ do
    $\mathsf{S}_{l-1} \leftarrow$ uncoarsening($\mathsf{G}_{l-1}$, $\mathsf{G}_l$, $\mathsf{S}_l$);
    decrease $l$;
Return: $\mathsf{S}$
```

mance relative to the state-of-the-art methods for community detection. However, it is a computationally costly algorithm prohibitive for large-scale networks.

We employed our proposed framework to create a multilevel implementation of $LPAwb+$ (from now on identified as the OPM algorithm) that adopts HEM or RM as the coarsening methods, i.e. OPM_{hem} and OPM_{rm}, respectively. Both were executed with parameters $rf = 0.5$ and $L = [1, 2, 3]$ in a set of 15 synthetic weighted bipartite networks, identified as R1-R15. The synthetic networks were obtained by a community model described by (Beckett, 2016) that creates networks with unbalanced and randomly positioned community structures and different community sizes. We generated networks of sizes $n = |V_1 + V_2|$ within the range $[1,000, 15,000]$ at increments of $1,000$, with the number of communities set to $0.01 * n$. Edge weights were randomly assigned from a skewed negative binomial distribution and noise was introduced in the connection patterns by rewiring a percentage of edges between and within the communities.

The performance was measured with the normalized mutual information (NMI), which compares a solution found by a particular algorithm with a reference solution (Labatut, 2013), and the execution times were also measured. Experiments were conducted in a 8-core Linux machine with 3.7 GHz of CPU and 64 GB RAM. The algorithm was implemented in Python with *igraph* library[1]. We report average values obtained from 30 executions for algorithms that rely on random strategies.

Table 1 shows the accuracy values measured by NMI on the 15 synthetic networks. The highest values are shown in bold and values equal to or higher than those of the baseline solution are highlighted with a gray background. The best performances were achieved by OPM_{hem} with one level of coarsening ($L = 1$) on 11 out of the 15 networks. The baseline community detection algorithm $LPAwb+$ yielded the best performance in 3 networks, whereas the worst results were obtained with OPM_{rm} for ($L = 1$). In one of on the 15 synthetic networks, OPM_{hem} and $LPAwb+$ were equivalent.

Indeed, the random strategy RM yields very poor accuracies, which renders its application unfeasible in real contexts. However, the greedy strategy HEM yielded accuracy values similar to those of $LPAwb+$. Furthermore, limited coarsening levels (mainly $L = 1$) yielded higher accuracy values, whereas accuracy decreases as the coarsening level ($L = 3$) increases. For $L = 3$ the extensive collapsing of vertices tends to blur the boundaries between adjacent communities. The effect of parameter L depends on network size, i.e.

[1] available from http://igraph.org/python/

Algorithm		Dataset														
Name	Levels[L]	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15
$LPAwb+$	0	0.918	0.926	0.983	0.972	0.964	0.99	0.984	0.999	0.999	0.985	0.989	0.996	0.995	0.987	0.992
OPM_{hem}	1	**0.991**	**0.987**	**0.985**	**0.984**	**0.991**	0.99	**0.985**	0.992	0.991	**0.995**	**0.99**	**0.991**	0.992	**0.991**	**0.993**
OPM_{hem}	2	0.973	0.985	0.981	0.982	0.989	0.987	0.989	0.989	0.988	0.989	0.989	0.988	0.989	0.990	0.991
OPM_{hem}	3	0.873	0.952	0.960	0.966	0.963	0.971	0.972	0.975	0.972	0.975	0.973	0.974	0.975	0.976	0.976
OPM_{rm}	1	0.312	0.358	0.409	0.412	0.407	0.414	0.442	0.462	0.448	0.462	0.468	0.483	0.483	0.502	0.498
OPM_{rm}	2	0.146	0.169	0.147	0.135	0.158	0.171	0.157	0.162	0.150	0.150	0.150	0.161	0.148	0.152	0.161
OPM_{rm}	3	0.1	0.119	0.119	0.098	0.105	0.105	0.090	0.079	0.082	0.069	0.078	0.099	0.084	0.082	0.072

Table 1: NMI accuracy values of the OPM variants and baseline $LPAwb+$ in 15 synthetic networks.

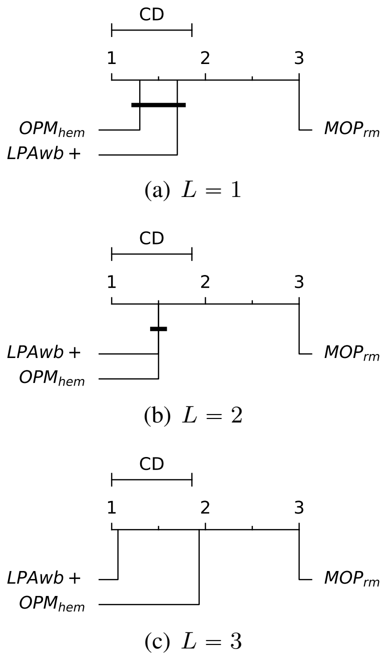
Algorithm		Dataset															sum
Name	Levels[L]	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	
$LPAwb+$	0	14	96	308	904	2782	2800	7146	5925	9197	56119	66729	75990	97392	224032	302442	851876
OPM_{hem}	1	3	14	43	127	326	540	973	1933	2989	6983	9628	13695	18473	22875	36439	115041
OPM_{hem}	2	2	6	21	51	106	139	296	396	854	2215	3112	3776	5442	6826	10538	33780
OPM_{hem}	3	**1**	**6**	**13**	**33**	60	**55**	80	128	264	818	1604	1889	2351	2863	3769	13934
OPM_{rm}	1	8	57	124	220	419	476	780	1484	1991	5660	9619	13127	11291	17691	19202	82149
OPM_{rm}	2	4	25	40	58	103	150	146	179	345	1005	1626	1873	1902	1950	2041	11447
OPM_{rm}	3	3	16	28	**33**	**49**	67	**64**	**83**	**194**	**712**	**754**	**849**	**870**	**890**	**903**	**5515**
sum		35	220	577	1426	3845	4227	9485	10128	15834	73512	93072	111199	137721	277127	375334	1113742

Table 2: Absolute runtime (seconds) of OPM implementations and baseline $LPAwb+$ in each network.

differences in algorithm accuracy are likely to decrease as network sizes increase, which suggests that higher values of L might be adopted when handling larger networks.

A Nemenyi post-hoc test (Demšar, 2006) was applied to the results in Table 1 to detect statistical differences in the performances of the different algorithms. The results are shown in Figure 5 for (a) $L = 1$, (b) $L = 2$ and (c) $L = 3$. The critical difference (CD) is indicated at the top of each diagram and the methods' average ranks are placed on the horizontal axes (better ranked on the left). A black line connecting algorithms indicates no significant difference has been detected between them. The critical value of F-statistics with 2 and 28 degrees of freedom and at 90% is 2.50 for all diagrams.

When $L = 1$ (Figure 5(a)), OPM_{hem} was ranked best, followed by $LPAwb+$ and OPM_{rm}. Furthermore, no statistically significant difference was observed between OPM_{hem} and $LPAwb+$. For $L = 2$ (Figure 5(b)), OPM_{hem} and $LPAwb+$ were ranked first, and no statistically significant difference was observed between them. Finally, for $L = 3$ (Figure 5(c)) $LPAwb+$ was ranked first, with a statistically significant difference between OPM_{hem} and OPM_{rm}.



(a) $L = 1$

(b) $L = 2$

(c) $L = 3$

Figure 5: Nemenyi post-hoc test applied to the results from $LPAwb+$ and two OPM variants.

The scalability of OPM was also assessed considering its performance on each individual network. Table 2 shows the absolute execution times (in seconds) of the algorithm in each network - values refer to average times relative to 30 executions.

The longest execution time of $LPAwb+$ was 302,442 seconds (time to process the largest network, R15) and the shortest was 14 seconds (time to process the smallest one, R1). The most expensive OPM_{hem} ($L = 1$) consumed 36,409 seconds in R15 and 3 seconds in R1. Therefore, OPM_{hem} run 8.3 to 4.6 times faster than $LAPwb+$, relative to their maximum and minimum execution times, respectively. The maximum and minimum times of the least expensive OPM_{rm} ($L = 3$) were 903 seconds and 3 seconds, respectively. Therefore, OPM_{rm} runs 335 to 14 times faster than $LAPwb+$.

5 Conclusions

This article has introduced an approach that enables using the multilevel paradigm for scaling a community detection algorithm to handle large-scale bipartite networks. While previous multilevel methods consider only unipartite networks, our one-mode projection-based multilevel method enables handling bipartite networks with standard coarsening algorithms.

Tests on a large suite of synthetic networks have shown that this solution yields results with accuracy comparable to that of standard methods, demanding considerably shorter execution times. We tested two popular matching strategies for coarsening, namely HEM and RM. RM yielded expressive speedups or even improved the asymptotic convergence, but with poor results regarding accuracy, which prevents its practical application. However, HEM achieved rather good approximation in terms of accuracy and acceptable speedups, e.g., execution times over 8 times shorter as compared to the standard method.

Some issues that deserve further investigation include: using refinement strategies in the uncoarsening process and parallel or distributed paradigms to further increase scalability, as well as further exploring how the choice of rf, the reduction factor parameter, impacts accuracy and speedups.

Acknowledgments

Author A. Valejo is supported by a scholarship from the Brazilian Federal Agency for Support and Evaluation of Graduate Education (CAPES). This work has been partially supported by the State of São Paulo Research Foundation (FAPESP) grants 15/14228-9 and 17/05838-3; and the Brazilian Federal Research Council (CNPq) grants 302645/2015-2 and 305696/2013-0.

References

Amine Abou-Rjeili and George Karypis. 2006. Multilevel algorithms for partitioning power-law graphs. In *20th International Parallel and Distributed Processing Symposium, IPDPS 2006*. pages 124–124.

Taher Alzahrani and K. J. Horadam. 2015. Community Detection in Bipartite Networks: Algorithms and Case studies. In *Complex Systems and Networks*, pages 25–50.

R. Baños, C. Gil, J. Ortega, and F. G. Montoya. 2004. A parallel multilevel metaheuristic for graph partitioning. *Journal of Heuristics* 10(3):315–336.

R. Banos, C. Gil, J. Ortega, and F. G. Montoya. 2004. Parallel heuristic search in multilevel graph partitioning. In *12th Euromicro Conference on Parallel, Distributed and Network-Based Processing, 2004. Proceedings.*. pages 88 – 95.

Stephen J. Beckett. 2016. Improved community detection in weighted bipartite networks. *Royal Society open science* 3(1):140536.

Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research* 7:1–30.

Hristo N. Djidjev. 2008. A Scalable Multilevel Algorithm for Graph Clustering and Community Structure Detection. In *Fourth International Workshop, WAW 2006*. pages 117–128.

Hristo N. Djidjev and Melih Onus. 2013. Scalable and accurate graph clustering and community structure detection. *IEEE Transactions on Parallel and Distributed Systems* 24(5):1022–1029.

C. f. Dormann and R Strauss. 2013. Detecting modules in quantitative bipartite networks: the QuaBiMo algorithm. *arXiv preprint* 1304.3218.

Carsten F. Dormann and Rouven Strauss. 2014. A method for detecting modules in quantitative bipartite networks. *Methods in Ecology and Evolution* 5(1):90–98.

Kayhan Erciye, Ali Alp, and Geoffrey Marshall. 2005. Serial and Parallel Multilevel Graph Partitioning Using Fixed Centers. In *31st Conference on Current Trends in Theory and Practice of Computer Science Liptovský Ján*. pages 127–136.

Santo Fortunato. 2010. Community detection in graphs. *Physics Reports* 486(3-5):75–174.

Michelle Girvan and M. E. J. Newman. 2002. Community structure in social and biological networks. In *Proceedings of the National Academy of Science of the United States of America*. volume 99, pages 7821–7826.

H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A. L. Barabasi. 2000. The large-scale organization of metabolic networks. *Nature* 407(6804):651–654.

George Karypis and Vipin Kumar. 1998. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. *SIAM Journal on Scientific Computing* 20(1):359–392.

V. Labatut. 2013. Generalized Measures for the Evaluation of Community Detection Methods. *CoRR* abs/1303.5:44.

Daniel B. Larremore, Aaron Clauset, and Abigail Z. Jacobs. 2014. Efficiently inferring community structure in bipartite networks. *CoRR* abs/1403.2933.

Dominique LaSalle and George Karypis. 2013. Multi-threaded graph partitioning. In *Proceedings - IEEE 27th International Parallel and Distributed Processing Symposium, IPDPS 2013*. pages 225–236.

Dominique Lasalle and George Karypis. 2015. Multi-threaded modularity based graph clustering using the multilevel paradigm. *Journal of Parallel and Distributed Computing* 76:66–80.

Hassan Mahmoud, Francesco Masulli, Stefano Rovetta, and Giuseppe Russo. 2014. *Community Detection in Protein-Protein Interaction Networks Using Spectral and Graph Approaches*, Springer International Publishing, pages 62–75.

M. E. J. Newman. 2001. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences of the United States of America* 98(2):404–409.

Andreas Noack and Randolf Rotta. 2009. Multi-level algorithms for modularity clustering. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 5526 LNCS:257–268.

Tore Opsahl. 2010. Triadic closure in two-mode networks: Redefining the global and local clustering coefficients .

Benigno Padrón, Manuel Nogales, and Anna Traveset. 2011. Alternative approaches of transforming bimodal into unimodal mutualistic networks. the usefulness of preserving weighted information. *Basic and Applied Ecology* 12(8):713 – 721.

Rafael Geraldeli Rossi, Alneu de Andrade Lopes, and Solange Oliveira Rezende. 2016. Optimization and label propagation in bipartite heterogeneous networks to improve transductive classification of texts. *Information Processing and Management* 52(2):217 – 257.

Randolf Rotta and Andreas Noack. 2011. Multilevel local search algorithms for modularity clustering. *Journal of Experimental Algorithmics* 16(2):2.1.

Philipp Schuetz and Amedeo Caflisch. 2008. Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics* 77(4):1–7.

E. A. Schweitz and D. P. Agrawal. 2007. A Parallelization Domain Oriented Multilevel Graph Partitioner. *IEEE Transactions on Computers* 51(12):1435 – 1441.

Elisa Thébault. 2013. Identifying compartments in presence-absence matrices and bipartite networks: Insights into modularity measures. *Journal of Biogeography* 40(4):759–768.

Aleksandar Trifunovic and William J. Knottenbelt. 2004a. A parallel algorithm for multilevel k-way hypergraph partitioning. In *Parallel and Distributed Computing*. IEEE, pages 114–121.

Aleksandar Trifunovic and William J. W. J. Knottenbelt. 2004b. Parkway 2.0: A parallel multilevel hypergraph partitioning tool. In *Computer and Information Sciences - ISCIS 2004*, pages 789–800.

Alan Valejo, Brett Drury, Jorge Valverde-Rebaza, and Alneu de Andrade Lopes. 2014a. Identification of related brazilian portuguese verb groups using overlapping community detection. In *International Conference on Computational Processing of the Portuguese Language*. Springer, Cham, pages 292–297.

Alan Valejo, Jorge Carlos Valverde Rebaza, and Alneu de Andrade Lopes. 2014b. A multilevel approach for overlapping community detection. In *BRACIS 2014*. Springer, Berlin.

Alan Valejo, Jorge Valverde-Rebaza, Brett Drury, and Alneu de Andrade Lopes. 2014c. Multilevel Refinement Based on Neighborhood Similarity. In *Proceedings of the 18th International Database Engineering and Applications Symposium*. pages 67–76.

Chris H. Walshaw and Mark Cross. 2007. JOSTLE: Parallel multilevel graph-partitioning software - an overview. In *Mesh partitioning techniques and domain decomposition methods*. pages 27–58.

D. J. Watts and S. H. Strogatz. 1998. Collective dynamics of 'small-world' networks. *Nature* 393(6684):440–2.

Zhenqing Ye, Songnian Hu, and Jun Yu. 2008. Adaptive clustering algorithm for community detection in complex networks. *Phys. Rev. E* 78:046115.