Eur. Phys. J. Spec. Top. (2021) 230:2801–2811
https://doi.org/10.1140/epjs/s11734-021-00159-0

THE EUROPEAN
PHYSICAL JOURNAL
SPECIAL TOPICS

Regular Article

A review and comparative analysis of coarsening algorithms on bipartite networks

Alan Demétrius Baria Valejo[1,a], Wellington de Oliveira dos Santos[1], Murilo Coelho Naldi[1,b], and Liang Zhao[2,c]

[1] Department of Computer Science, Federal University of São Carlos (UFSCar), São Carlos, SP, Brazil
[2] Department of Computing and Mathematics (DCM), FFCLRP, University of São Paulo (USP), Ribeirão Preto, SP, Brazil

Received 19 November 2020 / Accepted 23 April 2021 / Published online 7 June 2021
© The Author(s), under exclusive licence to EDP Sciences, Springer-Verlag GmbH Germany, part of Springer Nature 2021

Abstract Coarsening algorithms have been successfully used as a powerful strategy to deal with data-intensive machine learning problems defined in bipartite networks, such as clustering, dimensionality reduction, and visualization. Their main goal is to build informative simplifications of the original network at different levels of details. Despite its widespread relevance, a comparative analysis of these algorithms and performance evaluation is needed. Additionally, some aspects of these algorithms' current versions have not been explored in their original or complementary studies. In that regard, we strive to fill this gap, presenting a formal and illustrative description of coarsening algorithms developed for bipartite networks. Afterward, we illustrate the usage of these algorithms in a set of emblematic problems. Finally, we evaluate and quantify their accuracy using quality and runtime measures in a set of thousands of synthetic and real-world networks with various properties and structures. The presented empirical analysis provides evidence to assess the strengths and shortcomings of such algorithms. Our study is a unified and useful resource that provides guidelines to researchers interested in learning about and applying these algorithms.

1 Introduction

A broadly pervasive class of networks are bipartite (two-layer or two-mode) networks, with two types of nodes, each type is at a "layer" and every link must connect nodes of different layers. Such networks are a realistic model of real-world systems, being widely used in science and technology to represent pairwise relationships between two categories of entities or phenomena, e.g., documents and terms [1], patient and gene expression (or clinical variables) [2] and individuals and songs (or books, or films) [3]. There has been a growing interest in bipartite networks given their relation to many data analytic problems, such as community detection [4] or text classification [5].

Recent advances in bipartite networks investigated coarsening algorithms to address hard machine learning problems, providing a broad spectrum of applications that includes network visualization [6], trajectory mining [7], optimization of high-expensive algorithms [8], community detection [6] and dimensionality reduction [8]. They build a hierarchy of reduced networks from an initial bipartite network, yielding multiple levels of detail, as presented in Fig. 1. Coarsening is well known for generating multiscale networks and, most notably, as a step of the well-known multilevel method [9].

Despite the potential and applicability of coarsening algorithms in bipartite networks, there is no comparative analysis that submits these algorithms to strict tests to evaluate their performance and robustness. Such a guideline is needed as the algorithms' performance can vary substantially for different scenarios. For instance, some algorithms can be sensitive to the level of noise in the network (a disturbance or error in the dataset and can negatively affect the algorithm's performance in terms of accuracy), may not be recommended to deal with unbalanced communities or sparse/dense networks. Furthermore, some aspects, properties, and variants of the current algorithms have not been explored in their original or complementary studies. For instance, some of these algorithms match pairs or groups of nodes based on a specific similarity measure called a common neighbor. However, other similarity measures can be used in these algorithms, including Jaccard, Sorensen, Adamic Adar, and Resource Allocation [10], whose results may vary in different networks. Therefore, a conceptual analysis of these algorithms, including a description of their direct variations, remains unexplored and, therefore, open to scientific investigation.

This review presents a comparative analysis of coarsening algorithms on bipartite networks. Specifically, it is composed of three-fold contributions:

[a] e-mail: alanvalejo@ufscar.br (corresponding author)
[b] e-mail: naldi@ufscar.br
[c] e-mail: zhao@usp.br

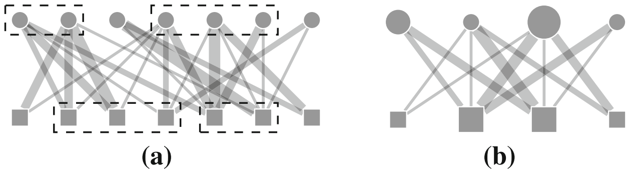


Fig. 1 Coarsening process in a bipartite network: In **a** group of nodes are matched (or selected) following an defined strategy; in **b** a coarsened bipartite network is build collapsing selected nodes into a single super-node and any link incident to matched nodes are collapsed into the so-called super-links. Henceforth line thickness denotes the relative link-weight

- *Literature review* We provide a conceptual and illustrative description of coarsening algorithms specifically designed to deal with bipartite networks. In addition to the key concepts, our discussion covers new aspects and variants of the current algorithms not yet been analyzed in the literature.
- *Applications* We present illustrative examples of how representative problems in bipartite networks can be addressed using the coarsening algorithms. We trust this is an essential reference material to encourage novel usages of these methods.
- *Comparative analysis* We performed a comparative study of coarsening algorithms and their proposed variants. The investigation is conducted on a representative set of thousands of networks, covering synthetic and real-world networks.

The remainder of the paper is organized as follows: in Sect. 2 we introduce some basic concepts, formally describes the state-of-the-art algorithms, and illustrates applications; empirical comparative analysis of the algorithms are reported in Sect. 5; lastly, in Sect. 6 we summarize our findings and discuss future work.

2 Coarsening algorithm in bipartite networks

An unipartite network is represented by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \sigma, \omega)$, wherein $\mathcal{V}$ and $\mathcal{E}$ defines the set of nodes and links, respectively, and a link $(v, u) = \{(u, v) = (v, u) \mid u, v \in \mathcal{V}\}$, i.e. an undirected network. The number of nodes and links are denoted by $n = |\mathcal{V}|$ and $m = |\mathcal{E}|$. A link (u, v) and a node u may have an associated weight, denoted by $\omega(u, v) : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}^+$ and $\sigma(u) : \mathcal{V} \rightarrow \mathbb{R}^+$. The network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \sigma, \omega)$ is bipartite if $\mathcal{V}$ is partitioned into two sub sets $\mathcal{V}^1$ and $\mathcal{V}^2$, such that $\mathcal{V}^1 \cap \mathcal{V}^2 = \emptyset$ and $\mathcal{E} \subseteq \mathcal{V}^1 \times \mathcal{V}^2$.

The h-hop neighborhood of u, denoted by $\Gamma_h(u)$, is the set of nodes whose distance from u is less than or equal to h. E.g, $\Gamma_1(u)$ is the set of adjacent nodes to u; $\Gamma_2(u)$ is the set of nodes 2-hops away from u, and so forth. The degree of a node u, denoted by $\kappa(u)$, is the number of its incident edges, i.e. $|\Gamma_1(u)|$.

A similarity measure (or *index*) quantify common characteristics between a pair of nodes (u, v), yielding values (scores) in the range $[0, 1] \subset \mathbb{R}$, from lowest (0) to highest (1) similarity [11]. A fundamental structural index is given by the number of common neighbors, defined as $CN(u, v) = |\Lambda(u, v)|$, wherein $\Lambda(u, v) = \Gamma_1(u) \cap \Gamma_1(v)$. A few of several well-known proposed indices are shown in Table 1.

Coarsening algorithms are adopted as a strategy to solve large-scale problems (or data-intensive machine learning problems) through a multiscale analysis of the original problem, involving a coarsening process that builds a sequence of networks at different levels of scale. These algorithms are also employed as a step of the well-known multilevel method, whose aims at reducing the computational cost of a target algorithm by applying it to the coarsest representation [9]. It operates in three phases:

- *Coarsening* The original network $\mathcal{G}_0$ is iteratively coarsened into a hierarchy of smaller networks $\{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_\mathcal{H}\}$, wherein $\mathcal{G}_\mathcal{H}$ is the coarsest one. The process implies collapsing nodes and links into single entities, referred to as super-node and super-link.
- *Initial solution* The target algorithm is applied or evaluated in the coarsest representation $\mathcal{G}_\mathcal{H}$, in which the initial solution is created.
- *Uncoarsening* The uncoarsening phase, known as solution projection, transfers the solution available at a current level to the next level in the hierarchy. The solution obtained in the coarsest network $\mathcal{G}_\mathcal{H}$ is successively projected through intermediate bipartite networks $\{\mathcal{G}_{\mathcal{H}-1}, \mathcal{G}_{\mathcal{H}-2}, \ldots, \mathcal{G}_1\}$ up to the initial network $\mathcal{G}_0$.

Notably, the coarsening is the key component of the multilevel method since it is problem-independent, in contrast to the other two phases designed according to the target task [9]. Therefore, several algorithms have been developed, and some strategies able to handle bipartite networks have gained notoriety recently.

One of the first, proposed in [4,12], called OPM (one-mode projection-based matching algorithm), textcolorreddecomposes the bipartite network $\mathcal{G}$ into two unipartite networks, one for each layer, i.e., $\mathcal{G}^1$ and $\mathcal{G}^2$. In this decomposition, called one-mode projection, nodes of the same type are connected if they share at least one common neighbor in the original bipartite representation, and the link-weight is defined by the number of neighbors (CN index) shared between them.

Hence, one-mode projection is a good view of information for bipartite networks. Since most algorithms and measures in network analysis consider unipartite networks, it is often practical to analyze a bipartite network on its one-mode projections. Therefore, it notably increases the range of analysis options since classic and

Table 1 Similarity measures between a pair (u, v)

Weighted Common Neighbors (WCN)	$\sum_{z\in\Lambda(u,v)} {}^{\omega(u,z)+\omega(v,z)}/_{log(1+\kappa(z))}$
Jaccard (JAC)	${}^{\lvert\Gamma_1(u)\cap\Gamma_1(v)\rvert}/_{\lvert\Gamma_1(u)\cup\Gamma_1(v)\rvert}$
Adamic Adar (AA)	$\sum_{z\in\Lambda(u,v)} {}^{1}/_{\log\kappa(z)}$
Hub Promoted (HP)	${}^{\lvert\Gamma_1(u)\cap\Gamma_1(v)\rvert}/_{\min\{\lvert\Gamma_1(u)\rvert,\lvert\Gamma_1(v)\rvert\}}$
Resource Allocation (RA)	$\sum_{z\in\Lambda(u,v)} {}^{1}/_{\kappa(z)}$
Preferential Attachment (PA)	$\kappa(u)\times\kappa(u)$

already established algorithms in the literature can be applied to bipartite networks. As an example, it is possible to apply to bipartite networks the well-known, and popular algorithm Heavy-Edge Matching (HEM) [13], in which a random node u is matched with the adjacent node v, if edge (u, v) has maximum weight overall adjacent edges to u.

Figure 2 illustrates the OPM execution using CN index and Fig. 2a shows the original bipartite network. Fig. 2b depicts the one-mode projections of each layer and the matching $\mathcal{M} = \{\{u_1, u_2\}, \{u_3, u_4\}, \{u_6, u_7\}, \{u_8, u_9\}\}$ obtained with the HEM algorithm to $\mathcal{G}_1$ and $\mathcal{G}_2$. Then, $\mathcal{M}$ is transferred to the original bipartite network and, finally, the algorithm creates the coarsened bipartite network, shown in Fig. 2c.

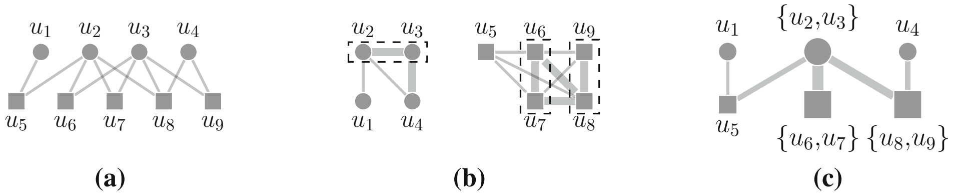
Employing one-mode projections has been the correct choice for most scenarios. However, some problems are derived from the network transformation, e.g., the connectivity may be artificially inflated with the introduction of fully connected sub-graphs, and some latent features can be lost [14,15]. An alternative is to directly perform the coarsening process on the bipartite structure, since it captures the system's behavior. Moreover, avoiding the one-mode projection results in computational savings.

From this perspective, [8] introduced two novel algorithms, called RGMb (Random greedy matching) and GMb (Greedy matching). They consider a two-hop neighborhood restriction, implying that nodes can only be matched with other nodes in their two-hop neighborhood set. Thereupon, this restriction reduces the local search space and the computational cost of finding a match.

In the RGMb, a node picked randomly u is matched with its unmatched two-hop neighbor v with maximal CN(u,v). Figure 3 illustrates RGMb. In an initial iteration, depicted in Fig. 3a, a selected node u_1 is allowed to match non-adjacent nodes u_2 or u_4, which belong to the two-hop neighborhood of u_1. Let us assume the pair $\{u_1, u_2\}$ is matched. In the next iteration, in Fig. 3b, a selected node u_3 could match nodes u_2 or u_4. Since u_2 has been matched, the remaining choice is to match pair u_3 and u_4. In the final iteration, the only choice left for the selected node u_6 is u_7, Fig. 3c. Finally, the coarsened bipartite network is built, Fig. 3d.

Alternatively, the GMb strategy randomly selects a node u and then chooses the best possible match from a list of its two-hop neighborhood sorted in decreasing order of index CN(u,v). Note that RGMb does not ensure an optimal choice overall possible matching, in contrast with GMb, which is more robust and can yield better results, albeit slower than its random search counterpart.

Although OPM, RGMb, and GMb have been proved helpful in several scenarios [4,8,12], strategies based on collapsing node pairs are inherently limited. Matches



u_1 u_2 u_3 u_4
u_5 u_6 u_7 u_8 u_9
(a)
u_2 u_3
u_1 u_4
u_5 u_6 u_9
u_7 u_8
(b)
u_1 $\{u_2,u_3\}$ u_4
u_5 $\{u_6,u_7\}$ $\{u_8,u_9\}$
(c)

Fig. 2 OPM coarsening resulting from its two one-mode projections with the HEM algorithm: **a** shows the orginal bipartite network; **b** depicts the one-mode projections of each layer; and **c** illustrates the coarsend network

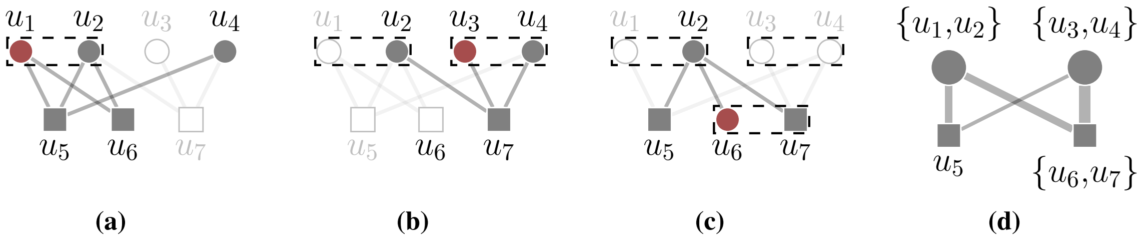


Fig. 3 RGMb algorithm based on two-hop neighborhoods: **a** shows the original bipartite network; **b** and **c** depict the matching; and **d** illustrates the coarsened network

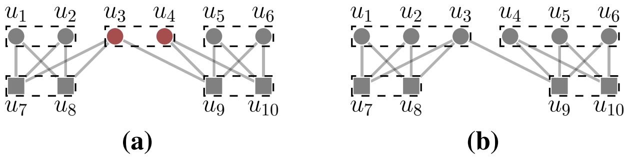


Fig. 4 In **a**, matching pair $\{u_3, u_4\}$ is a poor choice, as the resulting super-vertex will have predecessor vertices from distinct communities or cliques. In contrast, in **b**, group of nodes are matched together using MLPb coarsening algorithm, which avoids low-quality matches

at the early levels are more accurate than in-deep levels, i.e., they may introduce inconsistencies to the model that will be propagated to upper coarsening levels, which degrades the matching quality towards the final iterations. As a result, nodes from distinct communities may be matched, deteriorating the original topological structure. This problem is shown in Fig. 4a, which considers a network with two communities. Whereas suitable matching choices have been made in earlier iterations (to match pairs $\{u_1, u_2\}$ or $\{u_5, u_6\}$), the only choice in the final iteration is to match the pair $\{u_3, u_4\}$. This choice will degrade the community structure at the next coarsening level since it yields a super-node that includes nodes from distinct communities.

To suppress the presented drawback, [6] introduced a coarsening strategy based on a well-known Label Propagation Algorithm (LPA) [16] called MLPb (Multilevel label propagation for bipartite networks). It propagates node labels throughout the bipartite network, and a single label remains within a group of matched nodes that will collapse into a single super-node, allowing more than two nodes to collapse at once, as illustrated in Fig. 4b. Specifically, a unique label is assigned to each node, updated with the most frequent label in its two-hop neighborhood at each iteration. Like its predecessors, the MLPb inherits the label propagation based on the CN index.

3 Discussion

The aforementioned algorithms share an essential premise that relies on the local and structural CN index to compute nodes' similarity. Although these algorithms have been validated in different scenarios, the original studies do not analyze different similarity measures and their aftereffect in the coarsened representations.

According to [17], each index covers a specific network structural behavior. To cite a few: JAC index prevents hub nodes, i.e, the highest-degree nodes are often called hubs. to have a high score with other nodes; AA index implies that a lower-connected neighbor is more important than a hub; HP index assigns higher scores to nodes adjacent to hubs; RA index punishes the higher degree nodes more heavily; PA index is based on the growth of the network in the sense of new nodes emerging.

Different similarity measures are presented in [18, 19]. There, OPM defines link-weights in the projections by several indices, such as JAC, HP, RA, or PA, and can derive a specific connectivity pattern. Consequently, selecting the best can be a critical issue, as illustrated in Fig. 5, in contrast to the example depicted in Fig. 2. In this case, the same network is assessed with the OPM using the HP index to build the one-mode projections. Figure 5a illustrates the original networks, 5b shows the corresponding one-mode projection of each layer and the matching $\mathcal{M} = \{\{u_2, u_3\}, \{u_6, u_7\}, \{u_9, u_8\}\}$ and 5c depicts the coarsened bipartite network. The HP index reduces unmatched nodes since lower-degree nodes are matched first, whereas the CN index first matches hub nodes. Therefore, this OPM variant builds projections with different connectivity patterns and leads to a new coarsened representation.

Similarly, Fig. 6a and b report the RGMb in the same network using PA and JAC index, respectively. Note that PA prefers to connect high-degree nodes, like the pair $\{u_2, u_3\}$, whereas the JAC is based on the intersection over union, which makes hubs fail to influence all nodes.

4 Applications

In general, coarsening algorithms have been successfully used to optimize bipartite networks. In this case, the original network is successively coarsened until attaining a sufficiently small network, and, consequently, employing a computationally expensive algorithm over the result becomes feasible. However, coarsening algorithms can also be used directly on several machine learning problems; some are described in this section.

4.1 Community detection

Community detection established itself as a benchmark problem for the coarsening algorithms. It aims to split the network into disjoint groups of nodes densely connected between them, called communities, and sparsely connected to the other groups [20]. These structures are essential and frequently found in many real-world networks.

There are two types of community in bipartite networks: one-to-one correspondence, where there is the same number of communities in each layer, and the communities are correspondents between layers; many-to-many correspondence, with a different number of communities in the layers and the communities are independents between layers.

Employing a coarsening strategy directly as a community detection algorithm is straightforward [6] and refers to many-to-many correspondence. Figure 7 illustrates this process. The original networks are coarsened, 7b and c, then each super-node in the coarsest network is mapped as a community, 7d. The mapping

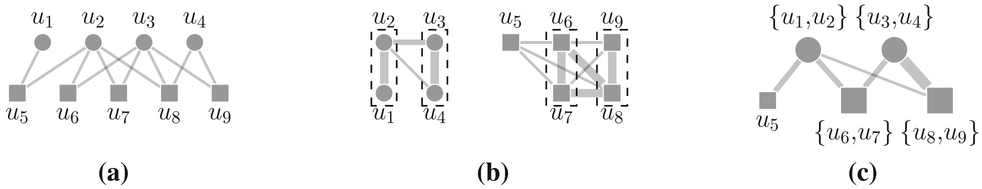

u_1 u_2 u_3 u_4
u_5 u_6 u_7 u_8 u_9
(a)
u_2 u_3 u_5 u_6 u_9
u_1 u_4 u_7 u_8
(b)
$\{u_1,u_2\}$ $\{u_3,u_4\}$
u_5 $\{u_6,u_7\}$ $\{u_8,u_9\}$
(c)

Fig. 5 OPM coarsening resulting from its two one-mode projections using HP index and then the HEM coarsening algorithm: **a** shows the orginal bipartite network; **b** depicts the matching; **c** and **d** illutrates the coarsened network

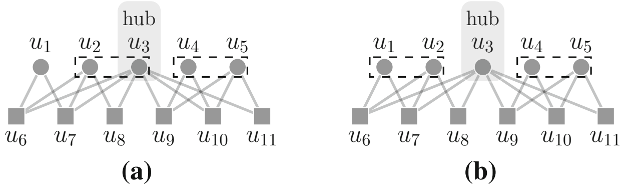

hub
u_1 u_2 u_3 u_4 u_5
u_6 u_7 u_8 u_9 u_{10} u_{11}
(a)
hub
u_1 u_2 u_3 u_4 u_5
u_6 u_7 u_8 u_9 u_{10} u_{11}
(b)

Fig. 6 RGMb algorithm using different indices to compute the similarity between vertices. In **a** the matching $\mathcal{M} = \{(u_2, u_3), (u_4, u_5)\}$ is chosen using the PA index; whereas **b** depicts the the matching $\mathcal{M} = \{(u_1, u_2), (u_3, u_4)\}$ selected using the JAC index

is successively projected back through the hierarchy so that nodes $\in \mathcal{G}_{\mathcal{H}}$ inherit the communities assigned to its successor super-nodes $\in \mathcal{G}_{\mathcal{H}+1}$, as illustrated in 7e and f.

4.2 Dimensionality reduction

Dimensionality usually refers to the number of attributes (or features) in a dataset. Machine learning tasks are often preceded by a dimension reduction step that leads the data to a low-dimensional space that retains some meaningful structural properties of its original form. A coarsening strategy for this application models the data matrix as a bipartite network, in which objects (rows) and attributes (columns) are associated with the two layers $\mathcal{V}^1$ and $\mathcal{V}^2$ and non-zero matrix entries denote the link weights. Afterward, the coarsening process is applied in the attribute layer, reducing the data's dimensionality while preserving the object layer.

Formally, a matrix $X_{r\times s}$ is modeled from a bipartite network, wherein $r = |\mathcal{V}_0^1|$, $s = |\mathcal{V}_0^2|$ and $\omega(u, v) = X_{u,v}$ if $X_{u,v} \neq 0$. The aim is to create a lower-dimensional matrix $X'_{r'\times s'}$ with $r'|\mathcal{V}_{\mathcal{H}}^1|$, $s' = |\mathcal{V}_{\mathcal{H}}^2|$, as illustrated by Fig. 8. The original matrix, its corresponding bipartite network, X_0 and $\mathcal{G}_0$ and the matching $\mathcal{M} = \{\{u_1, u_2\}, \{u_3, s_4\}\}$ is depicted in Fig. 8a. The coarsened network $\mathcal{G}_1$ and its low-dimensional matrix representation is shown in Fig. 8b.

An example is presented through two well-known datasets: Iris with four attributes, 150 objects, and three classes related to species of iris (Setosa, Versicolour, and Virginica) and Wine with 13 attributes, 178 objects, and three classes that denotes three types of wines. We reduce the number of both datasets' features to two dimensions and compare the multilevel strategy (MS) with the Principal Component Analysis (PCA) method. Figure 9a and b show the result of PCA and MS evaluated in Iris dataset, respectively. Figure 9c and d show the result of PCA and MS evaluated in Wine dataset, respectively. In both datasets, the results obtained by the MS are close to those obtained by the PCA. See [8] for an in-deep empirical analysis.

4.3 Combinatorial optimization problems on graphs

The multilevel method reveals the potential to support the solution of a range of combinatorial optimization problems defined on graphs, like traveling salesman problems, graph drawing, graph coloring, and matching, as reported in [21]. For instance, graph coloring is a fundamental problem in graph theory, widely used to solve scheduling problems, wherein access to shared resources must be synchronized. The goal is to assign different colors to adjacent nodes, ensuring a minimum number of colors.

One instance of this problem is to check whether a graph is bipartite or not. Such an assumption is true if it is possible to coloring the graph using only two colors. Breadth-First Search (BFS) and backtracking are the standard approaches. Alternatively, a coarsening-based solution imposes that nodes can match others only in their two-hop neighborhood set. If the resultant coarsest network is a 2-colorable graph, it is bipartite. If there is a self-loops in an arbitrary coarsening level, it implies that the graph is not bipartite. Figure 10 illustrates this strategy.

4.4 Visualization

Network visualization deals with creating visual representations of networked data that support the user-driven exploratory investigation. However, this task faces severe limitations when a large-scale network is evaluated: a large-scale network associated with a small screen can affect the readability of the connectivity and topological patterns, which implies overlapping the graphical elements. Furthermore, the high cost to compute the network layout may prevent real-time rendering [22,23].

Coarsening algorithms can potentially mitigate the presented limitations enabling the user to interact with the visualization in multiple levels of detail. This process is achieved as follows: the network is successively coarsened until attaining a sufficiently small representation; a layout algorithm is employed to draw the coarsest network; the user interacts with the initial visual representation, as illustrated in 11. The multiscale hierarchy allows an interactive process supported by on-demand local (or global) expansion at different levels of detail.

Figure 11a shows a dense network with seven unbalanced communities, $|\mathcal{V}^1| = 10,000$ and $|\mathcal{V}^2| = 6,000$, overlapping and noise. All layouts were computed with the Fruchterman-Reingold force-directed [24]. The high ratio of inter-community links in the network hinders the separation of communities on the screen, implying blurred community boundaries. Furthermore, the high number of graphical elements (nodes and links) in a small screen hamper the distinction of vertex types and links intra/inter communities. For example, figures b–d illustrate three levels of reduction. The network topology is more evident in the coarsened representations, i.e., they allow to observe the presence of seven communities and their boundaries clearly. Moreover, the reduced network preserves the design of the original network.

5 A comparative analysis

We conducted three experimental studies to assess the presented coarsening algorithms:

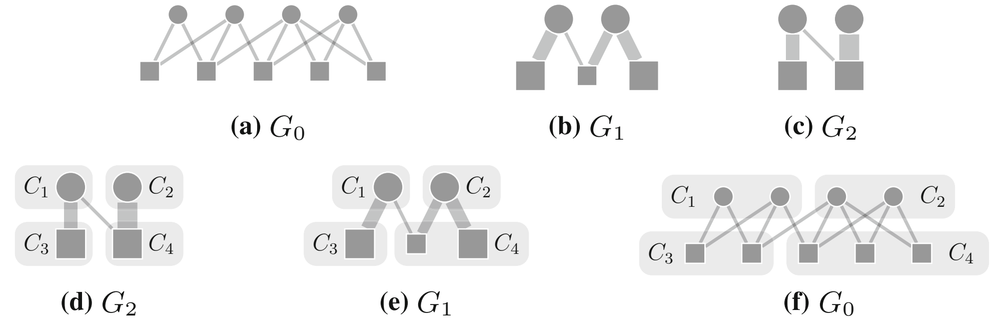
1. We analyzed the performance of the coarsening algorithms using different indices.
2. We compared the accuracy and runtime of different coarsening strategies.
3. We evaluated the coarsening algorithms in real-world networks.

In this study, the following measures were considered: normalized mutual information (NMI) [25], Murata's Modularity [26] and runtime (in s). A Nemenyi post-hoc test [27] was applied to the results to detect statistical differences in the performances of the algorithms. We used a black line to connect algorithms with no significant performance difference to visualize the test results. Synthetic networks were obtained employing a network generation tool called BNOC, proposed in [28]. Real-world bipartite networks used are available at KONECT (the Koblenz Network Collection) [29]. We selected the largest connected component of each of these networks. Experiments were executed on a Linux machine with a 6-core processor with 2.60 GHz and 16 GB main memory. We report the average values obtained from ten executions for all algorithms in each network.

5.1 Analysis of similarity measures

We evaluated the performance of the algorithms OPM and RGMb using different indices regarding two synthetic network settings, specifically, noise level and the number of communities. NMI was used to obtain the accuracy of the algorithms.

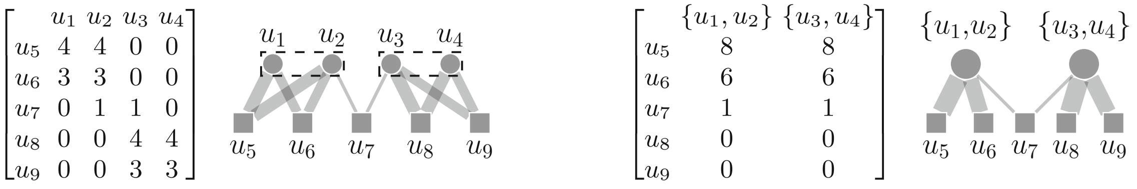
A set of 1000 synthetic bipartite networks with distinct noise level was made with the following characteristics: $|\mathcal{V}| = 2,000$ with $|\mathcal{V}^1| = |\mathcal{V}^2|$, noise within the range $[0.0, 0.5]$ and 20 communities for each layer. Figure 12a depicts the NMI values for the OPM variants as



(a) G_0 **(b)** G_1 **(c)** G_2

(d) G_2 **(e)** G_1 **(f)** G_0

Fig. 7 Coarsening algorithm as a step of the multilevel method to community detection in bipartite networks; **a** shows the original bipartite network; **b**, **c** **d** and **e** depict the coarsening process; and **f** illustrates the final solution, i.e. the community structure



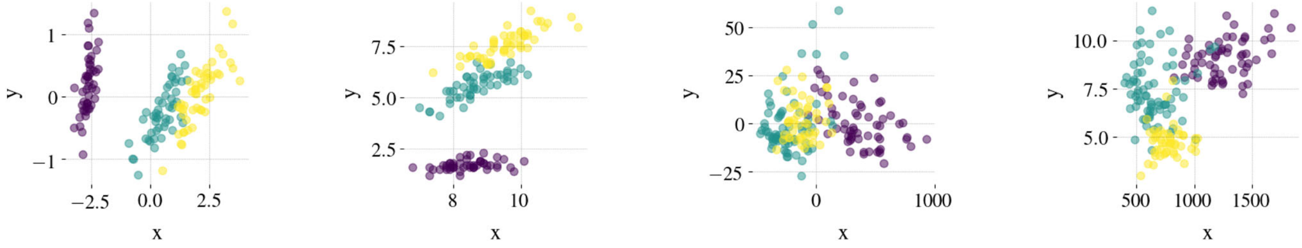
$$\begin{array}{c|cccc} & u_1 & u_2 & u_3 & u_4 \\ \hline u_5 & 4 & 4 & 0 & 0 \\ u_6 & 3 & 3 & 0 & 0 \\ u_7 & 0 & 1 & 1 & 0 \\ u_8 & 0 & 0 & 4 & 4 \\ u_9 & 0 & 0 & 3 & 3 \end{array}$$

(a) X_0 and G_0

$$\begin{array}{c|cc} & \{u_1,u_2\} & \{u_3,u_4\} \\ \hline u_5 & 8 & 8 \\ u_6 & 6 & 6 \\ u_7 & 1 & 1 \\ u_8 & 0 & 0 \\ u_9 & 0 & 0 \end{array}$$

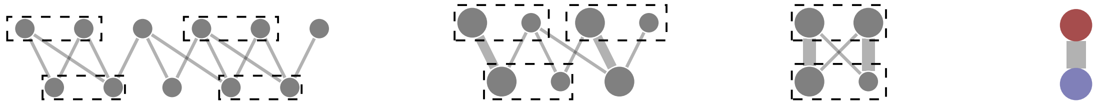
(b) X_1 and G_1

Fig. 8 Multilevel dimensionality reduction in a bipartite network: **a** report the original matrix and its bipartite representation and **b** summarizes the coarsened network and the low-dimensional matrix



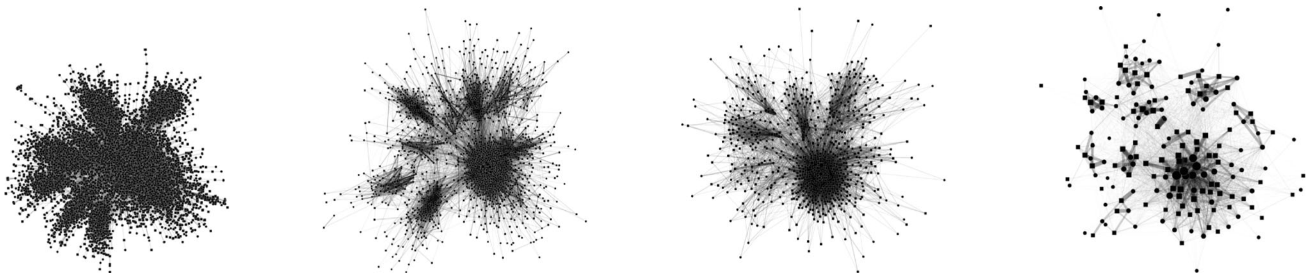
(a) Iris/PCA **(b)** Iris/MS **(c)** Wine/PCA **(d)** Wine/MS

Fig. 9 Results of the MS and PCA method in the Iris dataset, (**a**, **b**), and wine, (**c**, **d**) datasets. Each color represent a class



(a) G_0 **(b)** G_1 **(c)** G_2 **(d)** G_3

Fig. 10 Check whether an graph is bipartite or not using a multilevel strategy



(a) **(b)** **(c)** **(d)**

Fig. 11 Coarsening hierarchy generated from a bipartite network with $|\mathcal{V}| = 16,000$ nodes, seven unbalanced communities and extensive noise and overlapping: **a** shows the orginal bipartite network; and **b**, **c**, **d** and **e** illustrates the coarsening hiearchy

a function of the amount of noise. The noise level is the proportion of edges wrongly inserted, i.e., 0.5 means that half of the edges are not what they should be. OPM_{pa} revealed the worst accuracy among all variations. Regarding the other variants, OPM_{wcn} obtained the best NMI values with a low level of noise; however, it obtains the worst results after 0.3 noise level. Interestingly, OPM$_{\mathrm{hp}}$ variation obtained the best performances after 0.3 noise level. A Nemenyi post-hoc test is shown in Fig. 12c. The critical value for comparing the mean-ranking of two different algorithms at 95 percentile is 0.04. Both OPM$_{\mathrm{wcn}}$ and OPM$_{\mathrm{hp}}$ were ranked first followed by the group OPM$_{\mathrm{jac}}$, OPM$_{\mathrm{cn}}$, OPM$_{\mathrm{ra}}$ and OPM$_{\mathrm{aa}}$ with no statistically significant difference.

Additionally, we generated a set of 1000 synthetic bipartite networks with a variety of numbers of communities, as follows: $|\mathcal{V}| = 2,000$ with $|\mathcal{V}^1| = |\mathcal{V}^2|$, communities within the range $[1, 500]$ and 0.3 of the noise level. Figure 12b depicts the NMI values for the RGMb variants as a function of the number of communities. The results are similar to those obtained for the previous set regarding the OPM$_{\mathrm{wcn}}$ that starts with the best accuracy. However, its performance decrease with an increase in the number of communities. A Nemenyi post-hoc test is shown in Fig. 12d. The critical value for comparing the mean-ranking of two different algorithms in the 95 percentile is 0.04. OPM$_{\mathrm{jac}}$ was ranked first and OPM$_{\mathrm{pa}}$ in last.

Figure 12c and d depict the Nemenyi post-hoc test over the results in Fig. 12a and b. The critical value in the 95 percentile is 0.12 and 0.09, respectively. OPM$_{\mathrm{hp}}$ and RGMb$_{\mathrm{jac}}$ were best ranked in the first and second diagram.

5.2 Comparative analysis on synthetic networks

We compared the runtime and accuracy of coarsening strategies for different synthetic network settings. OPM was set with WCN index. A set of 1000 synthetic networks with distinct noise levels was generated as follows: $|\mathcal{V}| = 2,000$ with $|\mathcal{V}^1| = |\mathcal{V}^2|$, noise within the range $[0.0, 1.0]$ and 20 communities for each layer. Figure 13a depicts the NMI values for the algorithms as a function of the amount of noise. MLPb obtained high NMI values with low noise; however, the accuracy decreases quickly after the 0.22 noise level.

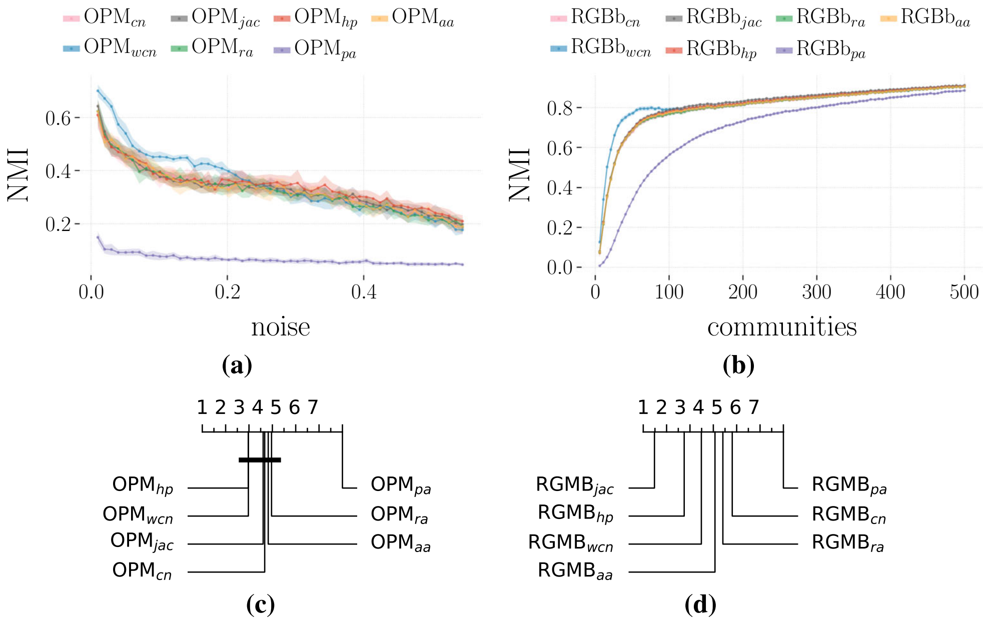


Fig. 12 Performance of seven OPM and RGMb variants in 2000 synthetic networks: **a**, **b** illustrates NMI results as a function of noise and number of communities, respectively. **c**, **d** depict the Nemenyi post-hoc test applied to the results shown in **a** and **b**, respectively

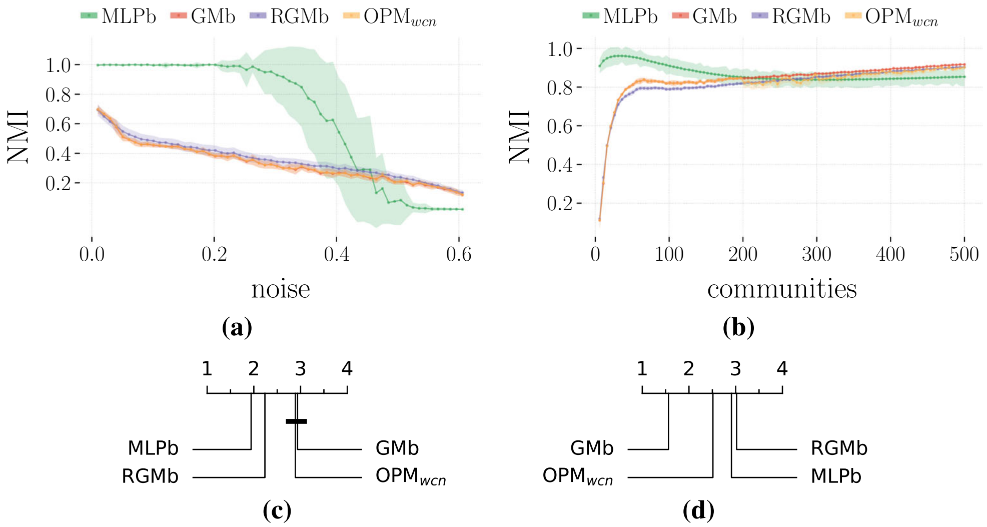


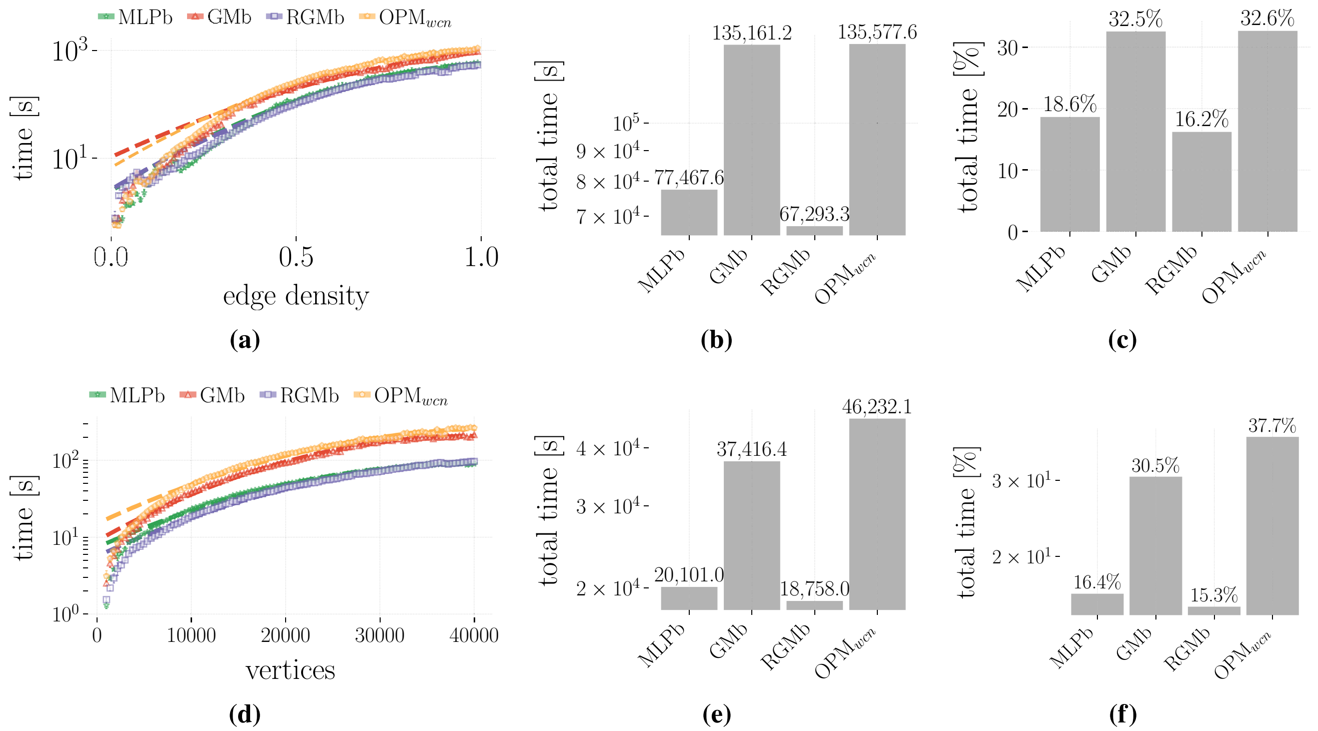
Fig. 13 Performance of the evaluated algorithms in 2000 synthetic networks: **a**, **b** illustrate NMI results as a function of noise and number of communities, respectively. **c**, **d** Depict the Nemenyi post-hoc test applied to the results shown in **a** and **b**, respectively

Therefore, MLPb revealed sensibility to a high-noise level. Although GMb, RGMb, and OPM algorithms obtained the lowest NMI values, mainly within the range $[0.0, 0.4]$, their performances decrease slowly compared with MLPb.

A set of 1000 synthetic networks with a different number of communities was generated, as follows: $|\mathcal{V}| = 2,000$ with $|\mathcal{V}^1| = |\mathcal{V}^2|$, communities within the range $[1, 500]$ and 0.3 of the noise level. Figure 13b depicts the NMI values for the evaluated algorithm as a function of the number of communities. GMb, RGMb, and OPM presented a high sensibility to a low number of communities, specifically, within the range $[1, 100]$. In contrast, MLPb obtained high NMI values in the same range. Within the range, $[200, 500]$ all algorithms obtained NMI values close to each other.

Figure 13c and d depict the Nemenyi post-hoc test applied to the results in Fig. 13a and b. The critical values at the 95 percentile are 0.06 and 0.05, respectively. MLPb and GMb were best ranked in the first and second diagram.

We assessed the scalability of the algorithms in terms of the absolute and relative total time spent. First, we built a set of 1000 synthetic networks with a variety



(a) **(b)** **(c)**

(d) **(e)** **(f)**

Fig. 14 Runtime as a function of the number of links (**a**–**c**) and nodes (**d**–**f**) for five coarsening algorithms to build the coarsest representation in 2000 synthetic networks: the pairs (**b**–**e**) and **c**–**f** shows the absolute and relative total time respectively

of link-densities within the range $[0.01, 0.99]$, wherein 0.01 indicates very sparse networks and 0.99 indicates very dense networks with $m \approx n^2$; $|\mathcal{V}| = 5000$ with $|\mathcal{V}^1| = |\mathcal{V}^2|$ and 20 communities at each layer. Figure 14 shows how each algorithm contributed to the total time, in both absolute values, Fig. 14a and b, and relative values, Fig. 14c (percentages shown on top of the bars). The total time spent running the experiments was $419,857.968$ s.

Moreover, a set of 1000 synthetic bipartite networks was created, varying the number of nodes within the range $[1,000, 40,000]$ and communities as a percentage of the number of nodes, i.e., $|\mathcal{V}| * 0.01$. Figure 14 shows how each algorithm contributed to the total time, in both absolute values, Fig. 14d and e, and relative values, Fig. 14f (percentages shown on top of the bars). The total time spent running the experiments was $128,151.711$ s or nearly 35 h. RGMb was the fastest in both cases and ran 18–35 times faster than the other algorithms. GMb and OPM$_{\text{hem}}$ were the most computationally expensive algorithms.

5.3 Comparative analysis on real-world networks

In order to verify if the results over synthetic data hold for real-world application, we considered six additional bipartite networks, which properties are detailed in Table 2(a). Murata's modularity was used to obtain the algorithms' accuracy by reducing the networks to 10%, 30%, 50%, 80%, and 95% of their original sizes.

For 10% of network reduction, summarized in Table 2(b), GMb and OPM$_{\text{wcn}}$ yielded the best values in six out of seven networks, losing for MLPb on Movielens solely. Considering 30%, 50%, 80%, and 95% of network reduction, summarized in Table 2(c)–(f), MLPb yielded the best values in almost all networks, leaving GMb and OPM$_{\text{wcn}}$ in second place with the best values for one.

To assess the results' statistical significance, the Nemenyi post-hoc test was applied and presented in Fig. 15. The critical at the 95 percentile is 0.38. MLPb was best ranked and performed statistically better than the other algorithms.

6 Conclusion

This review integrates the current knowledge of coarsening strategies specifically designed to deal with bipartite networks, a subject that was not deeply reviewed in previous studies. As an initial contribution, we present an overview (formal and illustrative) of coarsening algorithms. We also introduced a discussion covering the use of different similarity measures and their aftereffect in the coarsened representations.

Then, we presented illustrative examples on the use of coarsening algorithms in representative problems defined in bipartite networks, specifically, community detection, dimensionality reduction, visualization, and classical graph problems. Moreover, we conducted an empirical analysis of a representative set of thousands of

Table 2 Modularity scores of the algorithms: (b), (c), (d), (e) and (f) present modularity scores of the algorithms considering 10%, 30%, 50%, 80% and 95% of network reduction

(a)

Dataset	$\|\mathcal{V}^1\|$	$\|\mathcal{V}^2\|$	$\|\mathcal{E}\|$
Ucforum	248	610	1,249
MCrime	754	509	1,377
N-reactome	8,788	15,433	41,087
Condmat	13,861	19,466	53,628
Movielens	3,919	2,378	8,868
Dbpedia	54,909	19,866	98,895

(b)

Dataset	MLPb	GMb	RGMb	OPM_{wcn}
Ucforum	0.122	**0.133**	0.115	**0.133**
Moreno	0.499	**0.525**	0.497	**0.525**
N-reactome	0.344	**0.364**	0.332	**0.364**
Condmat	0.368	**0.411**	0.358	**0.411**
Movielens	**0.244**	0.240	0.231	0.240
Dbpedia	0.425	**0.451**	0.412	**0.451**

(c)

Dataset	MLPb	GMb	RGMb	OPM_{wcn}
Ucforum	**0.182**	0.135	0.120	0.135
Moreno	**0.565**	0.556	0.541	0.556
N-reactome	**0.435**	0.431	0.391	0.431
Condmat	**0.462**	0.454	0.409	0.454
Movielens	**0.279**	0.242	0.233	0.242
Dbpedia	**0.507**	0.475	0.449	0.475

(d)

Dataset	MLPb	GMb	RGMb	OPM_{wcn}
Ucforum	**0.266**	0.145	0.119	0.145
Moreno	0.618	**0.630**	0.583	**0.630**
N-reactome	**0.521**	0.467	0.427	0.467
condmat	**0.557**	0.530	0.454	0.530
movielens	**0.318**	0.248	0.234	0.248
dbpedia	**0.584**	0.517	0.48	0.517

(e)

Dataset	MLPb	GMb	RGMb	OPM_{wcn}
Ucforum	**0.404**	0.129	0.109	0.129
Moreno	0.731	**0.773**	0.663	**0.773**
N-reactome	**0.571**	0.496	0.437	0.496
Condmat	**0.701**	0.611	0.536	0.611
Movielens	**0.404**	0.254	0.22	0.254
Dbpedia	**0.690**	0.542	0.524	0.542

(f)

Dataset	MLPb	GMb	RGMb	OPM_{wcn}
Ucforum	**0.481**	0.184	0.092	0.172
Moreno	0.627	**0.756**	0.707	0.699
N-reactome	**0.623**	0.506	0.393	0.506
Condmat	0.600	**0.619**	0.533	**0.619**
Movielens	**0.458**	0.283	0.179	0.283
Dbpedia	**0.581**	0.531	0.503	0.531

The highest values are in bold

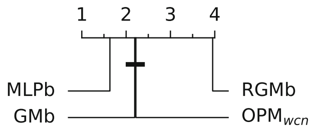


Fig. 15 Nemenyi post-hoc test summarizes the overall results depicted in Table 2

networks. Considering the study on similarity measures, the results allow us to conclude that WCN index is more accurate in general terms, except for more complex network settings, where JAC and HP indices are indicated. The comparative analysis between different coarsening algorithms analysis suggests that MLPb yielded more accurate and stable results and requires considerably lower execution time. However, in complex scenarios, e.g., the zoomed-in plots focusing on networks with high noise and number of community, GMb, and OPM_{wcn} algorithms showed better results. Furthermore, RGMb was the fastest, whereas GMb and OPM_{hem} were the most expensive algorithms.

Coarsening algorithms are commonly approached from parallel, distributed or GPU-based paradigms considering unipartite networks [30,31]; nonetheless, these paradigms have not yet been explored in bipartite networks, with research potential.

Along these lines, we trust the present work is an essential and unified reference material to encourage novel adoptions of the surveyed methods, inspire innovative lines of investigation, and pave further development of cutting-edge applications.

Acknowledgements This research was funded by the State of São Paulo Research Foundation (FAPESP) grants 2013/07375-0, 15/50122-0, 2019/09817-6, 19/07665-4 and 2019/14429-5; the Brazilian Federal Research Council (C NPq) grant 303199/2019-9; the C4AI grant FAPESP/IBM /USP: 2019/07665-4, the German Research Foundation (DFG) (Project IRTG/GRK 1740), and the Ministry of Science and Technology of China grant G20200226016.

Data Availability Statement My manuscript has no associated data or the data will not be deposited.

References

1. T.P. de Faleiros, R.G. Rossi, A.A. Lopes, Optimizing the class information divergence for transductive classification of texts using propagation in bipartite graphs. Pattern Recognit. Lett. **87**, 127–138 (2017)
2. T. Hwang, H. Sicotte, Z. Tian, W. Baolin, J.-P. Kocher, D.A. Wigle, V. Kumar, R. Kuang, Robust and efficient

identification of biomarkers by classifying features on graphs. Bioinformatics **24**(18), 2023–2029 (2008)

3. J. Grujić, Movies recommendation networks as bipartite graphs. In: Proceedings of the international conference on computational science (ICCS)
4. A. Valejo, V. Ferreira, M.C.F. Oliveira, A.A. Lopes, Community detection in bipartite network: a modified coarsening approach. In: International symposium on information management and big data (SIMBig), track on SNMAN. Communications in computer and information science book series (CCIS, vol. 795)
5. T. Faleiros, A. Valejo, A.A. de Lopes, Unsupervised learning of textual pattern based on propagation in bipartite graph. Intell. Data Anal. **24**(3), 543–565 (2020)
6. A. Valejo, T.P. Faleiros, M.C.F. Oliveira, A. Lopes, A coarsening method for bipartite networks via weight-constrained label propagation. Knowl. Based Syst. **195**, 105678 (2020)
7. D. Minatel, A. Valejo, A.A. Lopes, Trajectory network assessment based on analysis of stay points cluster. In: Brazilian conference on intelligent systems (BRACIS) (2018), pp. 564–569
8. A. Valejo, M.C.G. Oliveira, G.P.R. Filho, A.A. Lopes, Multilevel approach for combinatorial optimization in bipartite network. Knowl. Based Syst. **151**, 45–61 (2018)
9. Alan Valejo, V. Ferreira, R. Fabbri, M.C.F. Oliveira, A. Lopes, A critical survey of the multilevel method in complex networks. ACM Comput. Surv. **53**(2), 35 (2020)
10. A. Rawashdeh, A.L. Ralescu, Similarity measure for social networks-a brief survey. In: Modern AI and cognitive science conference (MAICS)
11. A. Valejo, J. Valverde-Rebaza, B. Drury, A.A. De Lopes, Multilevel refinement based on neighborhood similarity. In: International database engineering and applications symposium (IDEAS)
12. A. Valejo, A.A. Lopes, G.P.R. Filho, M.C.F. Oliveira, V. Ferreira, One-mode projection-based multilevel approach for community detection in bipartite networks. In: International symposium on information management and big data (SIMBig), track on social network and media analysis and mining (SNMAN) (2017), pp. 101–108
13. G. Karypis, V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J. Sci. Comput. **20**(1), 359–392 (1998)
14. M. Kitsak, D. Krioukov, Hidden variables in bipartite networks. Phys. Rev. E **84**(2), 026114 (2011)
15. M. Kitsak, F. Papadopoulos, D. Krioukov, Latent geometry of bipartite networks. Phys. Rev. E **95**, 032309 (2017)
16. U.N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks. Phys. Rev. E **76**, 036106 (2007)
17. L. Linyuan, T. Zhou, Link prediction in weighted networks: the role of weak ties. Europhys. Lett. **89**(1), 18001 (2010)
18. S.P. Borgatti, D.S. Halgin, Analyzing affiliation networks. Sage Handb. Soc. Netw. Anal. **1**, 417–433 (2011)
19. S. Banerjee, M. Jenamani, D.K. Pratihar, Properties of a projected network of a bipartite network. In: 2017 International conference on communication and signal processing (ICCSP) (IEEE, 2017), pp. 0143–0147
20. S. Fortunato, Community detection in graphs. Phys. Rep. **486**(3–5), 75–174 (2010)
21. C. Walshaw, M.G. Everett, Multilevel landscapes in combinatorial optimisation. Technical report, Computing and Mathematical Sciences, University of Greenwich (2002)
22. J. Díaz, J. Petit, M. Serna, A survey of graph layout problems. ACM Comput. Survey **34**(3), 313–356 (2000)
23. T. von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J.J. van Wijk, J.-D. Fekete, D.W. Fellner, Visual analysis of large graphs: state-of-the-art and future research challenges. Comput. Graph. Forum **30**(6), 1719–1749 (2011)
24. T.M.J. Fruchterman, E.M. Reingold, Graph drawing by force-directed placement. Softw. Pract. Exp. **21**(11), 1129–1164 (1991)
25. L. Danon, A. Diaz-Guilera, J. Duch, A. Arenas, Comparing community structure identification. J. Stat. Mech. Theory Exp. **2005**(09), P09008 (2005)
26. T. Murata, Modularities for bipartite networks. In: Proceedings of the 20th ACM conference on hypertext and hypermedia (2009), pp. 245–250
27. J. Demšar, Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)
28. A. Valejo, F. Goes, L.M. Romanetto, M.C.F. Oliveira, A.A. Lopes, A benchmarking tool for the generation of bipartite network models with overlapping communities. Knowl. Inf. Syst. **62**, 1641–1669 (2019)
29. J. Kunegis, Konect: the koblenz network collection. In: Proceedings of the 22nd international conference on World Wide Web (2013), pp. 1343–1350
30. G. Karypis, V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J. Sci. Comput. **20**(1), 359–392 (1998)
31. B.F. Auer, R.H. Bisseling, Graph coarsening and clustering on the GPU. Graph Partit. Graph Clust. **588**, 223 (2012)