Linear Time Community Detection by a Novel Modularity Gain Acceleration in Label Propagation

Sakineh Yazdanparast, Mohsen Jamalabdollahi, and Timothy C. Havens, *Senior Member, IEEE*

Abstract—Community detection is an important problem in complex network analysis. Among numerous approaches for community detection, label propagation (LP) has attracted a lot of attention. LP selects the optimum community (i.e., label) of a network vertex by optimizing an objective function (e.g., Newman's modularity) subject to the available labels in the vicinity of the vertex. In this paper, a novel analysis of Newman's modularity gain with respect to label transitions in graphs is presented. Here, we propose a new form of Newman's modularity gain calculation that quantifies available label transitions for any LP based community detection. The proposed approach is called Modularity Gain Acceleration (MGA) and is simplified and divided into two components, the local and global sum-weights. The Local Sum-Weight (LSW) is the component with lower complexity and is calculated for each candidate label transition. The General Sum-Weight (GSW) is more computationally complex, and is calculated only once per each label. GSW is updated by leveraging a simple process for each node-label transition, instead of for all available labels. The proposed technique is applied to selected state-of-the-art LP-based community detection methods and the resulting network modularity and execution time are compared with traditional methods over small to big real world data sets. By applying MGA to LP-based methods, the run-time is significantly reduced—sometimes finishing before the traditional approach even finishes one iteration—achieving the same modularity performance and number of communities, i.e., community detection result. The MGA approach leads to significant efficiency improvements by reducing time consumption up to 85% relative to the original algorithms with the exact same quality in terms of modularity value which is highly valuable in analyses of big data sets.

Index Terms—Social Network, Fast Louvain Clustering, Big Data Analysis, Community Detection, Modularity Maximization, Label Propagation, Graph Partitioning

1 Introduction

Network analysis is a very well researched topic in graph theory. Recently, community discovery for complex networks has drawn numerous attention. Community is a prominent structure in networks which refers to group of nodes that happens to have more connections (edge) among themselves relative to edges that connect them to the rest of the network. Community detection and graph clustering are categorized as NP-complete problems with no globally optimal solution [1]. Applications of community detection include social network analysis, online commodity recommendation system, user clustering, biology, communication networks analysis, engineering, economics, etc.

Over the decades numerous community detection algorithms have been proposed. Modularity-based clustering is one of the most prominent approaches in community detection [2]. Modularity based techniques optimize a quality objective function such as *modularity* introduced by Newman and Girvan [3] with respect to networks community. According to Newman's criteria, communities leading to higher modularity value have denser connections between the nodes within them compared to nodes of other communities [4–6].

- *S. Yazdanparast is with the Department of Electrical and Computer Engineering, Michigan Technological University, Houghton, Michigan, 49931. Email: syazdanp@mtu.edu.*
- *T.C. Havens is with the College of Computing, Michigan Technological University, Houghton, Michigan, 49931. Email: thavens@mtu.edu.*
- *M. Jamalabdollahi is with Cisco Systems, Inc, Richfiled, Ohio, 44286. Email: mjamalab@cisco.com.*

Non-modularity based techniques are a category of community detection. Meyerhenke et al. [7] proposed high quality graph partitioning by using a parallel evolutionary algorithm to the coarsest graph. Recently, Qiao et al. [8] introduced approximate optimization to achieve parallel community detection for complex networks. Other works, such as [1, 1, 1, 9], have used non-modularity clustering objective functions to find the best graph clusters. Recently, Berahmand et al. [1] proposed a new fast local clustering approach called ECES which is based on the detection and expansion of core nodes through extended local similarity of nodes. Authors in [1, 1] propose clique percolation-based approaches [1]. However, the proposed methods are not feasible for very big data sets with more than 10^4 nodes.

Although modularity based techniques provide competitive results, some of them, such as [1, 1, 1], are not feasible approach for big data sets due to high computational complexity. Among existing methods, label propagation (LP) [2] introduced a high performance computationally efficient community discovery algorithm for large scale networks. In order to improve the efficiency of LP approach, several modified versions of LP have been introduced [2, 2, 2, 2]. The authors in [2] considered the quantity of labels, or the weighted quantity based on the adjacency matrix [2, 2] and label influence [2]. However, the best results (in terms of modularity) are presented where each label is evaluated by its impact on the modularity improvement [2, 2, 2]. Blondel et al. [2] proposes an extremely-fast high-performance unsupervised modularity-based clustering technique by applying LP in two iterative phases, known as the Louvain algorithm. Although the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2020.2995621, IEEE Transactions on Big Data

Louvain approach and all other modified versions [2, 2, 2, 3] propose robust performance, they still suffer from high computational complexity. In [2, 2, 2] a modified version was exploited for calculation of modularity gain achieved by label transition instead of its direct calculation, but the proposed formula is still complex for very large scale networks where millions of candidate labels should be evaluated.

Here, we propose a novel strategy for calculation of modularity gain associated with label transitions. The proposed approach is called Modularity Gain Acceleration (MGA), as it is inspired by analysis of the general closed form model of Newman's modularity shown at (2). Unlike the traditional method proposed in [2, 2, 2, 2, 3], here, a new formulation of the modularity gain objective function is developed corresponding to the available candidate labels. Exploiting mathematical manipulations, the proposed objective function is simplified into two components: the Local Sum-Weight (LSW) and the General Sum-Weight (GSW). The LSW is the lower complexity component and is calculated once per each *candidate* label transition, for each node. The GSW is the computationally complex component and is calculated only once per each label at the initiation phase. However, the GSW needs an updating procedure to keep up with network's community membership variations throughout the LP process. Therefore, an update scheme is conducted over the GSWs corresponding to the source and destination communities at each label transition. This update process requires only two additions, which leads to a huge efficiency gain compared to direct calculation of GSW per each label transition.

The efficiency of the proposed technique is evaluated analytically by examining the required mathematical operations and comparing that with traditional approaches [2, 2, 2, 2, 3]. Moreover, the efficiency of the proposed technique is evaluated over small to big real world data sets, some with millions of nodes, exploiting the proposed MGA in state-of-the-art LP-based community detection techniques such as Louvain [2] and traditional LP [2]. The obtained results show that the MGA produces the same strong performance in terms of modularity, as expected; however, MGA offers significant speed-up proportional to the size of network. Simulation results demonstrate that MGA-LP and MGA-Louvain outperform most existing modularity based clustering approaches in terms of both time complexity and modularity.

The rest of the paper is arranged as follows. Section 2 introduces the mathematical model of the community detection and discusses the LP method, including the traditional approach for so-called effective calculation of modularity gain variation. Then the proposed MGA technique is presented in Section 3 followed by an analytical evaluation of computational complexity associated with the proposed MGA and traditional approach. Section 4 presents experiments, analysis, and discussions, and finally Section 5 concludes the paper. Table 1 contains the selected symbols and notations used throughout this paper.

TABLE 1: Notation and symbols

Symbol	Description
G	graph $G = (V, E, \mathbf{W})$ of network including N nodes
$\mathbf{U}$	partition matrix of network (graph) G
i, j, n	indices of node or the column of partition matrix
l	index of community (label) or the row of partition matrix
k	index of cycle or iteration
$\mathbf{u}_i$	the ith column of the partition matrix $\mathbf{U}$
u_{li}	element at the lth row and ith column of $\mathbf{U}$
$\mathbf{U}_{li}$	partition matrix of G indicating l as label of node i
$\tilde{\mathbf{U}}_i$	partition matrix $\mathbf{U}$ with all zeros at the ith column
$\mathbf{W}$	adjacency matrix of network G
m_i	degree of the ith vertex
$\mathbf{m}$	degree vector $\mathbf{m} = [m_1, m_2, ..., m_N]^T$
$\mathbf{B}$	modularity matrix
$\mathbf{b}_i$	the ith column of modularity matrix $\mathbf{B}$
b_{li}	element at lth row and ith column of $\mathbf{B}$
$\mathcal{N}_i$	neighbor's of node i
$\mathcal{L}'_i$	the set of candidate labels for node i
ℓ_i	label (community) of node i
$\mathcal{L}_i$	$\mathcal{L}'_i \cup \ell_i$
$\mathbf{l}$	list of all available labels at the each cycle
c_k	number of available labels (communities) at the kth cycle(iteration)
$[N]$	the set of integers from 1 to N
$\setminus\{i\}$	remove ith element from the integer set

2 Community Detection

2.1 Community Detection and Modularity

Consider a network as an undirected graph $G = (V, E, \mathbf{W})$, where V is a set of N vertices (nodes), E is a set of edges, and $\mathbf{W}$ is an $N \times N$ adjacency matrix. Here, the ith row and jth column of $\mathbf{W}$, w_{ij}, denotes the weight associated with the edge connecting vertices i and j, for $i, j = [N]$ [1]. The process of community detection aims to find a $c \times N$ partition matrix $\mathbf{U}$, where the element in the kth row and ith column of $\mathbf{U}$, u_{ki}, for $k = [c]$ and $i = [N]$, represents the membership of the ith vertex in the kth community. Crisp partitions are $\mathbf{U}$, such that $u_{ki} \in \{0, 1\}$ and $\sum_{k=1}^{c} u_{ki} = 1$.

The modularity value Q, is a metric to evaluate the correctness of an associated community [3] represented by a partition $\mathbf{U}$. It was by Newman and Girvan [3] as a metric to evaluate quality of non-overlapping communities in graph clustering, and is defined as

$$\mathcal{Q} = \frac{1}{\|\mathbf{W}\|} \sum_{k=1}^{c} \sum_{i=1,j=1}^{N} \left(w_{ij} - \frac{m_i m_j}{\|\mathbf{W}\|} \right) \delta(i, j), \qquad (1)$$

where $m_i = \sum_{j=1}^{N} w_{ij}$, $i = [N]$, $\|\mathbf{W}\| = \sum_{i=1}^{N} m_i$. $\delta(i, j) = 1$ if vertex i and vertex j are in the same community, else $\delta(i, j) = 0$. Liu et al. [1] introduced a new modularity objective function for overlapping community detection in networks. Havens et al. [3] is developed a more generalized modularity metric, given at (2), that works for evaluating either overlapping or non-overlapping partitions.

$$\mathcal{Q} = \frac{tr\left(\mathbf{U}\mathbf{B}\mathbf{U}^T\right)}{\|\mathbf{W}\|}, \qquad (2)$$

where $\mathbf{B} = \left[\mathbf{W} - \frac{\mathbf{m}^T\mathbf{m}}{\|\mathbf{W}\|}\right]$ is the modularity matrix, $\mathbf{m} = (m_1, m_2, \ldots, m_N)^T$, and $m_i = \sum_{j=1}^{N} w_{ij}$, $i = [N]$. In this paper we exploit (2) to develop a new approach for calculation of modularity gain achievable by a label transition in LP-based community detection.

2.2 Label Propagation Clustering

The LP algorithm starts with an initialization phase where each vertex in the graph is allocated a unique label representing its community. Hence, for graph $G = (V, E)$, there would be N unique labels at the initialization step. Then, the main body of the LP algorithm starts with an iterative process where at each

1. Note that the notation $[N]$ means the set of integers from 1 to N.

iteration all labels of the graph vertices are updated. The LP approach selects the best label (among all available labels in a node's neighborhood) with respect to the best gain in term of modularity value that can be obtained for each candidate label transition. This iterative process will be continued until no further improvement in modularity gain. This demands evaluation gain corresponding to every candidate label. In very large scale networks, numerous numbers of label transitions have to be evaluated at each iteration, which leads to huge computational complexity.

Consider traditional modularity gain

$$\Delta Q\left(i, p \to q\right) = \left[\frac{\Sigma_{in} + k_{i,in}}{\|\mathbf{W}\|} - \left(\frac{\Sigma_{tot} + k_i}{\|\mathbf{W}\|}\right)^2\right] + \left[\frac{\Sigma_{in}}{\|\mathbf{W}\|} - \left(\frac{\Sigma_{tot}}{\|\mathbf{W}\|}\right)^2 - \left(\frac{k_i}{\|\mathbf{W}\|}\right)^2\right], \tag{3}$$

where Σ_{in} is the sum of the weights between nodes labeled q, Σ_{tot} is the sum of the weights corresponding to nodes labeled q, k_i is the sum of the weights of node i, $k_{i,in}$ is the sum of the weights of the links from node i to nodes labeled q and $\|\mathbf{W}\|$ is defined at (1). Recently, a modified version of modularity gain was proposed [2]

$$\Delta Q\left(i, p \to q\right) = \frac{\sigma\left(i, q \backslash \{i\}\right) - \sigma\left(i, p \backslash \{i\}\right)}{\|\mathbf{W}\|} + \frac{\left(\Sigma\left(p \backslash \{i\}\right) - \Sigma\left(q \backslash \{i\}\right)\right) v_i}{2\|\mathbf{W}\|^2}, \tag{4}$$

where $\Delta Q\left(i, p \to q\right)$ is the acquired modularity gain by relabeling the ith node from p to q, and $\sigma\left(i, p\right) = \sum_{i,j:\ell(j)=p} w_{i,j}$, and $\Sigma\left(q\right) = \sum_{\forall i \in \nu_q} v_i$ for $v_i = \sum_{i,j:j\in\mathcal{N}_i} w_{i,j} + 2w_{i,i}$, where ν_q represents the set of all nodes labeled as q.

Considering (3) or (4), the objective function for selecting the best label for the ith vertex at the kth iteration of the LP algorithm is

$$\ell_i^{(k+1)} = \arg\max_{\ell_j} \left\{\Delta Q\left(i, \ell_i^{(k)} \to \ell_j\right)\right\}, \forall j \in \mathcal{N}_i, \tag{5}$$

where $\Delta Q\left(i, p \to q\right)$ is defined at (3) or (4) and $\ell\left(j\right)$ and $\mathcal{N}_i$ represent the label of the jth vertex and the set of neighbors of the ith node, respectively.

In this work, we introduce an efficient approach called MGA which selects the best available label among the labels of neighbors, according to the new modularity gain of label transitions.

3 Modularity Gain Acceleration

In this section the proposed MGA approach is explained in detail. First, the objective function corresponding to the attained modularity gain by label transition is proposed. Then the computational complexity of the proposed objective function is studied analytically.

3.1 The MGA Approach

Consider a label transition of the ith node from current label ℓ_i to the new label ℓ_j based on (2); the modularity gain ΔQ is obtained by

$$\ell_i^{(k+1)} = \arg\max_{\ell_j} \left\{\sum_{n\in\mathcal{N}_i} u_{\ell_j n} w_{ni} - \frac{m_i}{\|\mathbf{W}\|} \sum_{n\backslash\{i\}} u_{\ell_j n} m_n\right\}, \ell_j \in \mathcal{L}_i, \tag{6}$$

where the first sum, $S_{1,\ell_j,i} = \sum_{n\in\mathcal{N}_i} u_{\ell_j n} w_{jn}$, represents the LSW, which aggregates the edge weights corresponding to those nodes of the neighbors of the ith node labeled as ℓ_j. That is a low computationally complex process which requires search over $|\mathcal{N}_i|$ elements and $(|\{k \in \mathcal{N}_i, \text{ and } \ell_k = \ell_j\}| - 1)$ summations per each candidate label for each node. However, the second sum, $S_{2,\ell_j,i} = \sum_{n\backslash\{i\}} u_{\ell_j n} m_n$, called the GSW, is a significant time consuming process as it requires search over all nodes to find those labeled as ℓ_j, then summation of their corresponding m which requires to search over N elements and $(|\{k \in [N], \text{ and } \ell_k = \ell_j\}| - 1)$ summations per each candidate label for each vertex. Here, we propose to exploit the pre-calculated values of GSWs or $S_{2,\ell_j,i}$ for all available labels. However, $S_{2,\ell_j,i}$ depends on either ℓ_j and i which makes it very computationally complex and also expensive to save for large scale networks.

Here, we propose to use simple mathematical manipulations as follow to remove the node index subscript i for the proposed GSW. To this end, we need to remove the element excluding notation $(\backslash\{i\})$ from $S_{2,\ell_j,i}$ or $\sum_{n\backslash\{i\}} u_{\ell_j n} m_n$. This does not affect $S_{2,\ell_j,i}$ for $\ell_j \neq \ell_i$ as $u_{\ell_j i} = 0$. However, for $\ell_j = \ell_i$, the impact of one additional i which is added by removing the element excluding notation $(\backslash\{i\})$ from $S_{2,\ell_j,i}$, must be subtracted from the modified $S_{2,\ell_j,i}$. Thus, the modularity objective gain function can be simplified to

$$\ell_i^{(k+1)} = \arg\max_{\ell_j} \begin{cases} S_{1,\ell_j,i} - \frac{m_i}{\|\mathbf{W}\|} S_{2,\ell_j}, & \forall \ell_j, j \in \mathcal{N}_i, \\ S_{1,\ell_j,i} - \frac{m_i}{\|\mathbf{W}\|}\left(S_{2,\ell_j} - m_i\right), & \ell_j = \ell_i, \end{cases} \tag{7}$$

where

$$S_{1,\ell_j,i} = \sum_{n\in\mathcal{N}_i} u_{\ell_j n} w_{jn}, \tag{8a}$$

$$S_{2,\ell_j} = \sum_{n} u_{\ell_j n} m_n. \tag{8b}$$

As mentioned, (8b) is only computed once at the initialization stage. However, by propagating labels throughout the LP process, the membership of vertices are subjected to change. Therefore, the pre-calculated values of GSWs proposed at (7) are no longer valid. This problem is addressed by an efficient updating stage,that occurs which compensates for the impact of each label transition through the procedure. Considering label transition of the ith node from the ℓ_ith to the ℓ_jth community, the following update rules must be applied to the pre-calculated GSWs corresponding to ℓ_i and ℓ_j

$$S_{2,\ell_j} + m_i \to S_{2,\ell_j}, \tag{9a}$$

$$S_{2,\ell_i} - m_i \to S_{2,\ell_i}, \tag{9b}$$

where S_{2,ℓ_i} and S_{2,ℓ_j} represent the GSWs corresponding to the old (ℓ_i) and the new (ℓ_j) labels associated to the ith node respectively, and m_i is defined at (1).

The proposed equations at (7) and (8) along with the updating equations at (9) present the main contribution of this paper. The main novelty of this work is reforming the GSW such that the node subscripts are removed, which allows off-line calculation of the GSW per each label (S_{2,ℓ_j}) followed by an update process instead of on-line calculation of the GSW for all available labels of all nodes $(S_{2,\ell_j,i})$. Algorithms 1 and 2 detail the process of deploying the MGA technique into the original LP [2] and Louvain [2] algorithms, respectively.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2020.2995621, IEEE Transactions on Big Data

Algorithm 1: MGA Label Propagation (MGA-LP)

Require: adjacency matrix $\mathbf{W}$; initial no. of communities c_1
return label list $\mathbf{l}$
initialize vertices communities $\ell = 1, 2, ..., c_l$
$S_{2,\ell} = \sum_n u_{\ell n} m_n$, using initialized communities.
while $Q_{new} > Q_{old}$ **do**
  for $i = 1, 2, ..., N$ **do**
    for $j \in \mathcal{N}_i$ **do**
      if ℓ_j is equal to ℓ_i **then**
        $\Delta Q_{\ell_j} = \sum_{n \in \mathcal{N}_i} u_{\ell_j n} w_{ni} - \frac{m_i}{\|\mathbf{W}\|} \left(S_{2,\ell_j} - m_i\right)$
      else
        $\Delta Q_{\ell_j} = \sum_{n \in \mathcal{N}_i} u_{\ell_j n} w_{ni} - \frac{m_i}{\|\mathbf{W}\|} S_{2,\ell_j}$
      end if
    end for
    Update $\mathbf{l}(i) = \arg\max_{\ell_j} \{\Delta Q_\ell\}$,
    $S_{2,\ell_j} + m_i \rightarrow S_{2,\ell_j}$
    $S_{2,\ell_i} - m_i \rightarrow S_{2,\ell_i}$
  end for
end while

Algorithm 2: MGA Louvain Algorithm (MGA-Louvain)

Require: adjacency matrix $\mathbf{W}$;
return label list $\mathbf{l}$
Initialize each node as single community and set $N_c = N$
while $Q_{new} > Q_{old}$ **do**
  for $\ell = 1, 2, ..., N_c$ **do**
    $S_{2,\ell} = \sum_n u_{\ell n} m_n$, using initialized communities.
  end for
  for $i = 1, 2, ..., N_c$ **do**
    for $j \in \mathcal{N}_i$ **do**
      if ℓ_j is equal to ℓ_i **then**
        $\Delta Q_{\ell_j} = \sum_{n \in \mathcal{N}_i} u_{\ell_j n} w_{ni} - \frac{m_i}{\|\mathbf{W}\|} \left(S_{2,\ell_j} - m_i\right)$
      else
        $\Delta Q_{\ell_j} = \sum_{n \in \mathcal{N}_i} u_{\ell_j n} w_{ni} - \frac{m_i}{\|\mathbf{W}\|} S_{2,\ell_j}$
      end if
    end for
    Update $\mathbf{l}(i) = \arg\max_{\ell_j} \{\Delta Q_\ell\}$,
    $S_{2,\ell_j} + m_i \rightarrow S_{2,\ell_j}$
    $S_{2,\ell_i} - m_i \rightarrow S_{2,\ell_i}$
  end for
  Set N_c with the number of available communities
  Construct supper nodes for $i = [N_c]$
  Set each supper node as single community
end while

3.2 Computational Complexity Analysis

In this section the number of mathematical operations required for calculation of modularity gain variations associated with a label transition is evaluated analytically. The traditional approaches, proposed at (3) or (4), and the proposed MGA scheme are evaluated. Table 2 reviews the number of operations including summation, multiplication, and search required to calculate a modularity gain at (3) or (4), and the MGA approach proposed at (7). In (4), Σ_{in} and $k_{i,in}$ are the most complex components. First, N search operations must be applied to extract a set of nodes labeled as q, or ν_q, such that $\{\ell_j = q,\ \forall j \in \nu_q\}$. Then $k_j = |\mathcal{N}_j|$ searches per each node in ν_q are needed to reveal the set of weights corresponding to the neighbors labeled q, or $w^{(q)}_{j,j'}$, such that $\{j, j' \in \nu_q,\ j' \in \mathcal{N}_j\}$. Then $\left(\left|\mathcal{N}^{(q)}_j\right| - 1\right)$ summations are needed to aggregate weights in $w^{(q)}_{j,j'}$, $\forall j \in \nu_q$. Therefore, $\sum_{j \in \nu_q} |\mathcal{N}_j|$ searches and $\sum_{j \in \nu_q} \left(\left|\mathcal{N}^{(q)}_j\right| - 1\right)$ summations are needed to calculate Σ_{in}. Moreover, calculation of $k_{i,in}$ requires $k_i = |\mathcal{N}_i|$ search operations to reveal the set of weights corresponding to the neighbors labeled q, or $w^{(q)}_{i,i'}$, such that $\{i, i' \in \nu_q,\ i' \in \mathcal{N}_i\}$. Then, $\left(\left|\mathcal{N}^{(q)}_i\right| - 1\right)$ summations are needed to aggregate weights in $w^{(q)}_{i,i'}$. Therefore, overall $k_i = |\mathcal{N}_i|$ search operations and $\left(|\mathcal{N}^{(q)}_i| - 1\right)$ summations are needed to calculate $k_{i,in}$. Furthermore, (Σ_{tot} requires $|\nu_q| - 1$ summations to aggregate total weights of nodes labeled q. The rest of components, such as k_i and $\|\mathbf{W}\|$, are constant values and can be calculated off-line. Additionally, six summations and three multiplications are need to calculate the final value of ΔQ at (3).

TABLE 2: Required mathematical operations for calculation of modularity gain variations to move the ith node into the qth community.

Operation	traditional (3) or (4)	MGA (7)
Search	$\sum_{j \in \nu_q} \|\mathcal{N}_j\| + \|\mathcal{N}_i\| + N$	$\|\mathcal{N}_i\|$
Summation	$\sum_{j \in \nu_q} \left\|\mathcal{N}^{(q)}_j\right\| + \left\|\mathcal{N}^{(q)}_i\right\| + 4$	$\left\|\mathcal{N}^{(q)}_i\right\| + 4$
Multiplication	3	2

Taking a closer look at (4), it is observed that the modularity gain value is derived with respect to the same components at (3). The $\Sigma(i, q \backslash \{i\})$ component at (4) represents the sum of weights among nodes in ν_q, or Σ_{in} at (3). Moreover, the $\sigma(i, q \backslash \{i\})$ component at (4) represents the weights between the ith node and nodes in ν_q, or $k_{i,in}$ at (3). However, the most complex part at (7) is the LSW or $S_{1,\ell_j,i}$ which like the $k_{i,in}$, demands $k_i = |\mathcal{N}_i|$ search operations and $\left(|\mathcal{N}^{(q)}_i| - 1\right)$ summations.

As shown in Table 2, traditional approaches, such as (3) or (4), have overall computational complexity depends on the size of the network N and the size of each community $|\nu_q|$, which usually increases with the size of network. However, for the proposed MGA approach, the overall computational complexity that is in the order of node neighborhood size, $|N_i|$, which usually depends on network topology rather than its size. Practical evaluation of traditional approaches and MGA for the real-world data sets are presented next.

4 Experimental Results and Discussion

Experimental results are conducted to investigate the performance of the proposed MGA technique in terms of computational complexity efficiency. It should be noted that as the proposed MGA leads to the same modularity gain as the traditional method and the same final modularity. Therefore, here the final network topology (in terms of Newman's modularity) is evaluated beside the computational complexity (in terms of processing time). The evaluation process is conducted for the classic LP [2] and the Louvain [2] techniques as two LP-based community detection algorithms by the proposed MGA (Algorithms 1 and 2). In order to assess the quality of the proposed technique on real-world data sets versus a-state-of-the-art approach, we present the most recent no-LP algorithm, called ECES [1].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2020.2995621, IEEE Transactions on Big Data

TABLE 3: Network characteristics and parameters used in MGA

Network	Nodes	Edges	c_1 (LP)	GT
Dolphin [34]	62	159	50	Yes
Football [35]	115	613	100	Yes
Jazz [36]	198	2742	100	No
Metabolic [31]	453	4,596	100	No
Email [37]	1,133	5451	100	No
Ego-Facebook [38]	4,039	88,234	100	No
Email-Enron [39]	36,692	183,831	2000	No
Com-DBLP [39]	317,080	1,049,866	2000	Yes
Com-Youtube [39]	1,134,890	2,987,624	10,000	Yes
as-Skitter [40]	1,696,415	11,095,298	10,000	No
Live-Journal [41]	3,997,962	34,681,189	20,000	No

Table **??** shows the characteristics of the real-world networks, such as the number of nodes and edges, the number of initial community c_1 (only for MGA-LP), and whether the network has available ground truth (GT) or not. Here, different sizes of networks are selected to explore the performance and the scalability of the proposed technique compared to the state-of-the-art methods. The experiments are executed 100 times per each network on the same machine to develop a fair comparison. The experiments are coded and executed in MATLAB, using a laptop with an i7-6560U processor @2.20GHz with 16GB of memory. Code is available online at [33]. Note that ECES [13] is implemented in ANSI C++ using a PC with core i5 CPU (2.8 GHz) and a 6.0-GB of memory.

Table **??** shows the average convergence time over 100 runs of the traditional LP and Louvain approaches versus the proposed MGA-LP and MGA-Louvain methods in Algorithm 1 and 2, respectively. Moreover, the results of the ECES [13] technique are presented to compare the classic LP and Louvain exploiting the MGA with a state-of-the-art technique.

Figure depicts the learning curves of the traditional LP [20] versus MGA-LP proposed in Algorithm 1 for small data sets such as Dolphin [34], Football [35], Jazz [36], Metabolic [31], and Email [37]. Moreover, Fig. depicts the learning curves of the Louvain [26] versus MGA-Louvain proposed in Algorithm 2 for the same data sets. As expected, and as shown in Table **??**, the modularity values obtained at each iteration are the same; however, the MGA-enabled learning curve converges to the final solution much faster than traditional LP and Louvain methods. It is also observed that the difference between the convergence time—i.e., the speed-up—increases proportional to the size of the data set. That is, we obtain better speed-up with larger data sets.

Figure **??**LP versus MGA-LP proposed in Algorithm 1 for large data sets such as Ego-Facebook [38], Email-Enron, com-DBLP, Com-YouTube [39], Skitter [40], and Live-Journal [41]. Moreover, Fig. depicts the learning curve of the Louvain [26] versus MGA-Louvain proposed in Algorithm 2 for the same data sets. As expected, and as shown in Table **??**, the modularity values at each iteration are the same, however, the MGA-enabled learning curve converges to the final solution much faster than the traditional LP and Louvain methods. Again, we see the trend that speed-up is proportional to the size of the data set.fig:largeNetLP depicts the learning curves of the traditional LP [20] versus MGA-LP proposed in Algorithm 1 for large data sets such as Ego-Facebook [38], Email-Enron, com-DBLP, Com-YouTube [39], Skitter [40], and Live-Journal [41]. Moreover, Fig.

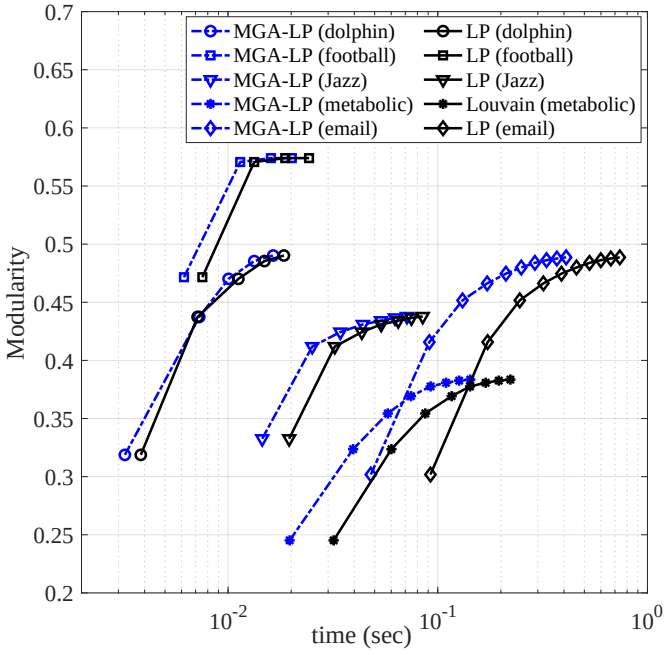


figure Modularity learning curves for Dolphin, Football, Jazz, Metabolic, and Email data sets comparing traditional LP [20] versus MGA-LP proposed in Algorithm 1

depicts the learning curve of the Louvain [26] versus MGA-Louvain proposed in Algorithm 2 for the same data sets. As expected, and as shown in Table **??**, the modularity values at each iteration are the same, however, the MGA-enabled learning curve converges to the final solution much faster than the traditional LP and Louvain methods. Again, we see the trend that speed-up is proportional to the size of the data set.The most interesting result of MGA is the linearity of computational complexity with the size of the network. This property shows that MGA is superior to the traditional approaches for big data sets. Moreover, it is observed that the proposed MGA-Louvain approach always outperform ECES in terms of time complexity and final modularity.

5 Conclusion

In this paper, we have introduced a novel objective function for calculation of the modularity gain corresponding to a community label transition of a node in a network. The computational complexity of the proposed technique is assessed analytically and compared with traditional approaches developed for the calculation of modularity gain. Then, two non-overlapping LP based community detection schemes incorporating traditional and the proposed MGA approach for modularity gain variations are applied to real-world data sets containing up to millions of nodes. The results on these real-world data sets validate the linear computational complexity of MGA with respect to network size. This opens a new era for all LP-based community detection techniques over very large data sets where available approaches can be prone to fail due to very high computational complexity. Moreover, application of the proposed MGA approach on

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2020.2995621, IEEE Transactions on Big Data

TABLE 4: Average processing time over 100 run in sec t_s and average Newman's modularity Q_m for real-world data set.

Algorithm	ECES [13]		LP [20]		MGA-LP		Louvain [26]		MGA-Louvain	
Network	t_s	Q_m	t_s	Q_m	t_s	Q_m	t_s	Q_m	t_s	Q_m
Dolphin	3	0.495	0.0184	0.4902	0.0164	0.4902	0.0255	0.5097	0.0162	0.5097
Football	5	0.549	0.0243	0.5740	0.0201	0.5740	0.105	0.604	0.0365	0.604
Jazz	5	0.291	0.0847	0.437	0.0713	0.437	0.152	0.443	0.0545	0.443
Metabolic	9	0.403	0.2219	0.3835	0.1428	0.3835	0.131	0.424	0.1085	0.424
Email	14	0.480	0.7381	0.4886	0.4094	0.4886	0.379	0.540	0.4073	0.540
Ego-Facebook	22	0.524	4.143	0.8086	1.255	0.8086	3.86	0.8323	1.232	0.8323
Email-Enron	150	0.517	2038.1	0.5531	22.69	0.5531	54.58	0.5845	9.499	0.5845
Com-DBLP	480	0.728	1,5361	0.6496	258.4	0.6496	1186.8	0.8099	296.2	0.8099
Com-YouTube	3240	0.569	91,769	0.6273	671.4	0.6273	21082	0.6987	2,589	0.6987
Skitter	–	–	156,060	0.6641	668.3	0.6641	40,915	0.837	5,236	0.837
Live-Journal	–	–	$> 10e6$	–	1,934	0.5973	327,560	0.7276	33,590	0.7276

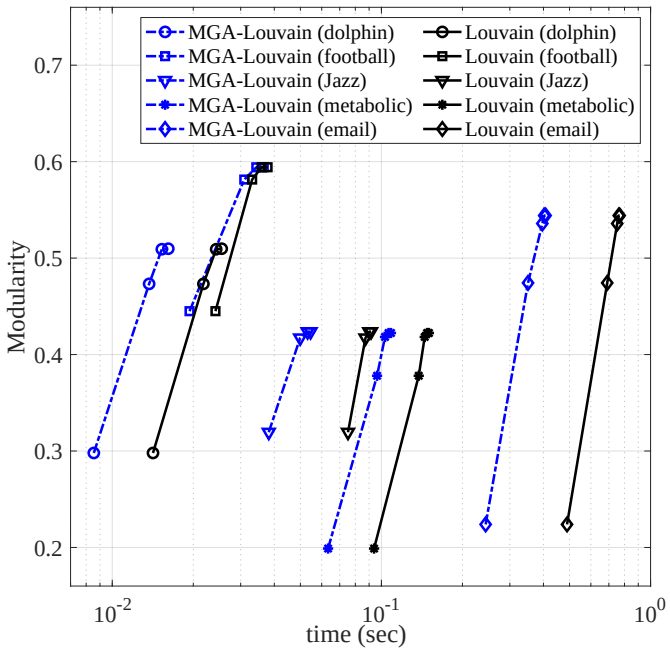
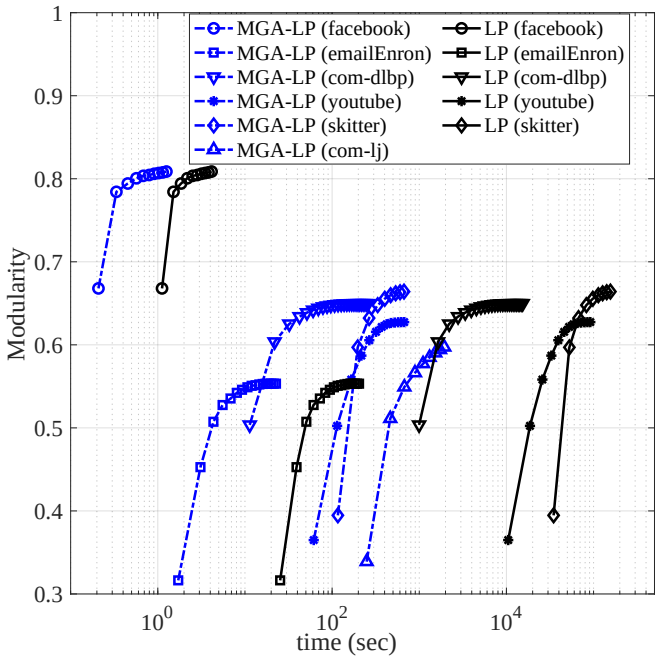


figure Modularity learning curve for Dolphin, Football, Jazz, Metabolic, and Email data sets comparing Louvain [26] versus MGA-Louvain proposed in Algorithm 2



figureModularity learning curve for Ego-Facebook, Email-Enron, com-DBLP, Com-YouTube, Skitter, and Live-Journal data sets comparing traditional LP [20] versus MGA-LP proposed in Algorithm 1

overlapping LP-based approaches can be considered, which is left for future study.

REFERENCES

[1] M. R. Garey, D. S. Johnson, and L. Stockmeyer, "Some simplified np-complete problems," in *Proceedings of the Sixth Annual ACM Symposium on Theory of Computing*, ser. STOC '74, 1974, pp. 47–63.

[2] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3, pp. 75–174, 2010.

[3] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, p. 026113, 2004, pRE.

[4] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E*, vol. 69, no. 6, p. 066133, pRE.

[5] T. Evans and R. Lambiotte, "Line graphs, link partitions, and overlapping communities," *Physical Review E*, vol. 80, no. 1, p. 016105, 2009.

[6] J. Hou Chin and K. Ratnavelu, "A semi-synchronous label propagation algorithm with constraints for community detection in complex networks," *Scientific Reports*, vol. 7, no. 45836.

[7] H. Meyerhenke, P. Sanders, and C. Schulz, "Parallel graph partitioning for complex networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 9, pp. 2625–2638, Sept 2017.

[8] S. Qiao, N. Han, Y. Gao, R. Li, J. Huang, J. Guo, L. A. Gutierrez, and X. Wu, "A fast parallel community discovery model on complex networks through approximate optimization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 9, pp. 1638–1651, Sept 2018.

[9] T. Cai and X. Li, "Robust and computationally feasible community detection in the presence of arbitrary outlier nodes," *arXiv preprint arXiv:1404.6000*, 2014.

[10] D. Jin, D. He, D. Liu, and C. Baquero, "Genetic algorithm with local search for community mining in complex networks," in *2010 22nd IEEE International Conference on Tools with Artificial Intelligence*, vol. 1, Oct 2010, pp. 105–112.

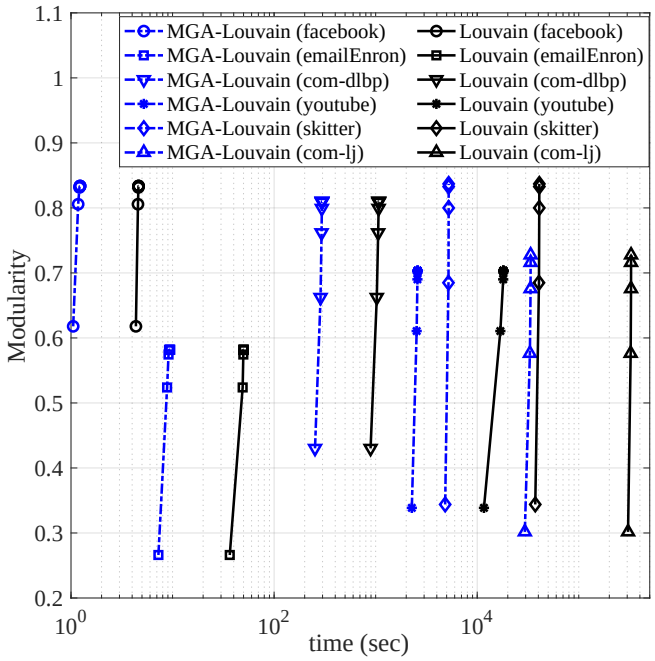
[11] L. Danon, A. Díaz-Guilera, and A. Arenas, "The effect of size heterogeneity on community identification in complex networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2006, no. 11, p. P11010, 2006.

[12] J. Liu, "Fuzzy modularity and fuzzy community structure in networks," *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 77, no. 4, pp. 547–557, 2010.

[13] K. Berahmand, A. Bouyer, and M. Vasighi, "Community detection in complex networks by detecting and expanding core nodes through extended local similarity of nodes," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 4, pp. 1021–1033, Dec 2018.

[14] Z. Lu, J. Wahlström, and A. Nehorai, "Community detection in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2020.2995621, IEEE Transactions on Big Data



figureModularity learning curve for Ego-Facebook, Email-Enron, com-DBLP, Com-YouTube, and Skitter data sets comparing Louvain [26] versus MGA-Louvain proposed in Algorithm 2

complex networks via clique conductance," *Scientific reports*, vol. 8, no. 1, p. 5982, 2018.

[15] C. Lee, F. Reid, A. McDaid, and N. Hurley, "Detecting highly overlapping community structure by greedy clique expansion," *arXiv preprint arXiv:1002.1827*, 2010.

[16] M. Wang, C. Wang, J. X. Yu, and J. Zhang, "Community detection in social networks: an in-depth benchmarking study with a procedure-oriented framework," *Proceedings of the VLDB Endowment*, vol. 8, no. 10, pp. 998–1009, 2015.

[17] J. Su and T. C. Havens, "Fuzzy community detection in social networks using a genetic algortihm," in *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, July 2014, pp. 2039–2046.

[18] Q. Dai, M. Guo, Y. Liu, X. Liu, and L. Chen, "Mlpa: Detecting overlapping communities by multi-label propagation approach," in *2013 IEEE Congress on Evolutionary Computation*, June 2013, pp. 681–688.

[19] S. Shi, Y. Chen, M. Fang, W. Li, and Shining, "A hierarchical multi-label propagation algorithm for overlapping community discovery in social networks," in *2014 11th Web Information System and Application Conference*, Sep. 2014, pp. 113–118.

[20] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E*, vol. 76, no. 3, p. 036106, 2007.

[21] X. Liu and T. Murata, "Community detection in large-scale bipartite networks," *Transactions of the Japanese Society for Artificial Intelligence*, vol. 25, no. 1, pp. 16–24, 2010.

[22] M. He, M. Leng, F. Li, Y. Yao, and X. Chen, *A Node Importance Based Label Propagation Approach for Community Detection*. Springer, 2014, pp. 249–257.

[23] J. Xie and B. K. Szymanski, "Community detection using a neighborhood strength driven label propagation algorithm," in *2011 IEEE Network Science Workshop*, June 2011, pp. 188–195.

[24] M. J. Barber and J. W. Clark, "Detecting network communities by propagating labels under constraints," *Physical Review E*, vol. 80, no. 2, p. 026129, 2009.

[25] Y. Xing, F. Meng, Y. Zhou, M. Zhu, M. Shi, and G. Sun, "A node influence based label propagation algorithm for community detection in networks," *The Scientific World Journal*, vol. 2014, 2014.

[26] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.

[27] C. L. Staudt and H. Meyerhenke, "Engineering parallel algorithms for community detection in massive networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 1, pp. 171–184, Jan 2016.

[28] P. D. Meo, E. Ferrara, G. Fiumara, and A. Provetti, "Generalized louvain method for community detection in large networks," *2011 11th International Conference on Intelligent Systems Design and Applications*, pp. 88–93.

[29] L. Speidel, T. Takaguchi, and N. Masuda, "Community detection in directed acyclic graphs," *The European Physical Journal B*, vol. 88, no. 8, p. 203, 2015.

[30] Y. Sun, B. Danila, K. Josić, and K. E. Bassler, "Improved community structure detection using a modified fine-tuning strategy," *EPL (Europhysics Letters)*, vol. 86, no. 2, p. 28004, 2009.

[31] J. Duch and A. Arenas, "Community detection in complex networks using extremal optimization," *Phys. Rev. E*, vol. 72, p. 027104, Aug 2005.

[32] T. C. Havens, J. C. Bezdek, C. Leckie, K. Ramamohanarao, and M. Palaniswami, "A soft modularity function for detecting fuzzy communities in social networks," vol. 21, no. 6, 2013, pp. 1170–1175.

[33] S. Yazdanparast and M. Jamalabdollahi, "Modularity gain acceleration." [Online]. Available: http://git.com/ayazdanp/

[34] D. Lusseau, K. Schneider, O. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations," *Behavioral Ecology and Sociobiology*, vol. 54, pp. 396–405, 2003.

[35] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, p. 78217826, 2002.

[36] . M. Gleiser and L. D. P, "Community structure in jazz," *Adv.Complex System*, p. 656573, 2003.

[37] R. Guimerà, L. Danon, A. Díaz-Guilera, F. Giralt, and A. Arenas, *Self-similar community structure in a network of human interactions*. American Physical Society, Dec 2003, vol. 68, p. 065103.

[38] J. Leskovec and J. J. Mcauley, "Learning to discover social circles in ego networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 539–547.

[39] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, Jun. 2014.

[40] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: densification laws, shrinking diameters and possible explanations," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, 2005, pp. 177–187.

[41] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," *Knowledge and Information Systems*, vol. 42, no. 1, pp. 181–213, 2015.