A Local Seed Selection Algorithm for Overlapping Community Detection

Farnaz Moradi, Tomas Olovsson, Philippas Tsigas
Department of Computer Science and Engineering
Chalmers University of Technology, Gothenburg, Sweden
Email: {moradi,tomasol,tsigas}@chalmers.se

***Abstract*—One of the widely studied structural properties of social and information networks is their community structure, and a vast variety of community detection algorithms have been proposed in the literature. Expansion of a seed node into a community is one of the most successful methods for local community detection, especially when the global structure of the network is not accessible. An algorithm for local community detection only requires a partial knowledge of the network and the computations can be done in parallel starting from seed nodes. The parallel nature of local algorithms allow for fast and scalable solutions, however, the coverage of the communities heavily depends on the seed selection. The communities identified by a local algorithm might cover only a subset of the nodes in a network if the seeds are not selected carefully.**

In this paper, we propose a novel *seeding algorithm* which is parameter free, utilizes merely the local structure of the network, and identifies good seeds which span over the whole network. In order to find such seeds, our algorithm first computes similarity indices from local link prediction techniques to assign a *similarity score* to each node, and then a *biased graph coloring* algorithm is used to enhance the seed selection. Our experiments using large-scale real-world networks show that our algorithm is able to select good seeds which are then expanded into high quality overlapping communities covering the vast majority of the nodes in the network using a personalized PageRank-based community detection algorithm. We also show that using our local seeding algorithm can dramatically reduce the execution time of community detection.

I. Introduction

The emergence of large-scale social and information networks have motivated numerous studies of the structural properties of these networks such as their community structure. A community typically refers to a group of densely connected nodes which have sparse connections with the rest of the nodes in the network, and a wide variety of algorithms have been proposed for identifying communities [8], [20].

Community detection algorithms can be divided into global and local algorithms. *Global algorithms* require a global knowledge of the entire structure of the network in order to uncover all the communities in that network. Since such knowledge might not be available for large-scale networks, local algorithms are gaining more popularity [5], [17], [19], [7]. *Local algorithms* typically start from a number of *seed* nodes (sets) and expand them into possibly overlapping communities by examining only the neighborhood of the seeds. Due to their nature, local algorithms can be parallelized and are scalable. However, they might only cover a subset of the nodes in a network if the seeds are not chosen carefully. A naive approach for achieving high coverage is therefore to consider all the nodes in a network as seeds. However, this approach is computationally expensive and leads to many redundant communities. Although the goal of local algorithms is not to achieve a complete coverage of a network, finding a small number of seeds which are well distributed over the network and can lead to a high coverage is very desirable.

Since our knowledge of the community structure of large-scale real-world networks is usually limited, finding good seeds that span over the network using only the knowledge of the local structure of a network is a challenging problem. In this paper, we present a novel local seed selection algorithm which achieves a high coverage and a community quality similar to the naive approach (where all nodes are used as seeds) but with a significantly lower execution time.

Our algorithm uses similarity indices from *link prediction* techniques. In link prediction, similarity indices are used to estimate the similarity of nodes which are expected to get connected, however, we use them to asses the similarity of nodes which are already connected. We assign a local *similarity score* to each node based on a similarity index and identify nodes that are similar to their neighbors and therefore are expected to be in the same community. Andersen et al. [2] theoretically showed that a seed set that is "nearly contained" in a target community is a good seed set for that community. We select a node as a seed if it has the highest score among its neighbors, and we show that this method is very effective in finding a small number of very good seeds in a network which can be expanded into high quality communities. However, similar to other existing local seeding algorithms, the communities expanded from these seeds do not achieve a high coverage of the network.

In order to improve the coverage, we propose to use distributed *graph coloring*. Although we show that we can select good seeds using graph coloring, we also introduce a new distributed *biased graph coloring* algorithm to further enhance our seeding algorithm, where the nodes with the highest local similarity score, which are expected to be good seeds, are assigned a specific color. Then the ties are broken at random so that no two adjacent nodes pick the same color. In the end, the nodes which received the specific color are selected as seeds. Our proposed algorithm is parameter free,

is computed locally, selects seeds from parts of the network where the other local similarity methods fail to pick any seeds, and does not lead to many duplicate communities since it does not pick any neighboring nodes as seeds.

The selected seeds are then expanded into overlapping communities using a personalized PageRank-based local community detection algorithm, which can be computed locally and is known to result in high quality communities [22]. We have empirically compared our proposed seeding algorithm with a number of existing seeding methods, as well as a state-of-the-art local community detection algorithm with respect to quality and coverage of the identified communities. The quality is assessed using ground truth data where such data exists, and *conductance* which is a widely used quality function.

Overall, our contributions in this paper are as follows.

- We define a similarity score which is calculated as the sum of the similarity of a node with all of its connected neighbor by adopting the similarity indices from link prediction techniques.
- We propose a new local seeding algorithm which uses these similarity scores (link prediction-based seeding).
- We propose to use graph coloring for picking random seeds in a network and introduce biased graph coloring for enhancing our seeding algorithm (biased coloring-based seeding).
- We empirically compare the different similarity indices which we have used in our seeding algorithm. We also experimentally evaluate our seeding algorithm and show that it can find a reasonably small number of seeds which are expanded into communities with high coverage and a similar quality compared to when all the nodes are used as seeds but with significantly reduced execution time.
- We show that our biased coloring algorithm is also successful in improving the coverage of other existing local seeding algorithms.

The remainder of the paper is organized as follows. Sections II and III present the related work and the background, respectively. Our seeding algorithm is presented in Section IV. Section V presents the experimental results. Finally, Section VI concludes our work.

II. Related Work

There have been numerous studies proposing different types of community detection algorithms [8], [20]. In this paper, we only consider local algorithms.

Coscia et al. [7] have proposed the *Demon* algorithm, which starts from all the nodes in a network to identify the local communities in each neighborhood and then uses merging to form the optimal global communities. A closely related approach is the *Node Perception* by Soundarajan et al. [17] which is a template for first finding local sub-communities and then identifying all the communities.

There are a variety of local community detection algorithms which assume that the seeds are given, e.g., [5] or can be picked at random, e.g., [11]. However, there are not many studies which have looked into the problem of selecting *good seeds*. Shen et al. [16] proposed to use maximal cliques, which form the core of the communities, as seeds which is computationally expensive. Gargi et al. [9] used the number of times a video has been viewed in the Youtube network to select the top videos as seeds, however, this type of non-structural information is not available for many networks.

Gleich et al. [10] showed that the *egonets* with low conductance are good seeds for finding the best communities of a network with respect to conductance. However, Whang et al. [19] showed that these communities do not achieve high coverage. Chen et al. [4] proposed an algorithm for selecting the nodes with local maximal degree as seeds. The authors suggested to remove the identified communities expanded from these seeds from the network and find new seeds in the remaining parts of the network repeatedly to improve the coverage. These methods are explained in more detail in the next section and are compared against our proposed seeding algorithm.

Whang et al. [19] have proposed two seeding algorithms which achieve high coverage. In the *Graclus centers* they run a partitioning algorithm to create k network partitions and then the nodes in the center of these partitions are selected as seeds. In the *spread hub* algorithm, at least k nodes with the highest degree in the network are selected as seeds. Both seeding algorithms require some global knowledge as well as the number of seeds to be known which is not a realistic assumption since we typically do not know the community structure of the real-world networks in advance.

Our seeding algorithm is parameter free and uses similarity indices from local link prediction and local graph coloring. Yan and Gregory [21] have used a similarity index to add edge weights to unweighted networks in order to improve the quality of existing global community detection algorithms. Psicologia et al. [6] have used simple graph coloring as the first step for a label propagation community detection algorithm. These works do not introduce local seeding algorithms and therefore are fundamentally different from our work.

Our algorithm can be used for seeding any local community detection algorithm. In this paper, we have used a variant of a personalized PageRank algorithm by Yang et al. [22]. Although Yang et al. have shown that this algorithm is very successful in identifying the communities to which a given seed belongs, they did not investigate the effect of using a seeding algorithm.

III. Background

A. Notations

Let $G = (V, E)$ be a connected, undirected, and unweighted graph, where V is the set of n nodes and E

is the set of m edges or links of G. Let $v \in V$ be a node in G. The set of the neighbors of v is denoted by $\Gamma(v) = \{u : u \in V, (u,v) \in E\}$. The degree of v is shown as $k_v = |\Gamma(v)|$, and Δ refers to the maximum degree in the graph. The egonet of v is the subgraph induced by the node and its neighbors and is defined as $egonet(v) = \{v\} \cup \{u : u \in \Gamma(v), (u,v) \in E\}$.

A local community detection algorithm expands a seed node s into a community C which is a set of nodes including s. We denote by $\mathcal{C} = \{C_1, \ldots, C_k\}$ the collection of overlapping communities expanded from k distinct seed nodes which are selected by a seeding algorithm. The *coverage* of the collection of communities $\mathcal{C}$ is defined as $cov(\mathcal{C}) = \frac{|\bigcup_{i=1}^{k} C_i|}{|V|}$. The *conductance* of a community, which is used both as a scoring function and as a quality function, is defined as $\phi(C) = \frac{\overline{m}(C)}{\min(vol(C), vol(V \setminus C))}$, where $\overline{m}(C) = |\{(u,v) \in E : u \in C, v \notin C\}|$ is the number of inter-cluster edges and $vol(C) = \sum_{v \in C} k_v$ is the volume of a community C and corresponds to the sum of the degree of all the nodes in the community.

B. Existing Seeding Methods

In this study, we have selected a number of state-of-the-art algorithms to be compared against our proposed algorithm.

Spread hub (SH) [19] In this method, first the nodes are sorted in order of decreasing degree. Then, as long as the number of selected seeds is less than k, the nodes with the maximum degree are greedily chosen as seeds. This algorithm can pick more than k seeds, where k is given as input, and only picks neighboring nodes as seeds when their degree is equal. The complexity of SH is $O(n \log n + k)$.

Low conductance cuts (EC) [10] Gleich et al. have shown that the low conductance egonets are good seed sets. This algorithm selects around 3% of the network nodes as seeds. A node v can be a seed if for all $u \in \Gamma(v)$, $\phi(egonet(v)) \leq \phi(egonet(u))$. EC can find these seeds with time complexity $O(m\Delta)$. Whang et al. [19] showed that this method performs poorly with respect to coverage.

Local maximal degree (MD) [4] This algorithm uses a list of nodes in the graph. If a node has the highest local degree, it is added to a seed set and is removed from the list together with all its neighbors with lower degrees. If a node is not a *local-maximal-degree* node, it is also removed from the list. This process is repeated until all the nodes are removed from the list. The complexity of MD is $O(n\Delta)$.

C. Link Prediction and Similarity Indices

Link prediction is the problem of predicting the relations that should exist in a network or are very likely to be formed in the future. These methods typically estimate the similarity of nodes which are not connected to each other using similarity indices. We have selected a number of basic and widely used similarity indices for local link prediction [12].

Neighbors index (CN) is a very basic metric which calculates the size of the neighborhood overlap of two nodes and is formally defined as $CN(u,v) = |\Gamma(u) \cap \Gamma(v)|$.

Hub promoted index (HP) assigns higher scores to the edges adjacent to high degree nodes (*hubs*) and is defined as $HP(u,v) = CN(u,v)/min(k_u, k_v)$.

Leicht-Holme-Newman index (LHN) assigns high values to the nodes that have many common neighbors compared to the expected number of neighbors and is defined as $LHN(u,v) = CN(u,v)/(k_u \times k_v)$.

Resource Allocation index (RA) is motivated by the resource allocation process where the common neighbors of two nodes act like transmitters which distribute their recourses to all their neighbors. Therefore, the amount of recourses a node u receives from a node v can be used for calculating their similarity as $RA(u,v) = \sum_{w \in \Gamma(u) \cap \Gamma(v)} \frac{1}{k_w}$.

Preferential Attachment (PA) is motivated by the preferential attachment mechanism, where the probability that a new link is connected to a node v is proportional to the degree of the node k_v and is defined as $PA(u,v) = k_u \times k_v$.

D. Graph Coloring

The problem of coloring the nodes of a graph with a small number of colors is a fundamental graph problem and has been widely studied. The goal of a graph coloring algorithm is to color the nodes in a graph with at most $\Delta + 1$ colors, where Δ is the maximum degree in the graph, so that no two neighboring nodes share the same color. Coloring has many applications such as assigning time or frequency slots for communications of wireless devices.

The most well-known distributed algorithm for $\Delta + 1$ graph coloring is a randomized algorithm based on the maximum independent set algorithm of Luby [13], [14] which needs $O(\log n)$ time. Barenboim et al. [3] have shown that deterministic distributed coloring can be implemented in linear $O(\Delta)$ time.

In distributed graph coloring, each node picks a color uniformly at random from the set of colors which are available to it, and solves the conflicts with its neighbors by picking new colors and exchanging confirmations. Eventually, the algorithm converges when each node has a color different from the colors of all its neighbors.

IV. Our Method

In this section we present our approach to overlapping community detection using our novel seeding algorithm and a personalized PageRank-based seed expansion algorithm.

A. Link Prediction-based Seed Selection

In our seeding algorithm, we propose to use similarity indices from link prediction methods to calculate the similarity of the nodes which are directly connected. Our intuition is that if a node has high similarity with its neighbors, it is expected that they belong to the same community. Moreover,

Algorithm 1 Link prediction-based seed selection

Input: A graph $G(V, E)$.
Output: The seed set S.
Let $S = \emptyset$;
for all $v \in V$ **do** *score*$(v) = \sum_{u \in \Gamma(v)}$ *sim*(u, v); **end for**
for all $v \in V$ **do**
  if *score*$(v) > 0$ **and** $\forall u \in \Gamma(v)$: *score*$(v) \geq$ *score*(u) **then**
    $S = S \cup \{v\}$;
  end if
end for
return S

Algorithm 2 Biased coloring-based seed selection

Input: A graph $G(V, E)$.
Output: The seed set S.
Let $S = \emptyset$;
for all $v \in V$ **do** *score*$(v) = \sum_{u \in \Gamma(v)}$ *sim*(u, v); **end for**
for all $v \in V$ **do**
  Let $SC = \emptyset$;
  $\forall u \in \Gamma(v)$, *confirm*$(u, v) = 0$; *converge*$(v) = false$; *color*$(v) = 0$;
  available_colors(v)=$\{c_1, ..., c_{k_v+1}\}$ where $k_v = |\Gamma(v)|$;
  $SC = \{$*score*$(u) : \forall u \in egonet(v)\}$;
  for all $u \in egonet(v)$ **do**
    if *score*$(u) = max(SC)$ **then** $color(u) = c_1$; **end if**
  end for
  if *color*$(v) = 0$ **then** *color*(v) = *pick_color*(*available_colors*(v)); **end if**
  while *converge*$(v) = false$ **do**
    for all $u \in \Gamma(v)$ **do**
      if *color*(v)=*color*(u) **and** *score*$(v) \leq$ *score*(u) **then**
        color(v) = *pick_color*(*available_colors*(v));
      else if *color*$(u) > 0$ **then** *confirm*$(u, v) = 1$; **end if**
    end for
    if $\forall u \in \Gamma(v)$, *confirm*$(u, v) = 1$ **and** *color*$(v) > 0$ **then**
      converge$(v) = true$;
    end if
  end while
  if $color(v) = c_1$ **and** $k_v > 1$ **then** $S = S \cup \{v\}$; **end if**
end for
return S

a node is a good seed if it has many neighbors in the target community [2]. Therefore, a node which is very similar to its neighbors can be a good representative for its neighborhood, thus can be selected as a seed for local community detection.

Our seed selection algorithm is presented in Algorithm 1. Each node v calculates its similarity with its direct neighbors and assigns a *score*(v) to itself based on the sum of the similarities. The *sim*(u, v) function refers to any of the similarity indices introduced in the previous section. Then, each node compares its score with its neighbors and decides if it is a seed or not.

Table I shows a summary of the names we use in the rest of the paper for the instances of our seeding algorithm when different similarity indices are used for calculating the score of the nodes.

B. Biased Coloring-based Seed Selection

Although our proposed seeding algorithm using similarity scores can be used on its own for seed selection, we propose to enhance it by adopting a graph coloring algorithm. Coloring helps us to pick seeds that are better distributed over the network and therefore can lead to improved coverage. First, we propose a basic random coloring method for seed selection based on the randomized distributed coloring algorithm of Luby [14].

Table I: Summary of the names used for the instances of our seed selection algorithm based on the similarity indices being used.

	Similarity index $sim(u, v)$	Instance name
Link prediction-based Seeding (Algorithm 1)	$CN(u, v)$	CN
	$HP(u, v)$	HP
	$LHN(u, v)$	LHN
	$RA(u, v)$	RA
	$PA(u, v)$	PA
Biased coloring-based Seeding (Algorithm 2)	$CN(u, v)$	CN + coloring
	$HP(u, v)$	HP + coloring
	$LHN(u, v)$	LHN + coloring
	$RA(u, v)$	RA + coloring
	$PA(u, v)$	PA + coloring
Random coloring	-	RN (coloring)

Random Coloring (RN) can be directly used for selecting seeds, by picking the nodes which have the same color, for example color c_1. The RN seed selection has some advantages over simply picking seeds at random. It does not require the number of seeds to be picked to be known and it does not pick two neighbors as seeds resulting in fewer redundant communities.

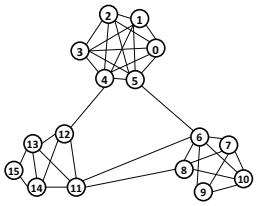
Although basic random coloring can be used for seed selection, we also propose a biased graph coloring algorithm which favors the nodes with high similarity scores to improve the seed selection. The main difference between the biased and the basic coloring is that, in biased coloring, the nodes which are expected to be better seeds with respect to link prediction-based similarity scores pick a specific color, but in basic coloring, random nodes get the specific color.

Algorithm 2 shows our enhanced seeding algorithm with our biased graph coloring. First each node v calculates its score using a local similarity function, and then assigns the color c_1 to the nodes with the highest score in its egonet, $egonet(v)$. If a node has not received the color c_1 from itself or any of its neighbors, it picks a color for itself at random from the set of available colors. In other words, if a node has the highest score in at least one neighborhood it gets the color c_1, otherwise, it picks a random color. After initialization, each node checks the color of its neighbors, if there is no conflict, the color is confirmed. Otherwise, if the score of the node is less than or equal to the score of its conflicting neighbor, the node picks a new color uniformly at random using *pick_color*. This makes sure that the nodes with high scores preserve their original color c_1.

The algorithm converges when all the nodes in the network have a confirmed color. After convergence, the nodes which have the color c_1 are selected as the seeds, since these nodes have the highest similarity score in their neighborhood and are expected to be good seeds.

Figure 1 shows two scenarios where coloring dramatically improves seed selection[1]. Figure 1a shows an example where three densely connected communities exist and therefore it

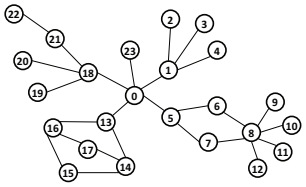
[1]In practice, due to the randomness in the coloring, the selected seeds are not deterministic. In our experiments section we discuss this topic further.



Method	Seeds
CN	0, 1, 2, 3, 4, 5, 13, 14
HP	0, 1, 2, 3, 13, 14
LHN	0, 1, 2, 3, 7, 13, 14
RA	4, 5, 13, 14
PA	5
MD [4]	4, 5, 6
EC [10]	0, 1, 2, 3, 7, 10, 13, 14
SH(k=3)	4, 5, 6

Method	Seeds
CN + coloring	5, 7, 14
HP + coloring	3, 6, 14
LHN + coloring	3, 10, 14
RA + coloring	5, 8, 14
PA + coloring	5, 8, 14
MD + coloring	4, 6, 14
EC + coloring	0, 9, 13
SH + coloring	0, 6, 14
RN (coloring)	3, 7, 13

(a)



Method	Seeds
CN	-
HP	-
LHN	-
RA	-
PA	0, 8
MD [4]	0, 8, 14, 16
EC [10]	1, 8, 18
SH(k=4)	0, 8, 14, 16

Method	Seeds
CN + coloring	5, 15, 21
HP + coloring	6, 13, 15, 21
LHN + coloring	6, 15, 17, 18
RA + coloring	1, 8, 16, 21
PA + coloring	7, 14, 16, 21
MD + coloring	0, 8, 15, 21
EC + coloring	5, 14, 16, 21
SH + coloring	0, 8, 14, 16, 21
RN (coloring)	6, 15, 21

(b)

Figure 1: Example graphs and the selected seeds using different methods. Biased coloring improves the seed selection.

is expected that a good seed selection algorithm can pick at least one seed in each community. However, it can be seen that while PA only picks one seed, the others pick many seeds including neighboring nodes. For instance, SH (see Section III-B) which requires the number of seeds k to be known in advance, picks node 4, 5, and 6 which have the highest degree in the network but are directly connected. We can also see that by adding biased coloring, the seed selection improves. For instance, PA combined with coloring selects one seed from each community and the methods which earlier picked many neighbors, now pick fewer seeds which are better distributed across the network.

Figure 1b shows another example where the neighboring nodes do not have any common neighbors. Therefore, by using the common neighbor-based similarity indices, i.e., CN, HP, LNH, and RA, all the nodes get a similarity score of zero, so our algorithm fails to pick any seeds at all. However, the figure also shows that when adding biased coloring to the local seeding methods, a number of seeds are selected which are well distributed over the graph. In these scenarios, the biased coloring actually works similar to the random coloring, since a node will only receive color c_1 if it has picked it at random.

Time complexity: The time complexity of link prediction-based score calculation is $O(n\Delta)$. Our distributed biased coloring algorithm which is used for enhancing seeding is based on the algorithm by Luby which can run in $O(\log n)$.

C. Local Community Detection

After selecting the seeds, any type of seed expansion algorithm can be used to identify local communities. In this paper, we use a local algorithm by Yang et al. [22] which uses truncated random walks to approximate personalized PageRank. The main advantages of random walk-based techniques are that they can be computed locally and in parallel, the time and space requirements of such algorithms do not depend on the size of the network [2], and the communities identified by these types of algorithms are structurally close to real-world communities [1].

The algorithm by Yang et al. works as follows. First, the *PageRank-Nibble* algorithm of Andersen et al. [2] is used to compute an approximate personalized PageRank vector starting from the seed node.[2] Then, the algorithm by Spielman and Teng [18] is used to create a collection of sets of nodes. The set which has the first local optima of a scoring function is selected as the final community. The details of the algorithm can be found in [22], [2], [18]. In this study, we have used conductance as the scoring function which has been shown to be good for identifying ground truth communities [22].

Time complexity: The overall complexity of the local community detection algorithm can be approximated with $O(\sum_{i=1}^{k}(vol(C_i)))$, where k is the number of the seeds obtained from the seeding algorithm[3].

Table II: Summary of the networks

Dataset	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert\mathcal{C}_T\rvert^*$
Amazon [22]	334,863	925,872	151,037
DBLP [22]	317,080	1,049,866	13,477
Youtube [15]	1,134,890	2,987,624	8,385
LiveJournal [22]	3,997,962	34,681,189	287,512
SoundCloud	5,187,722	36,989,364	N/A

*the number of ground truth communities

V. Experimental Results

In this section, we evaluate and compare our local seeding algorithm with other existing algorithms using large scale real-world networks.

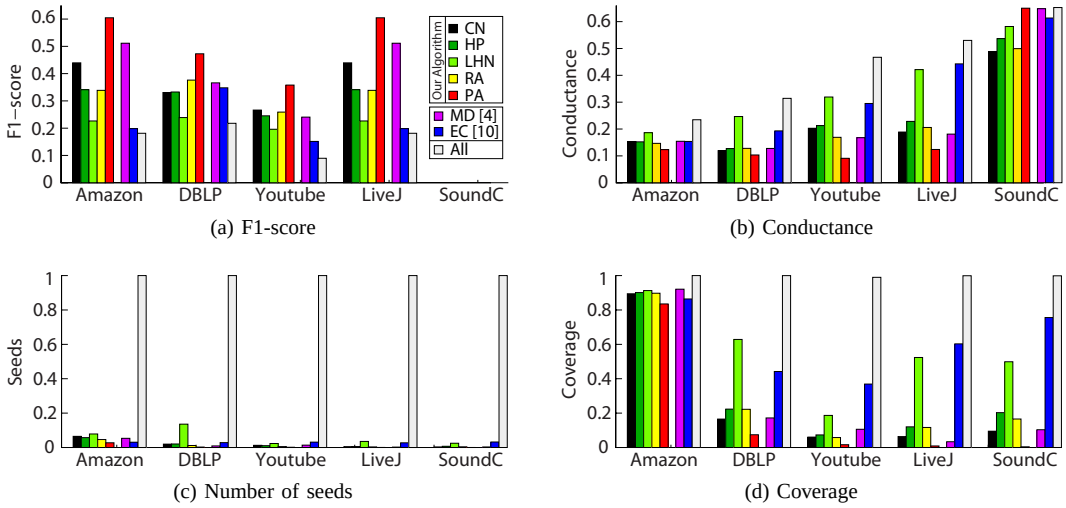
A. Datasets

The networks we have used for this study are listed in Table II. We have selected different types of publicly available real-world datasets. Additionally, we have collected a subset of users from an online social network of a sound sharing website (SoundCloud) and have generated a new network for this study.

Amazon is a product network in which nodes are products and two products have an edge if they were co-purchased frequently. *DBLP* is a collaboration network where nodes are authors and two authors are connected with an edge if they have co-authored at least one paper. In the *Youtube* and

[2]The community detection algorithm approximates PageRank with an accuracy value ϵ. In our experiments, we use a constant $\epsilon = 10^{-4}$ for comparing different seeding algorithms, instead of trying to find the accuracy value which leads to the best conductance.

[3]The complexity of PageRank-Nibble, which is the main components of the community detection algorithm, is $O(|S|\frac{\log^3 m}{\phi^2})$, where it can return a community S with conductance $< \phi$.



(a) F1-score (b) Conductance

(c) Number of seeds (d) Coverage

Figure 2: A comparison of different local seeding algorithms and the expanded communities from the selected seeds. CN, HP, LHN, RA, and PA refer to our local seeding algorithm (Algorithm 1) using the respective similarity indices (see Table I). EC [10] and MD [4] refer to the local seeding algorithms being compared with our algorithm, and All refers to when all the nodes in the network are used as seeds.

LiveJournal networks, the nodes are the users of the video sharing and online blogging websites, respectively, and the edges correspond to friendships. In the *SoundCloud* network, nodes are users and edges correspond to *following* relations.

B. Comparison

In order to compare the seeding algorithms, we have considered the number of nodes which are selected as seeds by each algorithm, the quality of the identified communities from these seeds, and the number of nodes being covered by these communities.

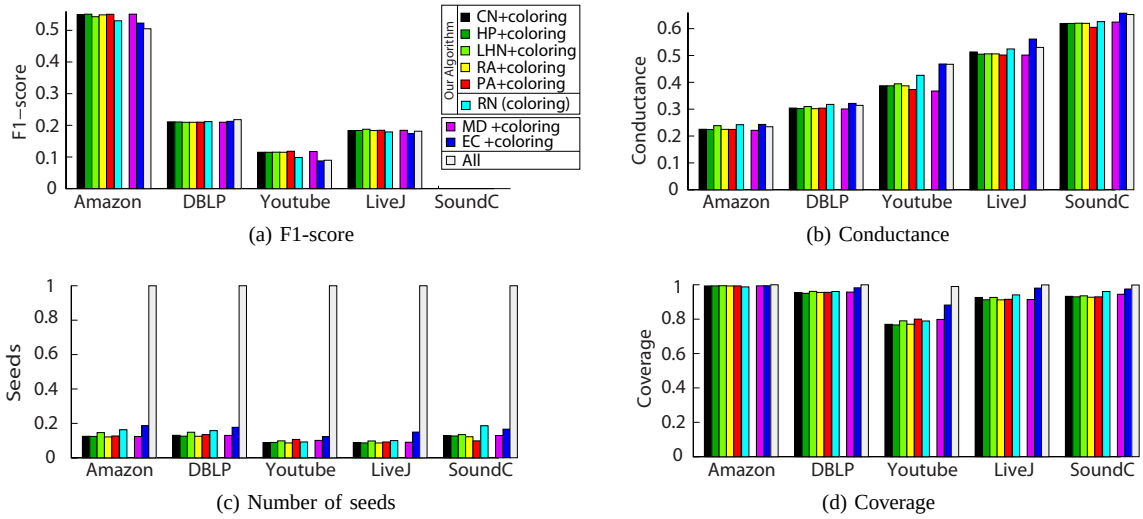
In order to compare the quality of the identified communities, we use both the conductance of the communities and the similarity with the ground truth communities. The similarity is calculated using the *F1-score* which is defined as $\textit{F1-score} = 2\ \frac{precision\ .\ recall}{precision+recall}$, where $recall = \frac{|S \cap C|}{|C|}$, $precision = \frac{|S \cap C|}{|S|}$, and S and C denote the detected and the ground truth community, respectively. The average f1-score over all the communities is used to compare the communities expanded from the seeds by different seeding algorithms.

If there is more than one community that overlaps with a ground truth community, we select the one with the highest f1-score, and the duplicate communities are ignored. Moreover, communities which do not have any common nodes with the ground truth communities are not considered in the calculation of the average f1-score. Such communities exist, since there are nodes in the networks which belong to a community but are not annotated to be in the ground truth community, i.e., the networks are "partially annotated" [19].

1) Link Prediction-based Seed Selection: Figure 2 shows a comparison of our link prediction-based seeding algorithm (Algorithm 1) using similarity indices CN, HP, LHN, PA, and RA (see Table I) with two other local seeding algorithms EC [10] and MD [4] (see Section III-B), as well as when all the nodes in the network are used as seeds (All). It can be seen that PA results in the highest average f1-score and the lowest average conductance for most of the networks being studied. The other four similarity indices used in our algorithm also succeed in selecting a small number of good seeds, which are expanded into high quality communities. However, none of the local seeding methods can achieve a high coverage in all the networks.

2) Biased Coloring-based Seed Selection: Figure 3 shows a comparison of seed selection enhanced with biased coloring, as well as the basic random coloring (RN). It can be seen, that by adding biased coloring, the coverage of the communities is dramatically improved regardless of the similarity index being used. Without biased coloring, our seeding algorithm (Algorithm 1) was able to identify a few very high quality communities, but after being enhanced with coloring (Algorithm 2), it selects a small number of seeds but now leads to communities with a similar average quality compared to when all the nodes are used as seeds (All). The figure also shows that using biased coloring has improved the coverage of existing local seeding methods, i.e., EC and MD (see Section III-B).

Note that the biased coloring is not deterministic since the color conflicts are resolved at random. Although it is possible to use a deterministic distributed coloring algorithms, e.g., [3], our experiments have shown that the induced randomness does not affect the community detection much



(a) F1-score (b) Conductance

(c) Number of seeds (d) Coverage

Figure 3: A comparison of different local seeding algorithms and the expanded communities from the selected seeds. CN, HP, LHN, RA, and PA refer to our local seeding algorithm enhanced with biased coloring (Algorithm 2) using the respective similarity indices, and RN refers to our basic random coloring algorithm (see Table I). EC + coloring and MD + coloring refer to existing local seeding algorithms which are also enhanced with our biased coloring algorithm, and All refers to when all the nodes in the network are used as seeds.

and the results are quite stable.[4]

3) Local versus Global Seeding: The seeding algorithms compared up to this point are all local methods. There are also seeding algorithms which assume that a global knowledge of a network exists, and therefore this knowledge can be used for selecting good seeds. In this study, we include the Spread hub (SH) algorithm [19] which requires the degree of all the nodes in the network to be known and which is shown to select good seeds (Section III-B). Table III shows the results using SH for three of the networks.

In addition to the global knowledge, SH requires the minimum number of seeds, k, to be known in advance. Unfortunately, our knowledge of the real community structure of many real networks is very limited, therefore it is not easy to estimate a correct value for k. It can be seen in the table that the selection of k dramatically affects the quality and the coverage of the communities. The table also shows the community quality and coverage when SH is enhanced with our biased coloring, and it can be seen that coloring can compensate for a bad selection of k. Although the global knowledge is available in this scenario, our experiments show that using local coloring for seed selection is a good and safe choice, since even with a global knowledge of the network, selecting the right number of seeds is not easy.

4) Execution Time: Finally, we have compared the execution time of personalized PageRank-based community detection using our seeding algorithm (PA + coloring) versus running the community detection for all the nodes in the network (All). We have also compared the execution times with an state-of-the-art local overlapping community detection algorithm, DEMON [7], which is based on the idea that different nodes have different views of the communities in their neighborhood and these communities can be merged into the global communities of the network. All the implementations we have used are in Python.[5]

Table III: Comparison of SH with different percentage of graph nodes as k

Dataset	k (% of n)	Seeds	F1-score	Conductance	Coverage
Amazon	3%	0.03	0.50	0.16	0.89
	10%	0.11	0.53	0.20	0.98
	15%	0.18	0.52	0.23	0.99
	3%+coloring	0.11	0.56	0.22	0.99
DBLP	3%	0.03	0.28	0.25	0.83
	10%	0.12	0.23	0.28	0.96
	15%	0.17	0.21	0.30	0.98
	3%+coloring	0.16	0.21	0.30	0.99
Youtube	3%	0.03	0.10	0.40	0.61
	10%	0.10	0.11	0.40	0.87
	15%	0.18	0.10	0.41	0.94
	3%+coloring	0.15	0.10	0.41	0.92

Table IV: Execution time

		Seeding	Community Detection	F1-Sc.	Cond.	Cov.
Amazon	PA+coloring	52 s	2 h 38 m	0.55	0.22	0.99
	All	-	17 h 15 m	0.51	0.23	1.00
	Demon	-	37 h 40 m	0.51	0.50	0.79
DBLP	PA+coloring	2 m 16 s	1 h 12 m	0.19	0.30	0.96
	All	-	8 h 42 m	0.21	0.31	1.00
	Demon	-	32 h 54 m	0.25	0.63	0.85
Youtube	PA+coloring	7 m 54 s	1 h 38 m	0.12	0.37	0.80
	All	-	14 h 47 m	0.09	0.47	0.99
	Demon	-	52 h 48 m	0.23	0.73	0.23

[4]In the figures, all the results for the coloring enhanced seeding methods are computed at least 5 times and the figures show the mean values with 95% confidence interval (the error bars were too small to be shown).

[5]We have used the implementation of Demon provided by its authors, and have used $\epsilon = 0.3$ and the default minimum community size for the experiments.

Table IV summarizes the execution times. It can be seen that our seeding algorithm (PA + coloring) is very fast and that the use of seeding dramatically reduces the execution time of the community detection. It can also be seen that our algorithm leads to a better combination of high coverage with good quality communities compared to DEMON.

VI. Conclusions

In this paper, a novel distributed parameter-free seed selection algorithm is presented which only requires local computations. In our algorithm, we have taken advantage of the similarity indices widely used for link prediction to select a small number of good seeds. We have also enhanced our seeding algorithm with a novel biased coloring algorithm to further improve the seed selection. The seeds identified by our algorithm have then been expanded into high quality overlapping communities using a personalized PageRank-based community detection algorithm which can also be computed locally.

Experiments using different types of large-scale real-world networks have shown that our seeding algorithm is able to pick nodes that are well-distributed over the networks and are expanded into communities with both high coverage and good quality. Our results also show that using seed selection can dramatically reduce the execution time of community detection while preserving the quality of the identified communities.

Acknowledgments. The research has received funding from the European Union Seventh Framework Programme (FP7/ 2007-2013) under grant agreement no. 257007.

References

[1] B. Abrahao, S. Soundarajan, J. Hopcroft, and R. Kleinberg. On the separability of structural classes of communities. In *Proceedings of the 18th ACM SIGKDD conference on Knowledge discovery and data mining*, 2012.

[2] R. Andersen and K. Lang. Communities from seed sets. In *Proceedings of the 15th conference on World Wide Web*, 2006.

[3] L. Barenboim and M. Elkin. Distributed (Delta+1)-Coloring in Linear (in Delta) Time. In *Procideeings of Symposium on Theory of Computing, STOC'09*, pages 111–120, 2009.

[4] Q. Chen and M. Fang. An Efficient Algorithm for Community Detection in Complex Networks. In *the 6th Workshop on Social Network Mining and Analysis*, 2012.

[5] A. Clauset. Finding local community structure in networks. *Physical Review E*, 72(2):026132, Aug. 2005.

[6] G. Cordasco and L. Gargano. Label propagation algorithm : a semi-synchronous approach. *International Journal of Social Network Mining*, 1(1):3–26, 2012.

[7] M. Coscia, G. Rossetti, F. Giannotti, and D. Pedreschi. DEMON: a local-first discovery method for overlapping communities. In *Proceedings of the 18th ACM SIGKDD conference on Knowledge discovery and data mining*, 2012.

[8] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, Feb. 2010.

[9] U. Gargi and W. Lu. Large-Scale Community Detection on YouTube for Topic Discovery and Exploration. In *Proceedings of the Fifth International Conference on Weblogs and Social Media*, 2011.

[10] D. F. Gleich and C. Seshadhri. Vertex neighborhoods, low conductance cuts, and good seeds for local community methods. In *Proceedings of the 18th ACM SIGKDD conference on Knowledge discovery and data mining*, pages 597–605, 2012.

[11] A. Lancichinetti and S. Fortunato. Community Detection Algorithms: A Comparative Analysis. *Physical Review E*, 80(5):1–11, Nov. 2009.

[12] L. Lü and T. Zhou. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150–1170, Mar. 2011.

[13] M. Luby. A simple parallel algorithm for the maximal independent set problem. In *Proceedings of the 17th annual ACM symposium on Theory of computing - STOC'85*, 1985.

[14] M. Luby. Removing Randomness in Parallel Computation Without a Processor Penalty. *Journal of Computer and System Sciences*, pages 162–173, 1988.

[15] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, page 29, 2007.

[16] H. Shen, X. Cheng, K. Cai, and M.-B. Hu. Detect overlapping and hierarchical community structure in networks. *Physica A: Statistical Mechanics and its Applications*, 388(8):1706–1712, Apr. 2009.

[17] S. Soundarajan and J. E. Hopcroft. Use of Local Group Information to Identify Communities in Networks. *ACM Transactions on Knowledge Discovery from Data*, 2014.

[18] D. A. Spielman and S.-H. Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the 36th annual ACM symposium on Theory of computing*, 2004.

[19] J. J. Whang, D. F. Gleich, and I. S. Dhillon. Overlapping community detection using seed set expansion. In *Proceedings of the 22nd ACM international Conference on information & knowledge management*, 2013.

[20] J. Xie, S. Kelley, and B. Szymanski. Overlapping community detection in networks: the state of the art and comparative study. *ACM Computing Surveys*, 45(4), 2013.

[21] B. Yan and S. Gregory. Detecting community structure in networks using edge prediction methods. *Journal of Statistical Mechanics: Theory and Experiment*, 2012(09):P09008, Sept. 2012.

[22] J. Yang and J. Leskovec. Defining and evaluating network communities based on ground-truth. In *Proceedings of the IEEE International Conference on Data Mining*, 2012.