Detecting Communities from Given Seeds in Social Networks

Jason Riedy David A. Bader Karl Jiang Pushkar Pande Richa Sharma

February 22, 2011

Abstract

Analyzing massive social networks challenges both high-performance computers and human understanding. These massive networks cannot be visualized easily, and their scale makes applying complex analysis methods computationally expensive. We present a region-growing method for finding a smaller, more tractable subgraph, a community, given a few example seed vertices. Unlike existing work, we focus on a small number of seed vertices, from two to a few dozen. We also present the first comparison between five algorithms for expanding a small seed set into a community. Our comparison applies these algorithms to an R-MAT generated graph component with 240 thousand vertices and 32 million edges and evaluates the community size, modularity, Kullback-Leibler divergence, conductance, and clustering coefficient. We find that our new algorithm with a local modularity maximizing heuristic based on Clauset, Newman, and Moore performs very well when the output is limited to 100 or 1 000 vertices. When run without a vertex size limit, a heuristic from McCloskey and Bader generates communities containing around 60% of the graph's vertices and having a small conductance and modularity appropriate to the result size. A personalized PageRank algorithm based on Andersen, Lang, and Chung also performs well with respect to our metrics.

1 Introduction

Modeling and analysis of the massive social networks now available challenges the capacity of both high-performance computers and human understanding. These massive networks resist attempts for global visualization and characterization. For example, the 500M user Facebook friendship graph [13] has approximately 30 thousand edges *per each pixel* on a 1600×1200 screen. Applying analysis kernels like k-betweenness centrality [17] to the entire graph is not yet feasible on commodity platforms. Extracting smaller, relevant communities of interest opens these massive networks to wider analysis.

Global community detection methods in large networks provide insights into the structure of the network. Global community detection methods apply to networks from fields including biological [15, 27], metabolic [26], social [28, 12, 31], co-citation [23, 16], and the World Wide Web [18]. These techniques partition the entire graph into communities without user intervention.

However, little work exists for selecting an "appropriate" community when given a small number of vertices. Users may want a community containing a few examples, or *seed vertices.* We expect users will select fewer than 100 seed vertices and often only up to ten. Clauset [9], Luo [20] and Bagrow [5] give algorithms for local community detection starting from a single seed vertex. Anderson and Lang [2] address the problem of expanding a large seed set into a community. We propose algorithms for detecting a community around a small set of seeds, from 2 to 10 seeds, and compare results on an artificial random graph.

Section 2 defines the following commonly used evaluation criteria beyond community size: modularity, Kullback-Leibler divergence, conductance, and clustering coefficients. We expect that no one criteria satisfies all needs. We discuss existing algorithms and modifications in Section 3 before introducing our new greedy algorithm in Section 4. Our new algorithm uses Clauset-Newman-Moore (CNM) [10] or McCloskey-Bader (MB) [4] approximate local modularity maximization to grow a community around the seed vertices.

Our sequential implementation is within SNAP (Small-world Network Analysis and Graph Partitioning), a framework for exploratory graph analysis and partitioning [3]. Section 5's experiments investigate communities in a graph generated using the Recursive MATrix (R-MAT) algorithm [8]. R-MAT graphs are representative of real-world networks with power-law degree and small-world characteristics. The largest component in our test graph contains around 240 thousand vertices and 32 million edges (average degree of over 134).

Our comparison, the first such comparison known to the authors, finds that the algorithm using the heuristic Clauset-Newman-Moore (CNM) growth criteria produces communities with larger modularity (a measure of edge distribution's deviation from a uniform random model) and clustering coefficients and lower conductance (a normalized measure of the cut) than other methods when limited to 100 or 1 000 vertices of a 240 thousand vertex graph. When unlimited, however, the CNM algorithm risks producing communities that contain nearly the entire graph.

When run without vertex size limits, the McCloskey-Bader modularity metric produces relatively small communities (around 60% of the vertices in the graph) with low conductance. The smaller, MB-generated communities also have a low modularity, which is surprising for an algorithm locally maximizing modularity. The resulting community is smaller, and the modularity is appropriate to the community size. A personalized PageRank algorithm based on Andersen, Chung, and Lang [1] typically produces communities around the same size (around 57% of the vertices) with higher (better) modularity and clustering measurements but also higher (worse) conductance.

The other methods, a baseline breadth-first algorithm and a random walk algorithm based on Anderson and Lang [2], are less competitive.

2 Background and definitions

There is no single, universally accepted definition of a "community" within a social network. One definition is that a community is a collection of vertices more strongly connected than would occur from random chance, leading to modularity-based methods. Other definitions [25] require vertices to be more connected to others within the community than those outside, either individually or in aggregate. To compare communities without delving too far into different definitions, we consider three of the most commonly used metrics:

- modularity, an estimate of a community's deviation from random chance [22, 4];
- the Kullback-Leibler divergence [19] from the same model;
- conductance, a measure of the community border's cut [2]; and
- clustering coefficients, a measure of interconnectedness within the community.

For interpreting the modularity and clustering coefficient metrics, larger values often are considered better indicators of a community. For conductance, smaller values often are considered better. Our new algorithm in Section 4 finds for a locally maximal modularity community enclosing the seeds.

The modularity metric proposed by Newman [21] compares the connectivity within a collection of vertices to the expected connectivity of a random graph with the same degree distribution. We use a formulation based on [4]. Let L be the number of edges in an undirected graph $G = G(V, E)$ with vertex set V and edge set E. Let $S \subset V$ induce a graph $G_S = G(S, E_S)$ with $E_S \subset E$ containing only edges where both endpoints are in S. Let L_S be the number of edges $|E_S|$, and let $\overline{L_S}$ be an expected number of edges in S given a background model. Then define the modularity of the community induced by S as

$$Q_S = \frac{1}{L}\left(L_S - \overline{L_S}\right). \tag{1}$$

The modularity represents the deviation of connectivity in the community in hand, S, from an expected background model. Modularity also can be defined as $Q_S = p_S - q_S$, there p_S is that the probability of an edge picked uniformly at random from E is adjacent to S, and $q_S = \overline{L_S}/L$ is the expected probability over all graphs from a background model.

Newman [21] considers the specific background model of a random graph with the same degree distribution as G where edges are independently and identically distributed. If X_S is the total number of edges in G where either endpoint is in S, then we have [4]

$$Q_S = \frac{1}{L}\left(L_S - \frac{X_S^2}{4L}\right). \tag{2}$$

A subset S is considered a module when there are more internal edges than expected, $Q_S > 0$. The L_S term in Equation (2) encourages forming large modules, while the X_S term penalizes modules with excess external edges. Maximizing Q_S should find communities with more internal connections than external ones.

Modularity has known limitations. Fortunato and Barthélemy[14] demonstrate that global modularity optimization cannot distinguish between a single community and a group of smaller communities. Berry, *et al.* [6] provide a weighting mechanism that overcomes this resolution limit. We do not consider weighting in this paper. Instead, Section 5 compares optimizing the modularity in Equation (2) with the normalizing method of Bader and McCloskey [4]. Bader and McCloskey's algorithm only merges vertices into the community when the change is deemed statistically significant within a simple statistical model assuming independence between edges.

A similar measure is the Kullback-Leibler divergence [19] from the same base model. Let $p_S = L_S/L$ and $q_S = \overline{L_S}/L$ as above. The Kullback-Leibler divergence is

$$D_{KL}(P_S||Q_S) = p_S \log_2 \frac{p_S}{q_S}, \tag{3}$$

where P_S is the distribution of probabilities that an edge selected uniformly at random from G is adjacent to vertex set S, and Q_S is the distribution of probabilities that an edge selected uniformly at random is adjacent to S in a random graph with the same degree sequence as G. We focus on small communities, so we are far from any large-sample convergence of the K-L divergence and modularity.

Another metric, graph conductance, measures the normalized cut between a graph induced by vertex set S and the graph induced by the remaining vertices $V \setminus S$. Let the volume of a set of vertices U be the sum of the degrees $d(v)$ for each vertex $v \in U$, $\text{Vol}(U) = \sum_{v\in U} d(v)$. The volume of the entire graph $\text{Vol}(G) = 2|E|$, where $|E|$ is the number of edges in E. Denote the cut by

$$\partial(S) = \{\{u, v\}|\{u, v\} \in E, u \in S, v \notin S\},$$

and the size of the cut by $|\partial(S)|$. Then the conductance of a vertex set S is defined [2] as

$$\phi(S) = \frac{|\partial(S)|}{\min\{\text{Vol}(S), \text{Vol}(V \setminus S)\}}. \tag{4}$$

If $S = V$ or $S = \emptyset$, let $\phi(S) = 1$, the largest obtainable value. For comparison, we can express Equation (4) using the same quantities as Q_S using $\text{Vol}(S) = X_S + L_S$, $\text{Vol}(V \setminus S) = 2L - (X_S + L_S)$, and $|\partial(S)| = X_S - L_S$. So

$$\phi(S) = \frac{X_S - L_S}{2\min\{X_S + L_S, 2L - (X_S + L_S)\}}.$$

The minimum conductance over all vertex sets $S \subset V$ is the graph's conductance. A large conductance value implies random walks rapidly converge to their stationary distribution [7]. Minimizing the conductance of a subset S rather than the entire graph intuitively implies a bottleneck between the induced subgraph and the remainder. Random walks will tend to stay in the induced subgraph.

The third metric we use for evaluating communities is the induced subgraph's clustering coefficient. The clustering coefficient of an undirected graph G is defined in terms of triplets. A triplet centered around vertex v is a pair of edges $\{a, v\}, \{v, b\} \in E$. If $\{a, b\} \in E$, then the triplet is *closed*, otherwise it is *open*. The clustering coefficient for a graph is defined [29] as

$$C = \frac{\text{number of closed triplets}}{\text{total number of triplets}}. \tag{5}$$

A clustering coefficient $C = 0$ implies the graph is a forest, and $C = 1$ implies the graph is a union of disjoint complete graphs. We do not define C on graphs with no triplets; they do not occur in our case. To evaluate vertex sets as communities, we compute the clustering coefficient of the induced subgraph. This ignores connections to the remainder of the graph and does not suffice to indicate a community. For example, every subset of a complete graph would appear the same as the graph itself considering only clustering coefficients. The clustering coefficient does, however, summarize the connectivity within a subgraph. The edge counts above (L_S and X_S) cannot express Equation (5).

Note that none of these metrics count numbers of vertices. Including isolated, degree-zero vertices in S does not affect S's modularity, conductance, or clustering coefficient. Algorithms may require handling isolated vertices specially.

3 Prior Work

We consider prior work from two approaches to seed-set expansion. One explicitly grows a region around the seeds. The other approach computes a metric across the graph's vertices, sorts the vertices, and includes the vertices incrementally to optimize some criterion. Other algorithms are possible, including examining the dendogram computed by an agglomerative clustering of the entire graph. We limit this paper's survey to seed set expansion using a *small* seed set, from two to ten seed vertices. In contrast, other work like [2] considers seed sets with hundreds of seed vertices or more, and not all seeds may be representative of the desired community. Instead of looking for a community similar to an example, we seek a community containing the seeds.

Section 4's algorithm will grow a region locally maximizing modularity (Equation (2)). As a baseline, we consider growing a community by a breadth-first expansion around the seeds. Breadth-first expansion runs in $O(\mathrm{Vol}(S'))$ operations where S' is the returned region. Our comparison in Section 5 expands a region by three steps or until a specified number of vertices are included. This method is labeled BFS.

The technique of Anderson and Lang [2] follows the second approach and computes the probability distribution of random walk steps starting from the seed set. Their algorithm orders vertices by a degree-weighted probability, sweeps across the vertices in order, and records the set of consecutively ordered vertices of least conductance (Equation (4)) for each step. Their algorithm returns least conductive set over multiple steps as the community. Their ultimate algorithm also optimizes each set locally using $s - t$ flows, bounds the volume of the sets, and can target an input conductance. Ignoring the $s - t$ flow optimization, each pass requires $O(\mathrm{Vol}(S^{(i-1)}) + |\tilde{S}^{(i)}| \log |\tilde{S}^{(i)}|)$ operations where $S^{(i-1)}$ is the set output by sweep $i - 1$ and $\tilde{S}^{(i)}$ is the size of the intermediate set at sweep i before thresholding. The comparison in Section 5, labeled RW, considers the algorithm without volume bounds or local optimization.

Andersen, Chung, and Lang propose an algorithm, *PageRank-Nibble* [1], which computes an approximate personalized PageRank vector and finds the least conductive set by incrementally including vertices in order of decreasing PageRank. A personalized PageRank vector computes the stationary distribution of random walks starting from the input seeds and including a regularization factor emulating a slight chance of a random jump not constrained by graph edges. The PageRank-Nibble algorithm targets an input conductance and volume. Our comparison in Section 5, labeled PR, does not include the conductance or volume targets and instead finds the set of least conductance including all the seeds. Using an approximate PageRank computation and treating its parameters as constants, our basic implementation of personalized PageRank runs in $O(\mathrm{Vol}(S') + |V| \log |V|)$ operations rather than the $O(\mathrm{Vol}(S') + |S'| \log |S'|)$ operations of [1]. This does not affect the metrics used in our comparison.

Two corner cases occur with these vertex-ordering algorithms: ensuring seeds' inclusion and preventing inclusion of zero-degree vertices. As stated, the Andersen and Lang [2] and Andersen, Chung, and Lang [1] algorithms do not ensure the seeds are members of output set. Both algorithms target large seed sets, those of hundreds to thousands of vertices. Such a large set of seeds serves more as an example, and some example seeds may not be relevant. Our purpose is different. We seek a community enclosing the given seeds, not a community similar to the given seeds. Section 5 evaluates performance under the assumption that all seeds must be in the final community. Our modified Andersen and Lang and Andersen, Chung, and Lang algorithms place the seeds into the output community of each pass and then minimizes the conductance up to the remaining output limit. This produces smaller but often disconnected communities.

The other corner case is preventing inclusion of zero-degree vertices. Such isolated vertices do not affect any of our non-size evaluation criteria. If a metric evaluates $0/0$ and produces a NaN (not-a-number), some sorting routines place the NaNs haphazardly throughout the array of scores. The conductance minimization may include these zero-degree vertices, inflating the community's size while not changing its other properties. This is a minor but important inconvenience that occurred during this paper's development. Substituting 0 for $0/0$ degree-weighted probabilities rather than removing the zero-degree vertices handles the problem without remapping the vertex numbers. Our comparison runs on a single component, so isolated vertices already are removed.

Algorithm 1: Greedy agglomerative expansion of a seed set S to a community S'; $O(|S'| \log \mathrm{Vol}(S'))$

```
Initialize each vertex as a community.                                              $O(|V|)$
Let $S \subset V$ be the input seed set, and $C$ be the communities for $S$'s vertices.
Construct a max-heap of the best adjacent merges.                          $O(\mathrm{Vol}(S) \log \mathrm{Vol}(S))$
while some adjacent merge is acceptable do
    Select the best merge between a community $W \in C$ and an adjacent community $U$.        $O(1)$
    Merge $U$ into $W$.                                                               $O(|U|)$
    Update the $\Delta Q(W', Z)$ for all communities $W' \in C$ and $Z$ neighboring $U$.   $O(\log \mathrm{Vol}(\cup_{W \in C} W))$
end while
Return $S' = \cup_{W \in C} W$ as the (possibly disconnected) community.                 $O(|S'|)$
```

4 Seed set expansion

We present a greedy agglomerative algorithm that grows a community around a given, small seed set. Starting from a set of seed vertices, our algorithm pulls adjacent vertices into the community to maximize modularity. Any incrementally computable metric can replace modularity. We focus on modularity to complement Clauset, Newman, and Moore's global method [10].

We follow the general agglomerative framework found in [10] and [3] for community detection. Algorithm 1 provides a high level description along with the asymptotic complexity of each step. The algorithm begins with the vertices V as the sole members of $|V|$ disjoint communities. Given an input set of seed vertices S, their communities form an initial set $C = \{\{v\} : v \in S\}$. The algorithm maintains a queue of the best community merges from those communities neighboring those in C into the communities in C. Any community not in C remains a singleton community. Once no merges remain that increase the objective, modularity, the algorithm returns the set of vertices $S' = \cup_{W \in C} W$. At most $|V| - 1$ merges are possible and vertex sets in C remain disjoint.

The input seeds may *not* be part of the same, modularity-maximizing set in C. In this case, the returned set of vertices S' induces a disconnected subgraph of G but still contains every seed in S. Section 5's unrestricted tests almost always produce only one connected subgraph, but tightly restricting the output's size frequently produces a disconnected "community."

Agglomeration evaluates possible merges against their modularity changes. For merging a set of vertices U into a disjoint set of vertices $W \in C$, we require that the change $\Delta Q(W, U) = Q_{W \cup U} - (Q_W + Q_U) > 0$. Expanding Equation (1),

$$\begin{aligned} L\Delta Q(W, U) &= Q_{W \cup U} - (Q_W + Q_U) \\ &= (L_{W \cup U} - (L_W + L_U) - (\overline{L_{W \cup U}} - (\overline{L_W} + \overline{L_U})) \\ &= L_{W \leftrightarrow U} - \overline{(L_{W \cup U} - (L_W + L_U))}, \end{aligned}$$

where $L_{W \leftrightarrow U}$ is the number of edges between vertices in sets W and U.

Assuming the edges are independent and identically distributed across vertices respecting their degrees [10], then

$$\overline{(L_{W \cup U} - (L_W + L_U))} = L \cdot \frac{L_W}{2L} \cdot \frac{L_U}{2L},$$

and

$$\Delta Q(W, U) = \frac{L_{W \leftrightarrow U}}{L} - \frac{L_W}{2L} \cdot \frac{L_U}{2L}. \tag{6}$$

Our algorithm tracks the best merge adjacent to the communities in C. For the classic CNM (Clauset-Newman-Moore) algorithm [10], the best merge is the merge with largest $\Delta Q(W, U)$. The algorithm terminates when no available merge will increase the modularity. The MB (McCloskey-Bader) variant [4] ranks $\Delta Q(W, U)$ values after normalizing by the standard deviation of $\Delta Q(W, Z)$ for all communities Z adjacent to W. MB also restricts merges to those deemed *statistically significant* by comparing $\Delta Q(W, U)$ to the mean $\overline{\Delta Q(W, Z)}$. The MB variant terminates either when no available merge increases the modularity or when all available merges are statistically insignificant with respect to a sliding window of historical ΔQ values. For our experiments, we accept merges within 1.5 standard deviations of the mean and track 2000 ΔQ values.

If the output is S', Algorithm 1's operation complexity is $O(|S'| \log \mathrm{Vol}(S'))$. Currently Algorithm 1 merges U into W eagerly. A lazy union-find reduces the cost of line 6 to practically $O(1)$, but in our case the sum of all $|U|$ is $|S'|$ and line 7 dominates.

In some cases one of the seed vertices may be inherently better connected than the others. Then all agglomerations may occur into only the community seeded by that vertex. The purely greedy Algorithm 1 does not bias the selection of the next merge. A future algorithm will cycle through the communities in C to expand more evenly around the seeds.

5 Evaluation

Our implementation of Algorithm 1 uses the SNAP [3] framework. To compare generated communities, our implementation also contains the breadth-first method, modified Anderson-Lang, and PageRank algorithms described in Section 3. For the Anderson-Lang algorithm, we take the best conductance over 60 steps. For the personalized PageRank method, we set the regularization factor $\alpha = 0.0625$ and use the approximation method of [1] with the accuracy of $\varepsilon = 2^{-24}$, equivalent to IEEE-754 binary single precision. We compare results when the output set size is limited to 100 vertices, limited to 1 000 vertices, or without any *a-priori* size limitation.

In these experiments, we evaluate the output set by size, modularity, conductance, and clustering coefficient. Our sample graph is a recursively generated R-MAT graph [8] G with 239 698 connected vertices and 32 238 438 edges for an average degree of 134.5. The R-MAT parameters are $a = 0.57$, $b = c = 0.19$, and $d = 1 - (a + b + c) = .05$. The R-MAT generator in SNAP targeted 300 thousand vertices and 45 million edges including multi-edges and self-loops. Our sample graph G is the largest component of the generated graph, simplified to remove multi-edges and self-loops.

We generate ten unique seed sets of each size $|S| \in \{2, 3, 5, 10\}$ in three different ways for a total of 120 seed sets. The three methods are as follows:

COMM Compute a global partitioning of G using the CNM algorithm [10]. Choose seeds uniformly at random from the second-largest community. None of the resulting seeded communities cover more than 65% of this base community.

RW-1 Choose a vertex v uniformly at random from G. Perform a uniform random walk of a single step and take the terminal vertex as a seed. Return to v and repeat to generate $|S|$ unique seeds.

RW-3 Choose a vertex v uniformly at random from G. Perform a uniform random walk of *three* steps and take the terminal vertex as a seed. Return to v and repeat to generate $|S|$ unique seeds.

For the random-walk methods RW-1 and RW-3, the vertex v is re-chosen if 100 random walks fail to generate a seed set of the desired size $|S|$. All 120 seed sets are unique.

The CNM-based seed generation method (COMM) permits comparing results with a globally agglomerative partitioning, while the random walk methods (RW-1, RW-3) reflect the model behind the PageRank and Anderson-Lang methods.

5.1 Community Sizes

Run without a community size limit, Figure 1 shows the community sizes as a percentage of G's number of vertices. Table 1 provides statistics on the output size $|S'|$. For our test graph and the algorithm settings discussed above, RW (Andersen and Lang's random walk) often produces the smallest communities, covering a median of 33% of the graph although with an unexplained jump in community size with ten seeds. We neither apply a volume threshold nor optimize the cut as in [2]. The CNM (Clauset-Newman-Moore) algorithm produces a curious mix of community sizes spanning from 36% to 97% of the graph with concentrations at the two extremes. The MB (McCloskey-Bader) and PR (PageRank) algorithms consistently cover around 61% and 57% of the graph, respectively. We suspect that the normalization in MB and regularization in PR smooths the distribution of changes, producing a more uniform result. The BFS (breadth-first expansion) algorithm subsumes almost the entire component in three hops.

Table 1: Statistics on the number of vertices in the component contained in the final community. The table is across all seed generation methods.

Algorithm	Statistics of $\lvert V_{\text{component}}\rvert$ as a percentage of $\lvert S'\rvert$					
	Minimum	1$^{\text{st}}$ Quartile	Median	Mean	3$^{\text{rd}}$ Quartile	Maximum
BFS	99.70%	100.00%	100.00%	99.98%	100.00%	100.00%
CNM	36.42%	36.83%	40.53%	57.90%	93.25%	97.20%
MB	60.12%	60.63%	60.81%	60.91%	61.12%	62.02%
PR	56.19%	56.86%	57.09%	57.11%	57.38%	58.88%
RW	25.88%	32.78%	33.10%	34.65%	33.71%	42.85%

When limited to 100 or 1 000 vertices, only the RW (random walk) frequently produces fewer than the requested size. RW finds a minimal conductance set of no more than the limit after ranking by a simulated random walk. PR runs a similar minimization but often produces sets of the limiting size. The other methods always produce sets of the limiting vertex size.

Figure 2 shows the number of components in the final community for each algorithm, each seed generation method, and for each size limit. The diffusive PageRank algorithm always produces a single component even when limited to communities of size 100. Other non-trivial algorithms produce more components as the size limit decreases. Even when seeds are at most two steps from each other (RW-1), imposing an output limit of 100 vertices prevents most algorithms from producing a single output component. The CNM and MB region-growing algorithms do not enforce any selection policies; growing initial regions in a round-robin fashion may produce fewer components.

The final community's volume is roughly proportional to the size of its vertex set for our test graph. Figure 3 shows the volumes of produced communities. The total volume of the graph is twice the number of edges, or 64 476 876. The RW algorithm often produces smaller communities and thus smaller volumes.

Figure 4 shows that nearly every algorithm and seed generation method often includes a vertex of maximum degree. Future research will investigate effects of restricting the volume or maximum degree within the community.

5.2 Modularity and Kullback-Leibler Divergence

The larger the modularity, the further the distance from a random subgraph with the same degree distribution. A positive modularity implies some structure more likely than a random distribution of edges. Figure 5 shows that almost all the methods produce communities of positive modularity.

Because BFS subsumes almost the entire component when permitted to grow without limit, its modularity is trivially large. Modularity in this extreme measures the likelihood of randomly not generating the original graph. When CNM also produces large communities, the results also have trivially large modularity. CNM's local, greedy unscaled modularity maximization sometimes drives the algorithm to include most of the graph. While MB also locally and greedily maximized modularity, the statistical rescaling described in Section 2 prevents subsuming the entire graph. Without a growth limit, MB produces *less modular* sets than do PR and RW.

When limited to 100 or 1 000 vertices, CNM's local maximization still produces more modular communities. The MB, PR, and RW algorithms often produce communities of approximately the same modularity. The smallest limit, 100 vertices, leads PR to produce lower modularity communities more often. BFS often produces vastly less modular communities. The only algorithms to produce *negative* modularity communities when limited are BFS and PR.

The Kullback-Leibler divergence in Figure 6 roughly mirrors Figure 5's modularity. At the smallest limit, however, MB and PR produce higher KL divergences than expected from the modularity.

5.3 Conductance and Cut Size

Conductance is a ratio of the cut size to the least neighboring volume. A smaller conductance implies a smaller relative cut, and a random walk is less likely to leave the community. Figure 7 shows the communities' conductance first for all limits in one, and then for the 1 000 and 100 vertex cases.

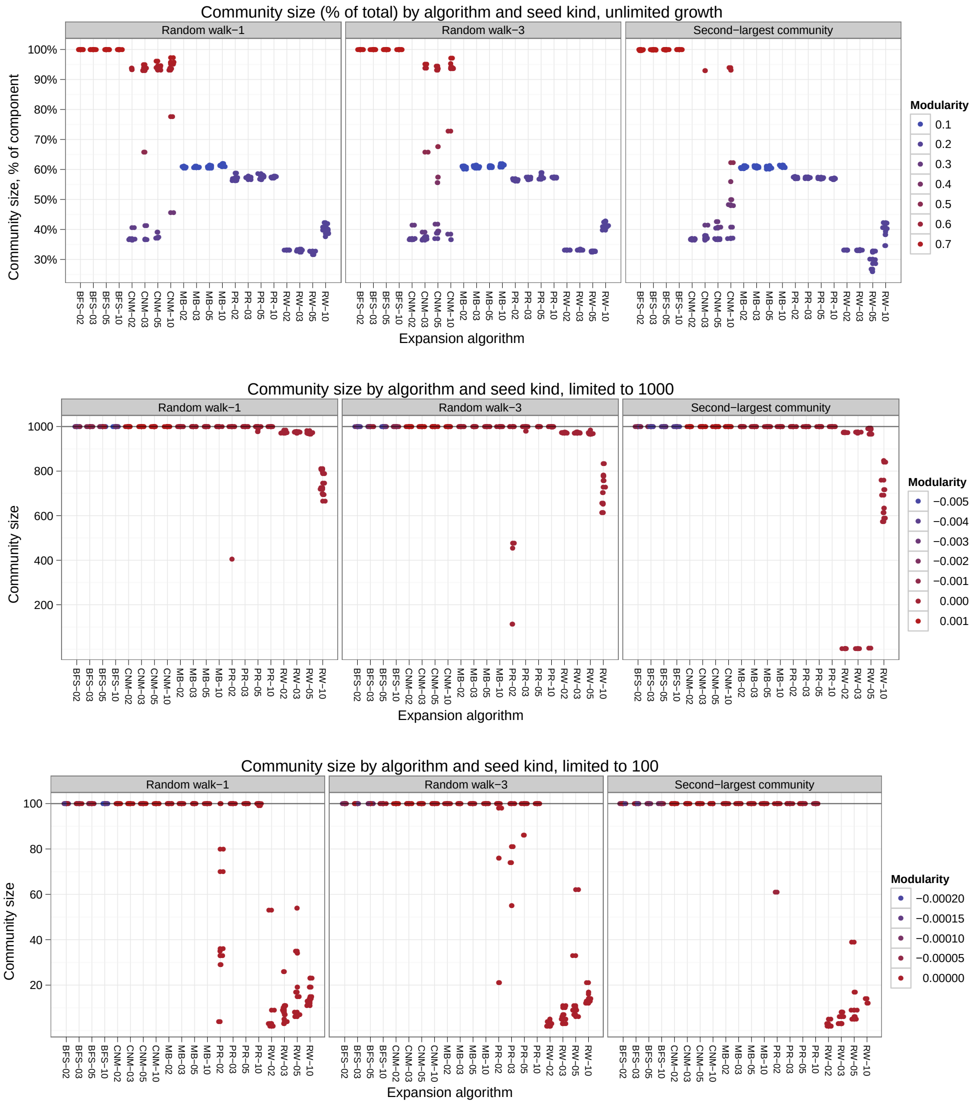


Figure 1: Community sizes as a percentage of the component size when the algorithms run without a size limit show that the breadth-first expansion (BFS) is not competitive with any method. The horizontal axis combines the abbreviation for the algorithm with the number of seeds. The separate panels show different seed generation methods. A darker solid horizontal line denotes a growth limit. Dots are jittered horizontally but not vertically.

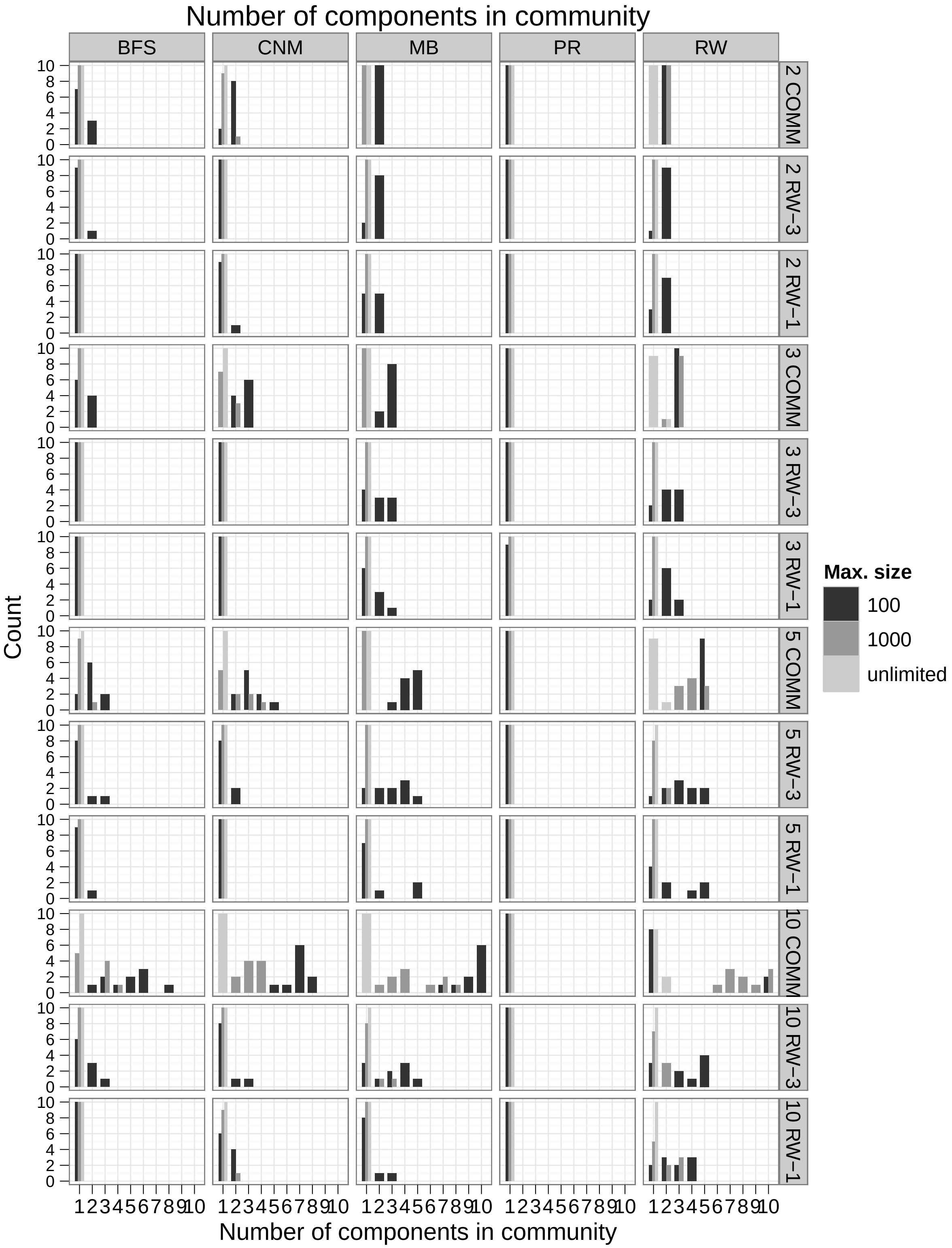


Figure 2: Decreasing the output size limit increases the number of components in the result community for all but the PageRank (PR) algorithm. The panels are indexed on the top by the algorithm and on the side by the number of seeds and the generation method.

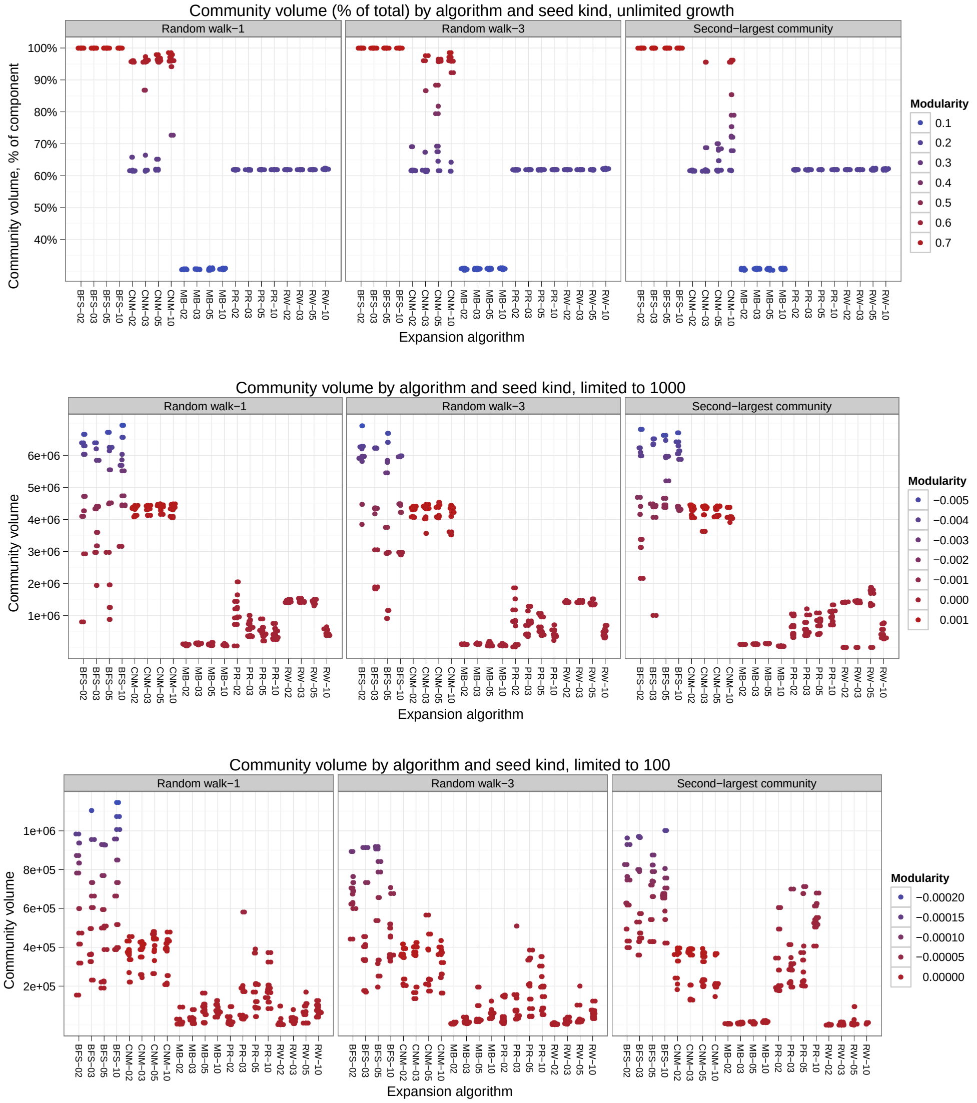


Figure 3: Community volumes in the unlimited case reflect the vertex size of the final community. When restricted, RW often produces smaller communities, leading to a smaller volume. The total volume of the graph is 64 476 876, so the largest volume of a restricted community is around half of a percent of the total. The horizontal axis combines the abbreviation for the algorithm with the number of seeds. The separate panels show different seed generation methods. Dots are jittered horizontally but not vertically.

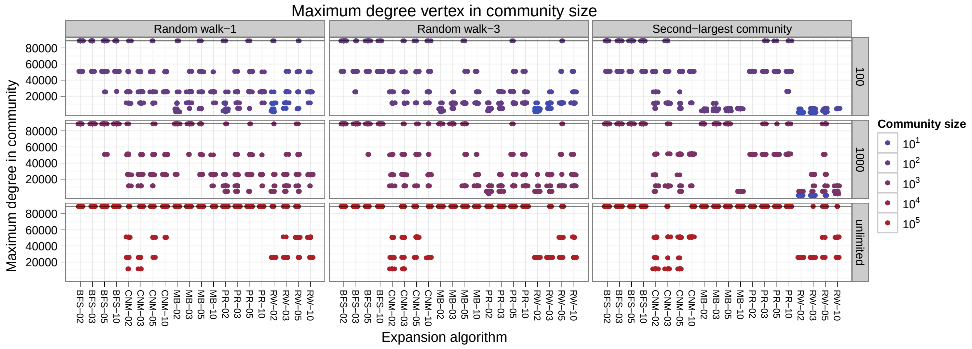


Figure 4: The algorithms often include a vertex with the graph's maximum degree. The dark horizontal line shows the maximum degree in the test graph.

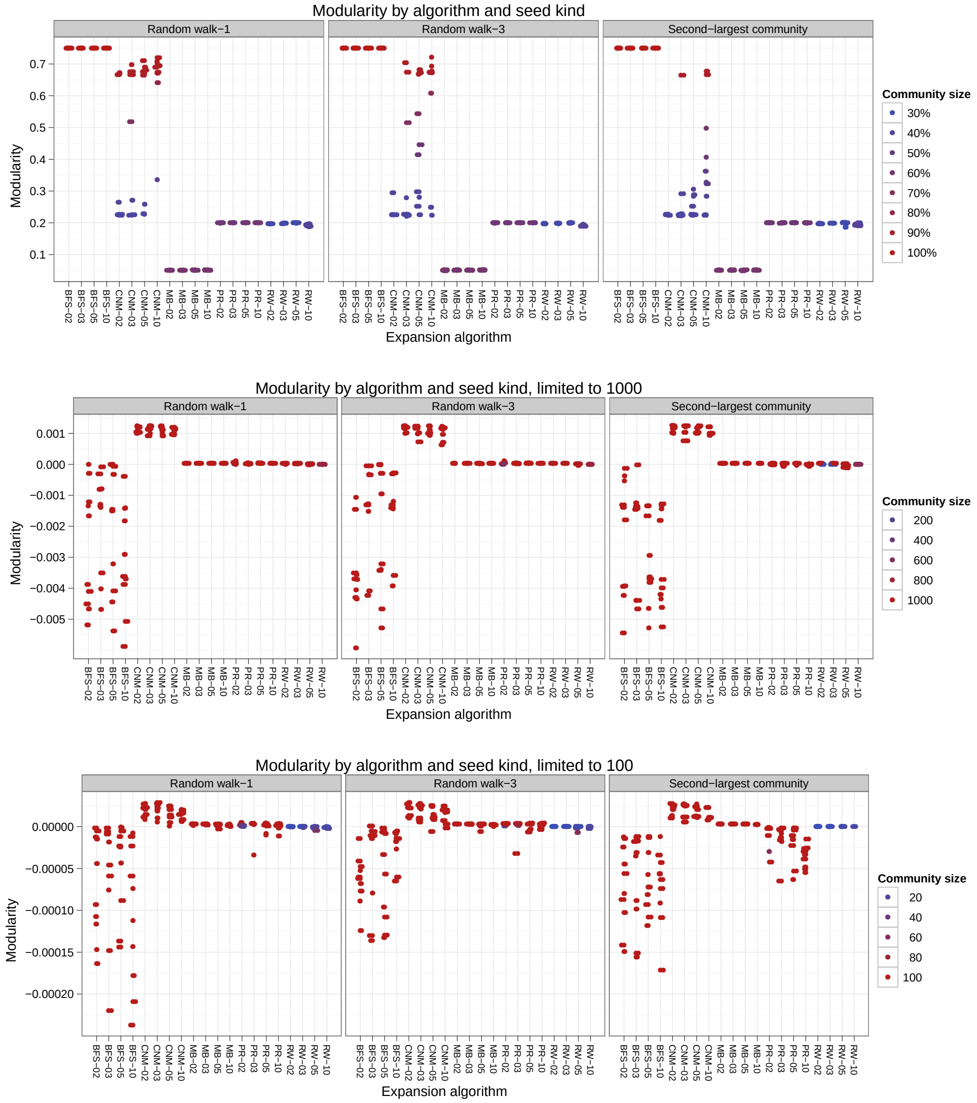


Figure 5: The size-limited runs produce modularity values far below the unlimited runs. Each plot has a different modularity scale.

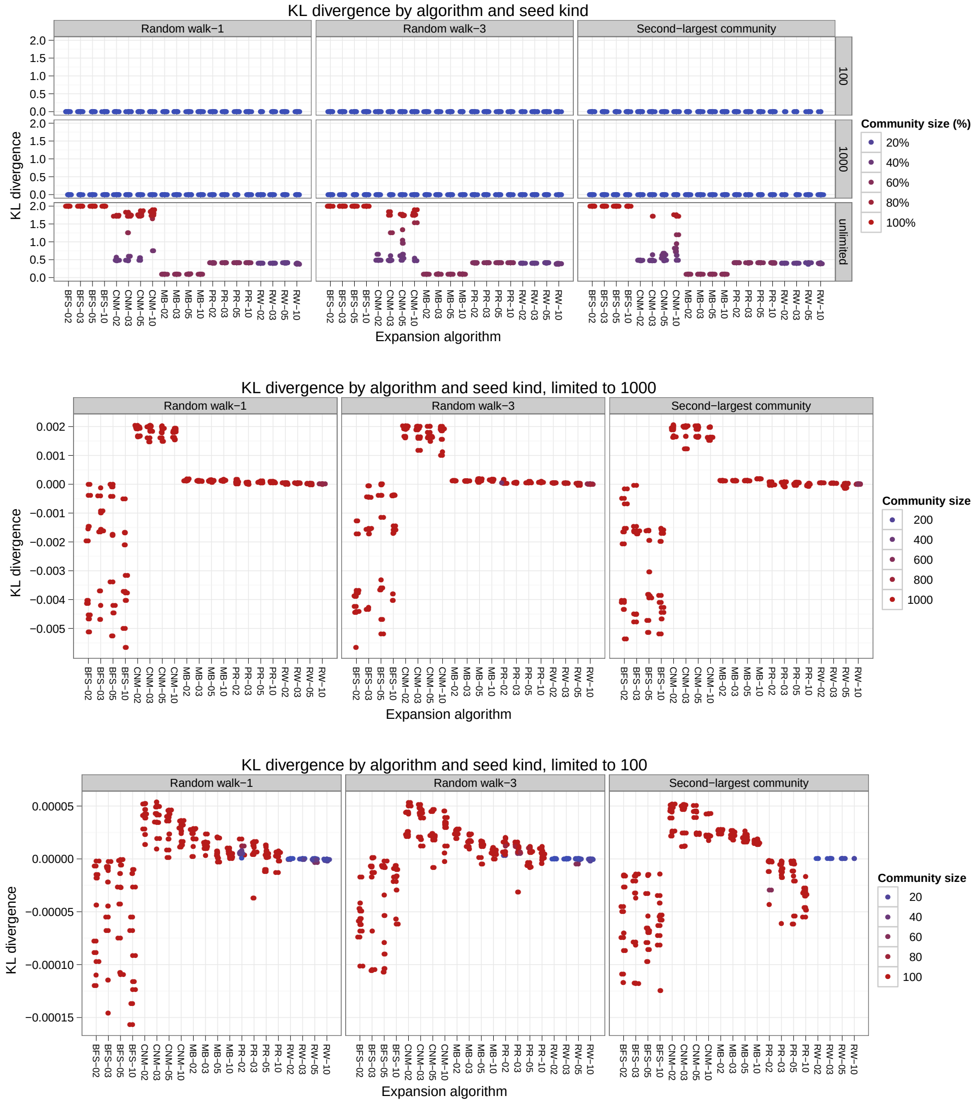


Figure 6: Result community Kullback-Leibler divergence from random graph with same degrees

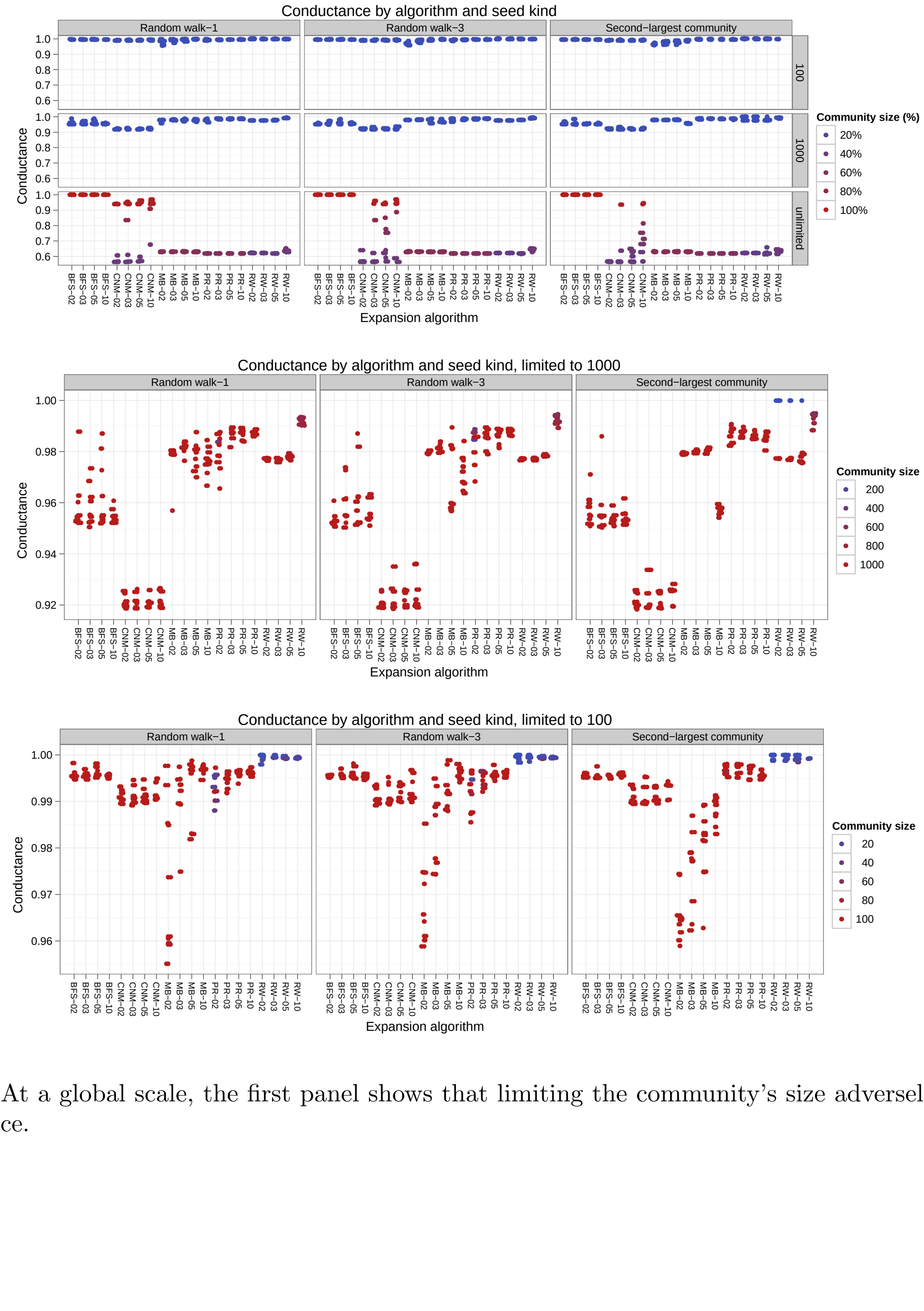


Figure 7: At a global scale, the first panel shows that limiting the community's size adversely affects the conductance.

The top panel of Figure 7 shows that limiting the community size adversely affects the conductance. In the unlimited case, CNM produces low conductance communities when it produces small communities. MB, PR, and RW all produce relatively small conductance communities. Unrestricted BFS communities leave only a handful of vertices or none in $V \setminus S'$ and have trivially large conductance.

With size limits, all the resulting communities have relatively similar conductance. The maximum possible conductance is one[1], and the largest difference from that maximum is a little more than 8%. Within this small range, both BFS and CNM produce surprising low conductance communities for the limit of 1 000 vertices. At the 100 vertex limit, MB produces the largest (but still tiny) range of conductance values.

Figure 8 shows the cut sizes, the numerator in the conductance. The wide variation in cut sizes shows that the small variation in Figure 7's conductance values reflects both the cut size and the limited communities' volumes. For limited growth cases, MB appears to produce smaller cut sizes.

5.4 Clustering coefficients

The clustering coefficients in Figure 9 show that even when CNM grows to almost the entire graph, the resulting community's clustering coefficients is slightly larger than the component's clustering coefficient of 5.74×10^{-2}. When limited, both BFS and CNM produce "more clustered" communities than the graph as a whole. MB's communities have lower clustering coefficients in general, and both PR and RW vary across the graph's clustering coefficient. The RW algorithm sometimes produces very small clustering coefficients, while other times the community is more clustered than the graph.

When limited, both BFS and CNM produce "more clustered" communities than the graph as a whole. MB's communities have lower clustering coefficients in general, and both PR and RW vary across the graph's clustering coefficient. The RW algorithm sometimes produces very small clustering coefficients, while other times the community is more clustered than the graph.

6 Conclusion and Future Work

Our new, greedy algorithm for growing communities around a small handful of seeds produces interesting and sometimes surprising results. When run without a vertex size limit, the MB algorithm produces moderately sized components with surprisingly low conductance and unfortunately low modularity for a modularity-maximizing algorithm. The PR-generated communities are around the same size with higher modularity but also low conductance. The CNM algorithm risks producing vastly larger communities but also often produces larger modularity values and low conductance.

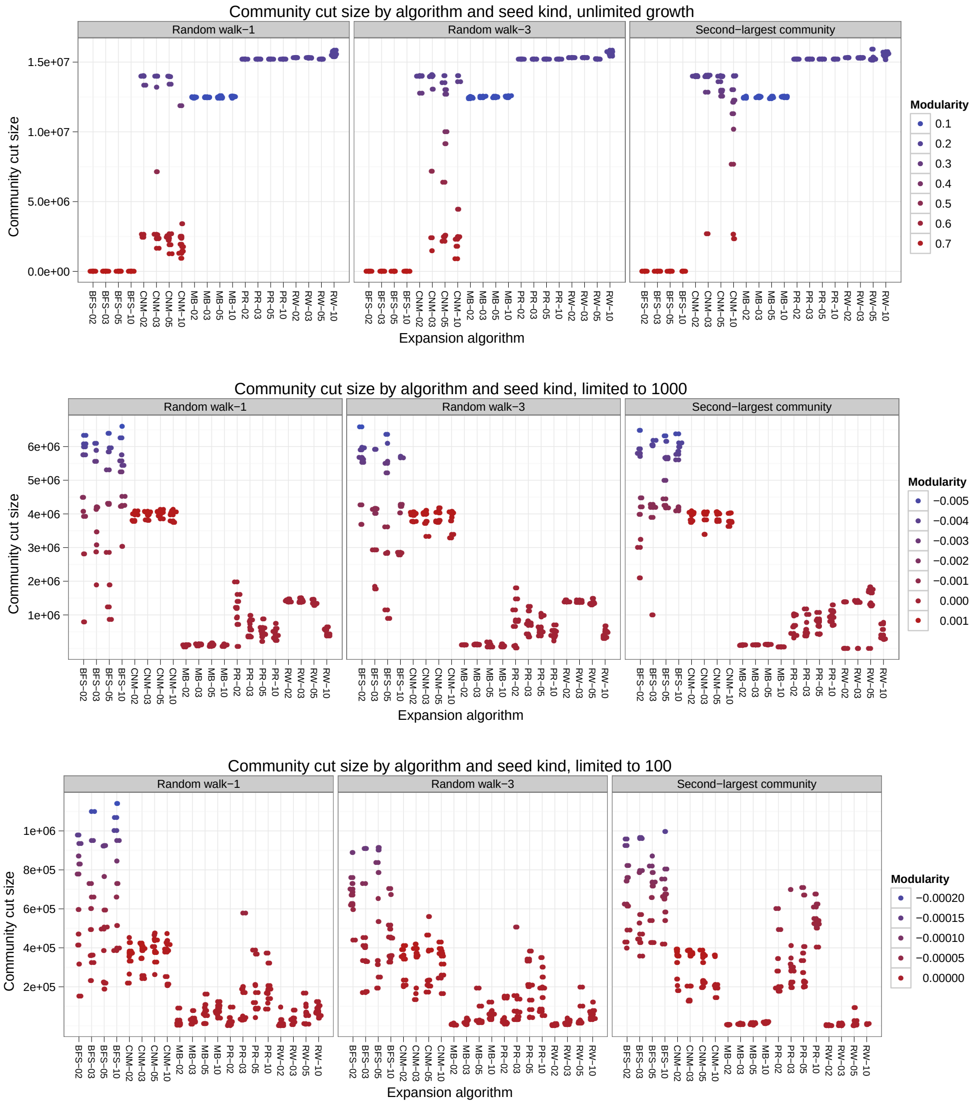
When limited to 1 000 or 100 vertices, a region-growing algorithm maximizing modularity with the CNM criteria produces maximum sized communities with larger modularity and lower conductance than other methods. The resulting communities also have higher clustering coefficients than the original graph.

A trivial BFS-based seeded community growth algorithm performs poorly in general. RW without the extra improvements from [2] does not perform as well as other algorithms. The additional improvements may remain necessary in seeded community detection.

Our comparison between seed set algorithms is, to our knowledge, the first such comparison to focus on a small number of seeds. Human queries are likely to generate such small seed sets. The final communities may be useful for additional analysis or simply for caching within a larger graph database. No single algorithm dominates the results, so the intended use must guide the choice of algorithm.

Future studies will include the improvements mentioned above along with limits on the communities' volumes and maximum degrees. The sensitivity to algorithm parameters and graph size also is of great interest. The PR method requires a regularization parameter α which may greatly change the results. Similarly, the distance from the mean used in MB likely affects the community size. The MB algorithm uses a simple statistical model, and a more refined model targeting particular data sources may perform better. Optimizing and parallelizing the implementations will allow a reasonable comparison of execution time and assist our primary goal, tackling massive social networks and their emergent structures.

[1] Defining $0/0 = 1$ for the limiting case here.


Community cut size by algorithm and seed kind, unlimited growth

Community cut size by algorithm and seed kind, limited to 1000

Community cut size by algorithm and seed kind, limited to 100

Figure 8: The wide variation in BFS cut sizes at smaller limits shows Figure 7's conductance values reflect both the cut size and the communities' volumes.

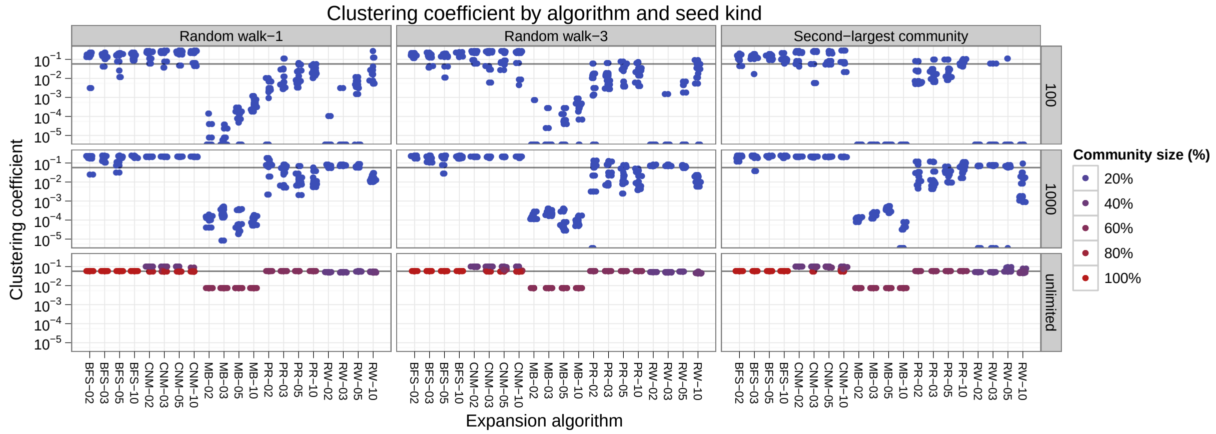


Figure 9: Community clustering coefficients relative to the component's clustering coefficient

Acknowledgments

This work was supported in part by the Pacific Northwest National Laboratory CASS-MT Center, Intel Corporation, and NSF Grant CNS-0708307. Thanks to Henning Meyerhenke for many useful suggestions and edits. Post-processing and plot production use the R statistical package[24] and the igraph[11] and ggplot2[30] packages for R.

References

[1] R. Andersen, F. Chung, and K. Lang. Local graph partitioning using pagerank vectors. In *FOCS '06: Proc. of the 47th Annual IEEE Symp. on Foundations of Computer Science*, pages 475–486, Los Alamitos, CA, USA, Oct. 2006. IEEE Computer Society.

[2] R. Andersen and K. Lang. Communities from seed sets. In *Proc. of the 15th Int'l Conf. on World Wide Web*, page 232. ACM, 2006.

[3] D. Bader and K. Madduri. SNAP, small-world network analysis and partitioning: an open-source parallel graph framework for the exploration of large-scale networks. In *Proc. Int'l Parallel and Distributed Processing Symp. (IPDPS)*, 2008.

[4] D. Bader and J. McCloskey. Modularity and graph algorithms. Presented at UMBC, Sept. 2009.

[5] J. Bagrow. Evaluating local community methods in networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008:P05001, 2008.

[6] J. Berry., B. Hendrickson, R. LaViolette, and C. Phillips. Tolerating the community detection resolution limit with edge weighting. *ArXiv e-prints*, 2009.

[7] B. Bollobás. *Modern Graph Theory*. Springer, July 1998.

[8] D. Chakrabarti, Y. Zhan, and C. Faloutsos. R-MAT: A recursive model for graph mining. In *SIAM Data Mining*, volume 6, 2004.

[9] A. Clauset. Finding local community structure in networks. *Physical Review E*, 72(2):26132, 2005.

[10] A. Clauset, M. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70(6):66111, 2004.

[11] G. Csárdi and T. Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.

[12] N. Du, B. Wu, X. Pei, B. Wang, and L. Xu. Community detection in large-scale social networks. In *Proc. of the 9th WebKDD and 1st SNA-KDD Workshop on Web Mining and Social Network Analysis*, pages 16–25. ACM, 2007.

[13] Facebook. User statistics, July 2010.

[14] S. Fortunato and M. Barthélemy. Resolution limit in community detection. *Proc. of the National Academy of Sciences*, 104(1):36–41, 2007.

[15] M. Girvan and M. Newman. Community structure in social and biological networks. *Proc. of the National Academy of Sciences*, 99(12):7821, 2002.

[16] J. Hopcroft, O. Khan, B. Kulis, and B. Selman. Natural communities in large linked networks. In *Proc. of the ninth ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, pages 541–546. ACM New York, NY, USA, 2003.

[17] K. Jiang, D. Ediger, and D. A. Bader. Generalizing k-Betweenness centrality using short paths and a parallel multithreaded implementation. In *The 38th Int'l Conf. on Parallel Processing (ICPP)*, Vienna, Austria, Sept. 2009.

[18] J. Kleinberg, S. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. The web as a graph: Measurements, models and methods. *Lecture Notes in Computer Science*, 1627/1999:1–17, 1999.

[19] S. Kullback and R. A. Liebler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[20] F. Luo, J. Wang, and E. Promislow. Exploring local community structures in large networks. *Web Intelligence and Agent Systems*, 6(4):387–400, 2008.

[21] M. Newman. Modularity and community structure in networks. *Proc. of the National Academy of Sciences*, 103(23):8577–8582, 2006.

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

[23] D. Price. Networks of scientific papers. *Nuovo Cimento*, 5:199, 1957.

[24] R Development Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2010. ISBN 3-900051-07-0.

[25] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. Defining and identifying communities in networks. *Proc. of the National Academy of Sciences*, 101(9):2658, 2004.

[26] E. Ravasz, A. Somera, D. Mongru, Z. Oltvai, and A. Barabási. Hierarchical organization of modularity in metabolic networks. *Science*, 297(5586):1551, 2002.

[27] M. Rosvall and C. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proc. of the National Academy of Sciences*, 105(4):1118, 2008.

[28] S. Wasserman and K. Faust. *Social network analysis: Methods and applications.* Cambridge Univ Pr, 1994.

[29] D. Watts and S. Strogatz. Collective dynamics of small world networks. *Nature*, 393:440–442, 1998.

[30] H. Wickham. *ggplot2: elegant graphics for data analysis.* Springer New York, 2009.

[31] B. Yang, W. Cheung, and J. Liu. Community mining from signed social networks. *IEEE Trans. on Knowledge and Data Engineering*, 19(10):1333, 2007.