# Overcoming Resolution Limits in MDL Community Detection

L. Karl Branting
The MITRE Corporation
Hanover, Maryland, USA
lbranting@mitre.org

## ABSTRACT

A popular approach to community detection in networks is to search for partitions that maximize modularity [7]. However, recent work has identified two important limitations of modularity as a community quality criterion: a resolution limit; and a bias towards finding equal-sized communities. Compression-based approaches that search for partitions that minimize description length are a recent alternative to modularity. This paper shows that two compression-based algorithms are themselves subject to a resolution limit, identifies the aspect of each approach that is responsible for the resolution limit, proposes a variant, SGE (Sparse Graph Encoding), that addresses this limitation, and demonstrates on three artificial data sets that (1) SGE does not exhibit a resolution limit on graphs in which other approaches do, and that (2) modularity and the compression-based algorithms, including SGE, behave similarly on graphs not subject to the resolution limit.

## General Terms

Community detection, minimum description length principle, modularity

## 1. INTRODUCTION

Complex networks, such as the Internet, metabolic pathways, and social networks, can often be characterized by a community structure in which related vertices are grouped together. Traditional clustering techniques group vertices by some measure of attribute similarity. More recent research has focused on detection of community structure from graph topology. Under this approach, the input to a community-detection algorithm is a graph in which vertices correspond to individuals (e.g., URLs, molecules, or people) and edges correspond to relationships (e.g., hyperlinks, chemical reactions, or marital and business ties). The output consists of a partition of the graph in which subgraphs correspond to meaningful groupings (e.g., web communities, families of molecules, or social clans).[1]

Community detection algorithms can be viewed as comprising two components: a utility function that expresses the quality of any given partition of a graph; and a search strategy that specifies a procedure for finding a partition that optimizes the utility function. The utility function most prevalent in recent community detection research is the modularity function introduced in [7]:

$$Q = \sum_{1<i\leq m} (w(D_{ii})/l - (l_i/l)^2) \quad (1)$$

where $i$ is the index of the communities, $w(D_{ii})$ is the number of edges in the graph that connect pairs of vertices within community $i$, $l_i = \sum_{j\leq i} w(D_{ij})$, i.e., the number of edges in the graph that are incident to at least one vertex in community $i$, and $l$ is the total number of edges in the entire graph. Modularity formalizes the intuition that community structure is manifested in a higher density of links within groups than between groups.

Because of the shortage of real-world data sets with known community structure, maximum modularity has sometimes even been equated with correct community structure. However, two important weaknesses have been identified in modularity as a community-structure criterion.

First, the group structure that optimizes modularity within a given subgraph can depend on the number of edges in the entire graph in which the subgraph is embedded. Stated differently, modularity is characterized by an intrinsic scale under which $Q$ is maximized when pairs of distinct groups having fewer than $\sqrt{2l}$ edges (where $l$ is the total number of edges in the graph) are combined into single groups [4]. This phenomenon is particularly apparent in ring graphs, i.e., connected graphs that consist of identical subgraphs each connected to exactly two other subgraphs by a single link. For example, in the graph shown in Figure 1 consisting of a ring of 15 squares, modularity is greater when adjacent squares are grouped together than when each square is a separate group.

A second weakness of modularity is that even when the resolution limit is not exceeded, modularity exhibits a bias towards groups of similar size. Intuitively, the sum of the square terms $(l_i/l)^2$, representing the expected number of intra-group edges within community $i$ under the null model,

[1]Some communities can overlap. Membership in such communities is better modeled as attributes of vertices rather than through a partition of the graph [6]. The focus of this paper, however, as in the bulk of community detection research, is on partition-based community structure.


*2nd SNA-KDD Workshop (SNA-KDD2008* August 24, 2008, Las Vegas, Nevada, USA.

is minimized, and $Q$ therefore maximized, when all $l_i$ are as nearly equal in size as possible.

One approach to the resolution limit of modularity is apply modularity recursively, so that the coarse structure found at one level is refined at lower levels [12].[2] An alternative approach is to substitute a different community-quality criterion for modularity.

One such alternative criterion for community quality that has recently been proposed is based on *compression*, that is, on minimizing description length [11, 2]. In this approach, the quality of a given partition of a graph is a function of the complexity of the community structure together with the mutual information between the community structure and the graph as a whole. The best community structure is one that minimizes the sum of (1) the number of bits needed to represent the community structure plus (2) the number of bits needed to represent entire graph given the community structure. Under this approach, the task of community detection is reduced to the problem of finding the community structure that leads to the minimum description length (MDL) representation of the graph, where description length is measured in number of bits.

The minimum description length criterion for community-structure quality does not in itself specify how to encode either the community structure or the graph given the community structure. Indeed, the close connection between MDL and Kolmogorov complexity [13], which is undecidable, suggests that MDL may itself be undecidable. The encoding algorithms of Rosvall and Bergstrom [11] (hereinafter "RB") and Chakrabarti [2] (hereinafter "AP", standing for "AutoPart") use quite different approaches to measuring the description length of community structures and graphs. As shown below, however, RB and AP have in common that both are characterized by a resolution limit similar to that observed in modularity.

This paper compares the encoding schemes of RB and AP, identifying the aspect of these schemes that is responsible for the resolution limit, proposes an alternative encoding, termed SGE (Sparse Graph Encoding) that addresses this limitation, and describes an empirical evaluation comparing RB, AP, SGE, and modularity on three classes of artificial data: rings of communities; uniform random graphs; and embedded BarrabasiAlbert (EBA) graphs, that is, graphs consisting of communities generated by the Barrabasi-Albert algorithm embedded in a sparse random graph. In this evaluation, SGE was least affected by the resolution limit, incorrectly combining adjacent communities in community rings only when the ring size was extremely large. By contrast, the other algorithms combining adjacent communities in a wide range of ring sizes. No one algorithm consistently outperformed the others in EBA or uniform random graphs, but modularity was consistently worse than the MDL algorithms on highly skewed uniform random graphs

## 2. MDL ENCODINGS

RB and AP decompose the task of encoding a graph and its community structure into similar steps, but they calculate the bits in each term differently. For the purposes of this comparison, the following notation will be followed:

- $n$ - the number of vertices in the graph
- $m$ - the number of groups
- $a_i$ - the number of vertices in group $i$
- $l$ - the total number of edges in the entire graph
- $l_i$ - the number of edges incident to group $i$
- $D_{ij}$ - a binary adjacency matrix between groups $i$ and $j$
- $n(D_{ij})$ - the number of elements in adjacency matrix $D$
- $w(D_{ij})$ - the number of 1's in $D_{ij}$, i.e., the number of edges between groups $i$ and $j$
- $P(D_{ij})$ - the density of 1's in $D_{ij}$, i.e., $\frac{w(D_{ij})}{n(D_{ij})}$
- $P'(D_{ij})$ - for a square matrix $D_{ij}$, the density of 1's ignoring the diagonal
- $H(D_{ij}) = -P(D_{ij})\log(P(D_{ij})) - (1-P(D_{ij}))\log(1-P(D_{ij}))$, i.e., the mean entropy of $D_{ij}$
- $H'(D_{ij}) = -P'(D_{ij})\log(P'(D_{ij}))-(1-P'(D_{ij}))\log(1-P'(D_{ij}))$, i.e., the mean entropy of $D_{ij}$ if values on the diagonal of $D_{ij}$ are ignored
- B - a matrix representing for each pair of groups whether the pair is connected, i.e., $B_{ij} = 1 \iff w(D_{ij}) > 0$

The encoding schemes used in RB and AP are as follows:

1. Bits needed to represent the number of vertices in the graph. Since this term doesn't vary with differing community structure, it is irrelevant to the choice between different community structures and can be ignored.
2. Bits needed to represent the number of groups.
   - RB. Not explicitly represented.
   - AP. $\log^*(m)$. $\log^*(x) = log_2(x) + log_2log_2(x) +$ ... where only positive terms are included in the sum. This series is apparently intended to represent the mean coding length of integers given that the probability of an integer of a given length is a monotone decreasing function of the integer's length, i.e., longer integers are less probable, but no maximum length is known [9].
3. Bits needed to represent the association between vertices and groups
   - RB. $n\log(m)$. The rationale appears to be that for each of the $n$ vertices, $log(m)$ bits are needed to identify the group to which the vertex belongs.
   - AP. If the groups are placed in decreasing order of length, i.e., $a_1 \geq a_2 \geq ... \geq a_m \geq 1$,
     $$\sum_{i=1}^{m-1} \lceil \log(\overline{a}_i) \rceil$$
     where $\overline{a}_i = (\sum_{t=1}^{m} a_t) - m + i$.
4. Bits needed for the group adjacency matrix, i.e., the number of edges between pairs of groups.

[2] See [10] for recent approach that addresses resolution limits by using an absolute evaluation of community structure rather than comparison to a null model.

- RB. $\frac{1}{2}m(m+1)\log(l)$. The first term ($\frac{1}{2}m(m+1)$) represents the number of pairs of groups, and the second term ($\log(l)$) the number of bits needed to specify the number of edges between any pair of groups.
- AP.

$$\sum_{1<i,j<m} \lceil \log(a_i a_j + 1) \rceil$$

This expression sums for every pair of groups sufficient bits to represent the number of edges between that pair.

5. Bits needed to represent the full adjacency matrix for vertices, given the group structure represented in terms 2-4.
   - RB.

$$\log(\prod_{i=1}^{m} \left( \begin{array}{c} a_i(a_i-1)/2 \\ w(D_{ii}) \end{array} \right) \prod_{i<j} \left( \begin{array}{c} a_i a_j \\ w(D_{ij}) \end{array} \right)$$

The expression following the first product sign represents the number of ways to choose the actual pairs that are connected within a single group from the set of all possible pairs. The expression following the second product sign is the number of ways to choose the actual pairs between vertices in two different groups from the set of possible edges between vertices in those groups.

   - AP.

$$\sum_{i=1}^{m}\sum_{j=1}^{m} a_i a_j H(D_{ij})$$

For each pair of groups, the entropy of the adjacency matrix for that pair, i.e., the size of the matrix times its the mean entropy.

RB and AP clearly calculated each term quite differently. In general, RB uses encodings that are much larger than those used in AP. However, a key similarity is in term 4, the bits needed to encode the number of edges between pairs of groups. In both RB and AP at least one bit is required for each pair of groups regardless of how few groups are actually connected (i.e., how few pairs of groups have at least one edge from a vertex in one to a vertex in the other). The number of bits arising from this term therefore increases with the square of the number of groups, regardless of the sparsity of their interconnections. One would expect that for sufficiently large graphs with sparse community structure the savings in term 4 from combining groups would be greater than the added cost in term 5 of specifying the vertex adjacencies for the resulting relatively sparse group, and that this would lead to conflation of distinct groups similar to that observed when modularity is used as a community quality function. As discussed below, this conflation is in fact observed. For example, RB and AP combine adjacent groups when applied to the graph shown in Figure 1.

## 3. SPARSE GRAPH ENCODING (SGE)

The observations that RB and AP (1) assign at least one bit per pair of communities, regardless of how few are actually connected and (2) conflate distinct groups in large sparse graphs (as shown experimentally below) suggests the hypothesis that an encoding in which the bits required to encode the number of edges between pairs of groups grow more slowly than the square of the number of groups would be less prone to the resolution limit. SGE is an encoding scheme designed to test this hypothesis.

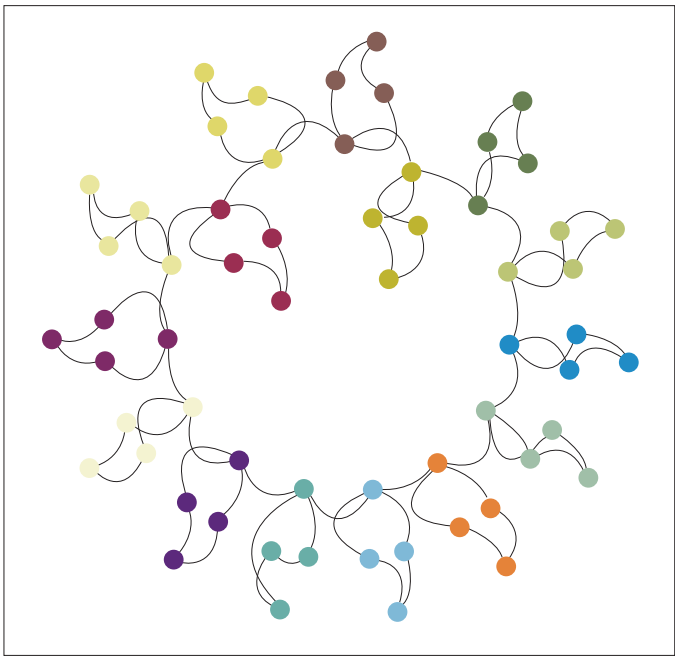

**Figure 1: Ring graph $R_{15,4}$ consisting of 15 communities, each containing 4 vertices.**

The key idea is to encode the group adjacency matrix using two terms. The first term encodes, for each pair of groups, whether the groups are connected. The number of bits required for this is equal to the entropy of B, the binary matrix representing for each pair of groups whether those groups are connected. The mean entropy of B is at most 1.0, if each group is randomly connected to exactly half the others. If few, or most, groups are connected to one another, the mean entropy is less than 1.0, and the total entropy is therefore less than the square of the number of groups.

Moreover, the number of bits needed to represent B can be further reduced by noting that the value of B's diagonal need not be explicitly represented because it can be determined from the number of nodes in each group. Singleton groups have no within-group edges (assuming that self-loops are prohibited) and groups with more than one element must have at least one within-group edge (if there are no within-group edges, the density of within-group edges can not be higher than the density of between-group edges, the basic characteristic of a group).

The bits needed to represent B are therefore:

$$m(m-1)H'(B) \tag{2}$$

where $H'(B) = -P'(B)\log(P'(B)) - (1-P'(B))\log(1-P'(B))$ and $P'(B)$ is the density of 1's in B, ignoring the diagonal.

The second term contains, for each connected pair, the number of bits needed to represent the number of edges be-

tween that pair (the second sum is needed if, as we assume, edges from a vertex to itself are forbidden):

$$\sum_{i \neq j \wedge w(D_{ij}) \geq 0} \log(a_i a_j) + \sum_{i=j \wedge w(D_{ij})>0} \log(a_i(a_j - 1)) \quad (3)$$

If the cost of representing the group adjacent matrix is calculated as expressions 2 + expression 3, the cost will grow with the number of connected pairs rather than with the total number of pairs.

SGE employs several additional minor modifications to reduce the description length. The entire calculation is as follows:

1. Bits needed to represent the number of vertices in the graph. As with RB and AP, these bits are ignored.

2. Bits needed to represent the number of groups. The log* function of (Rissancn 1983) used in AP is predicated on the assumption that no maximum integer size is known a priori. Here, however, the maximum number of groups is bounded by both the machine word size and the virtual memory size of the machine on which the algorithm is executed. Therefore, SGE uses instead RB' s calculation:

$$\log(m)$$

3. Bits needed to represent the association between vertices and groups. No group can contain more than $n-m+l$ vertices (since each group must have at least one vertex). Accordingly, the following expression contains sufficient bits to represent the number of vertices in all m groups:

$$m \log(n - m + 1)$$

4. Bits needed for the group adjacency matrix, i.e., the number of edges between pairs of groups. As discussed above, the number of bits is:

$$H'(B) + \sum_{i \neq j \wedge w(D_{ij})>0} \log(a_i a_j) + \sum_{i=j \wedge w(D_{ij})>0} \log(a_i(a_j - 1))$$

5. Bits needed to represent the full adjacency matrix for vertices given the group structure represented in terms 2-4. This consists, for every pair of groups $i$ and $j$, of size of the $i, j$ adjacency matrix, $a_i a_j$, times the entropy per entry in the corresponding binary matrix, $H(D_{ij})$. This is equivalent to the AP calculation, shown above:

$$\sum_{i=1}^{m} \sum_{j=1}^{m} a_i a_j H(D_{ij})$$

In summary, the relationship between SGE, RB, and AP is as follows:

1. Bits needed to represent the number of vertices in the graph. Ignored as in RB and AP.

2. Bits needed to represent the number of groups. Follows RB.

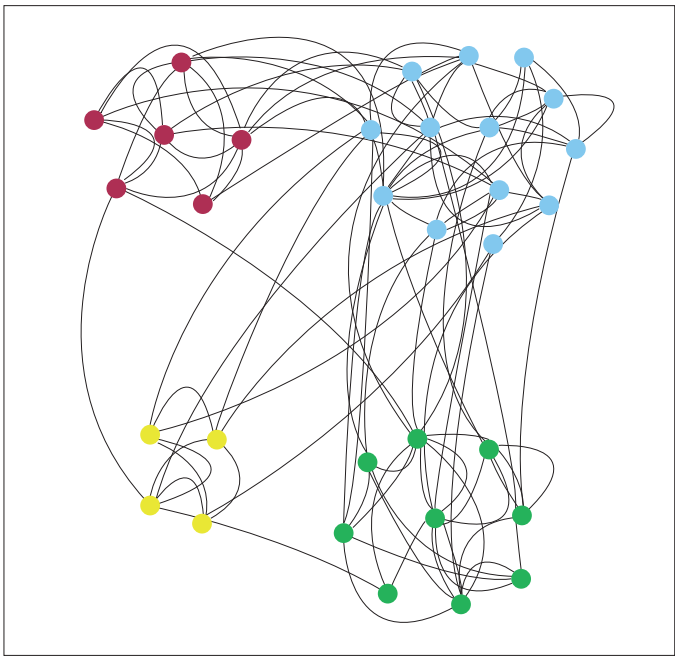

**Figure 2: A uniform random graph with 32 vertices, 4 groups, size ratio 1.25, and io ratio 0.67. "Mod" is modularity, "RS" is Rosval-Bergstrom, "AP" is AutoPart, and "SGE" is "sparse graph encoding."**

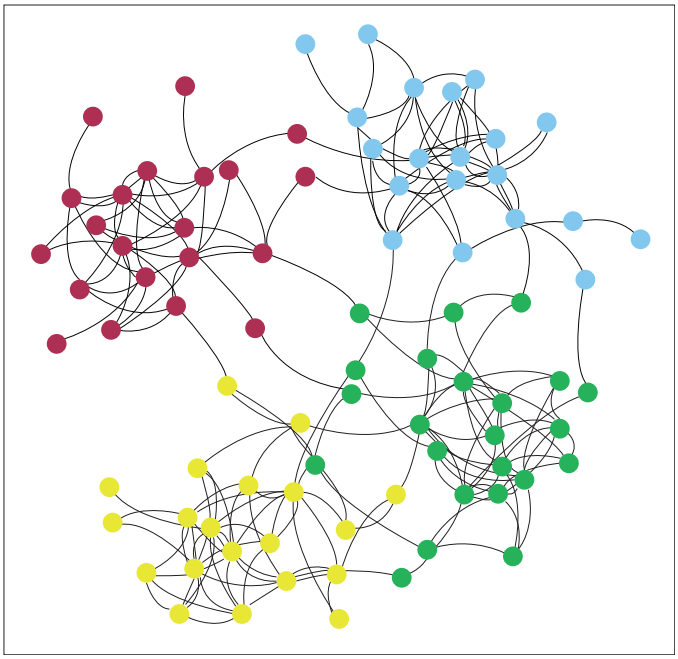

**Figure 3: An Embedded Barabasi-Albert (EBA) graph with 4 communities, each with 5 initial vertices per community, 3 new edges per time step, 10 time steps, and 25 singleton-group edges.**

3. Bits needed to represent the association between vertices and groups. Uses an expression with fewer bits than that used in RB, and that is simpler than that used in AP.

4. Bits needed for the group adjacency matrix. The primary novelty of SGE, in that for sparse adjacency matrices this term grows more slowly than the square of the number of groups.

5. Bits needed to represent the full adjacency matrix for vertices. Follows AP.

# 4. EXPERIMENTAL PROCEDURE

To test the hypothesis that the observed resolution limit would be reduced by changing the calculation of the number bits required to represent the group adjacency matrix to an expression that grows more slowly than the square of the number of groups, the performance of SGE was compared to that of RB, AP, and modularity on three different artificial data sets. In all tests, the search strategy selected was the greedy divisive clustering procedure that iteratively removes the edge with the highest betweenness centrality described in [7]. The evaluation criterion was applied to each partition created during this procedure and the partition with the optimal value (highest value for modularity, lowest for MDL criteria) was returned by the procedure. Using a single search strategy removes the potentially confounding disparity of the search algorithms used in published descriptions of RB, AP, and modularity. For example, [11] used simulated annealing as their search strategy, whereas [2] used a separate hill-climbing procedure for each value of m. Modularity has been combined with various search strategies, including both greedy agglomerative [7] and greedy divisive hierarchical clustering [7].

The artificial data sets created for the evaluation were as follows:

- **Ring graphs.** Ring graph $R_{m,c}$ consists of $m$ communities, each consisting of a ring of $c$ vertices, connected to two other communities each by a single link such that all communities are connected. Ring graphs are similar to the clique rings of [4] but differ in that the individual communities are themselves rings rather than cliques. For example, Figure 1 depicts ring graph $R_{15,4}$.

  The evaluation compared RB, AP, SGE, and modularity on 91 ring graphs for which $\langle m, c\rangle \in \{4 \ldots 16\} \times \{3 \ldots 9\}$.[3]

- **Uniform random graphs.** This data set was intended to duplicate the artificial graphs used in the evaluations set forth in [7] and [11]. The graphs were generated by grouping $n$ vertices into $m$ communities. The relative size of the communities was determined by a size ratio $s$ such that if the groups were placed in ascending order, $\frac{a_{i+1}}{a_i} = s$. The connections among the vertices were determined by the average vertex degree $d$ and in/out ratio $i$ such that the average number of within-community edges incident to each vertex was $i*d$ and the average number of cross-community edges incident to each vertex was $(1-i)*d$. For example, Figure 2 shows a uniform random graph with $s = 1.25$ and $i = 0.6$. Tests were performed for each combination of $n = 32, m = 4, d = 6, s \in \{1.0, 1.25, 1.5, 1.75, 2.0\}$, and $i \in \{0.6, 0.75, 0.9\}$.

- **Embedded Barabasi-Albert Graphs.** A wide range of naturally occurring graphs, including those mentioned in the introduction (the Internet, biochemical pathways, social networks) exhibit a power-law degree distribution that is not present in uniform random graphs [1]. However, few such "scale-free" graphs are annotated with correct community structure. The third data set consists of communities with scale-free structure embedded in a sparse random graph. Each graph consists of $m$ communities generated by the Jung 1.74 implementation of the Barabasi-Albert preferential attachment algorithm, each starting with $i$ initial vertices in each community, with $e$ new edges per time step following the preferential attachment rule of [1] for each of $t$ time steps, together with $c$ singleton-group vertices. The singleton-group vertices were connected to $1 \ldots e$ vertices randomly selected from the entire graph, i.e., including both community and singleton-group vertices. The graphs used tor testing had 4 communities, 4 initial vertices per community, 2–4 edges added per time step, 20 time steps, and 25 singleton-group vertices. For example, Figure 3 depicts an EBA graph with 3 edges added per time step.

Fifty trials were performed under each experimental condition for uniform random and EBA graphs (there is no randomness in the construction of ring graphs, so a single trial was sufficient).

## 4.1 Evaluation Criteria

Various objective functions have been proposed for evaluating the quality of a proposed community structure given the actual correct community structure, including the Rand index [8], the adjusted Rand index [5], and f-measure. There is no consensus regarding the most informative objective function, so f-measure was selected here since its use in information retrieval has made it familiar to a wide range of researchers. The intuition underlying the use of f-measure is that group structure can be expressed as a relation $c(G) = \{\langle v_i, v_j\rangle \mid \exists g \in G \ni v_i, v_j \in g\}$, that is, the community structure can be represented by specifying for each pair of vertices whether that pair is in the same group. The similarity of the proposed to the actual group structure can be evaluated by comparing $c$(proposed) with $c$(actual). One way to make the comparison is to view each pair in c(proposed) that is also in c(actual) as a true positive, whereas each pair in c(proposed) that is not in c(actual) is a false positive. Under this view, recall and precision can be defined as follows:

- Recall = $\frac{|c(proposed)| \cap |c(actual)|}{|c(actual)|}$
- Precision = $\frac{|c(proposed)| \cap |c(actual)|}{|c(proposed)|}$

F-measure is the harmonic mean of recall and precision:

- f-measure = $\frac{2 * recall * precision}{recall + precision}$

[3]Note that for $m, c > 3$ ring graphs contain no triangles. Therefore, community detection techniques based on clustering coefficient, e.g., [3], are ineffective for finding communities in such ring graphs.

In evaluating EBA graphs, singleton-group vertices were ignored, regardless of whether they were grouped into new communities or added to existing communities.

# 5. EXPERIMENTAL RESULTS

## 5.1 Ring graphs

- **SGE.** The partition having the optimal (lowest) SGE had the correct partition (i.e., no separate communities were conflated) in every graph except for $R_{4,3}$ and $R_{13,3}$ In other words, the correct community structure was found all but 2 ring graphs.
- **RB and AP.** The correct community structure was never found by optimizing either RB or AP. In other words, the partition having the optimal (lowest) value for RB and AP contained at least one pair of communities that were grouped together in every ring graph tested.
- **Modularity.** Optimizing modularity led to incorrect community structure for rings of more than 8 triangles, more than 10 squares, more than 11 pentagons, or more that 13 hexagons or heptagons. In other words, the correct partitions were obtained with modularity only for rings and communities of the following sizes:
  - $R_{4,3} - R_{8,3}$
  - $R_{4,4} - R_{10,4}$
  - $R_{4,5} - R_{11,5}$
  - $R_{4,6} - R_{13,6}$
  - $R_{4,7} - R_{13,7}$
  - $R_{4,8} - R_{16,8}$
  - $R_{4,9} - R_{16,9}$

This evaluation confirmed empirically the existence of the resolution limit for modularity derived formally in [4]. The evaluation also showed the surprising result that optimizing RB and AP leads to even more conflation of distinct communities than does modularity. The observation that optimizing SGE led to the correct community structure provides confirmation for the hypothesis that the conflation of communities in RB and AP arises from term 4, which uses more bits than necessary to represent the number of edges connecting groups in sparse graphs. Substituting rings of cliques for rings of graphs that are themselves rings leads to almost identical results to those described here.

## 5.2 Uniform Random Graphs

Figures 4, 5, and 6 show the results of the 4 algorithms on uniform graphs for $i \in \{0.6, 0.75, 0.9\}$ respectively. For $i \in \{0.75, 0.9\}$, in which the community structure is relatively distinct, all four algorithms led to similar results except when the size ratio $s$ was equal to 2.0 (i.e., the sizes of the groups were highly skewed). Under these circumstances, modularity led to much lower f-measure than the other algorithms. When $i$ was equal to 0.6 (i.e., the community structure was relatively unclear) modularity was best and AP worst for low size ratio, and RB and AP were best for high size ratio. These results are consistent with [11], which showed better performance for RB than modularity for skewed community sizes, but comparable performance when community sizes were equal.

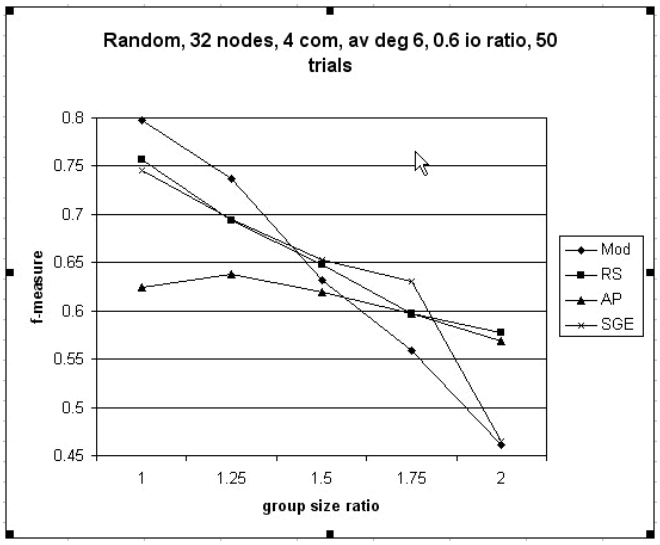


**Figure 4: F-measure for uniform random graphs with i=0.6 (weak community structure).**

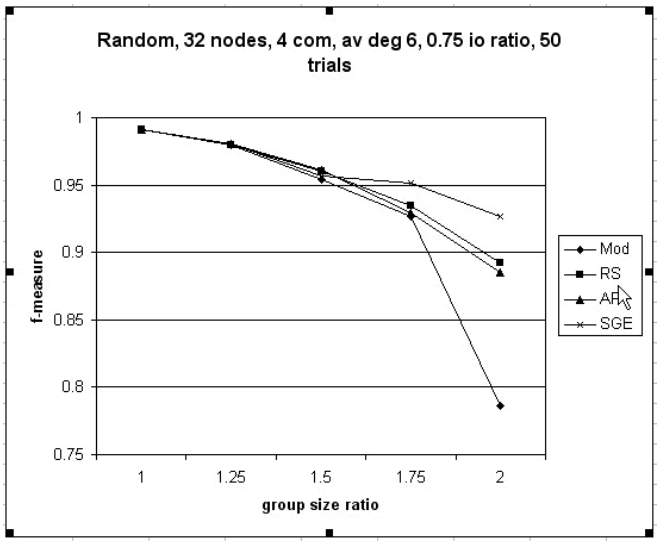


**Figure 5: F-measure for uniform random graphs with i=0.75 (moderate community structure).**

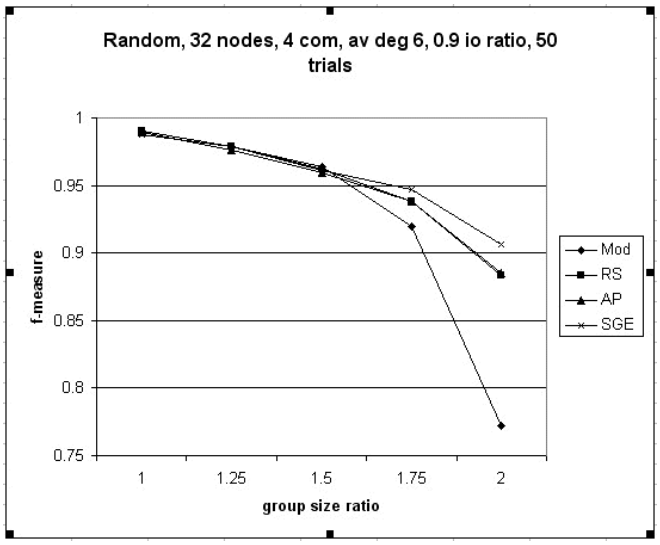


**Figure 6: F-measure for uniform random graphs with i=0.9 (strong community structure).**

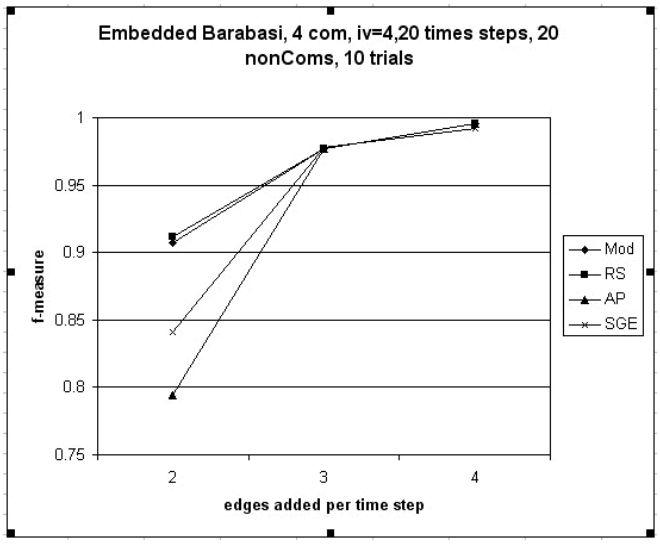


**Figure 7: F-measure for embedded Barabasi-Albert graph with 2–4 edges added per time step.**

## 5.3 EBA Graphs

As shown in 7, the behavior of all four algorithms was quite similar when the number of edges added per time step was 3 or 4, which leads to relatively densely connected graphs. When only 2 edges were added per time step (i.e., the communities where quite sparse), AP's performance was much worse, and SGE's somewhat worse, than that of the other two algorithms.

# 6. CONCLUSION

The empirical evaluation demonstrated that RB and AP conflate distinct communities in ring graphs, and that changing the calculation of the number of bits needed to represent the group adjacency matrix eliminated this conflation over the range of ring graphs tested. Ring graphs are artifacts not likely to occur in many real-world graphs of interest, but there is a risk that a community-quality function that handles ring graphs incorrectly would produce inaccurate or misleading results when applied to more realistic graphs.

No one algorithm consistently outperformed the others in EBA or uniform random graphs, but modularity was consistently worse than the MDL algorithms on highly skewed uniform random graphs, and AP and SGE had lower performance than the others on sparse EBA graphs.

SGE's description length calculation does not entirely eliminate resolution limits in clustering. For example, SGE combines adjacent communities in extremely large rings, such as $R_{100,4}$. Moreover, SGE combines adjacent communities in $R_{3,4}$ and $R_{13,3}$.

While MDL is clearly a powerful tool for identifying community structure, there are many options for MDL encodings, and the consequences of each choice can be difficult to anticipate. SGE demonstrates that one limitation of RB and AP can be easily addressed, but the fact that SGE did not, on the whole, outperform RB or RB on other types of graphs suggests that considerable subtlety is required to identify the MDL encoding most effective over a wide range of graph and community types.

# 7. ACKNOWLEDGMENTS

This work was funded under contract number CECOM Wl5P7T-08-C-F600. The MITRE Corporation is a nonprofit Federally Funded Research and Development Center chartered in the public interest.

# 8. REFERENCES

[1] A. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, Oct 15 1999.

[2] D. Chakrabarti. Autopart: Parameter-free graph partitioning and outlier detection. In *Proceedings of the European Conference on Machine Learning and Practice of Knowledge Discovery in Databases*, pages 112–124, 2004.

[3] N. Du, B. Wu, X. Pei, B. Wang, and L. Xu. Community detection in large-scale social networks. In *WebKDD/SNA-KDD '07: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 16–25, New York, NY, USA, 2007. ACM.

[4] S. Fortunato and M. Barthelemy. Resolution limit in community detection. *PROC.NATL.ACAD.SCI.USA*, 104:36, 2007.

[5] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.

[6] P. Koutsourelakis and T. Eliassi-Rad. Finding mixed-memberships in social networks. In *Papers from the 2008 AAAI Spring Symposium on Social Information Processing, Technical Report WW-08-06*, pages 48–53. AAAI Press, 2008.

[7] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.

[8] W. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846–850, 1971.

[9] R. Rissanen. A universal prior for integers and estimation by minimum description length. *The Annals of Statistics*, 2:416–431, 1983.

[10] P. Ronhovde and Z. Nussinov. An improved potts model applied to community detection. *physics.soc-ph*, 2008.

[11] M. Rosvall and C. Bergstrom. An information-theoretic framework for resolving community structure in complex networks. *Proc Natl Acad Sci USA*, 104(18):7327–7331, May 2007.

[12] J. Ruan and W. Zhang. Identifying network communities with a high resolution. *PhysRevE*, 2007.

[13] C. S. Wallace and D. L. Dowe. Minimum message length and Kolmogorov complexity. *The Computer Journal*, 42(4):270–283, 1999.