# Edge-Ratio Network Clustering by Variable Neighborhood Search

Sonia Cafieri, Pierre Hansen, Nenad Mladenovic

**Edge-Ratio Network Clustering by
Variable Neighborhood Search**

S. Cafieri, P. Hansen,
N. Mladenović

# Edge-Ratio Network Clustering by
# Variable Neighborhood Search

## Sonia Cafieri

*Lab. MAIAA, École Nationale de l'Aviation Civile*
*Toulouse, France*

sonia.cafieri@enac.fr

## Pierre Hansen

*GERAD & HEC Montréal*
*Montréal (Québec) Canada, H3T 2A7*

pierre.hansen@gerad.ca

## Nenad Mladenović

*GERAD & School of Mathematics*
*Brunel University*
*West London, United Kingdom*

nenad.mladenovic@brunel.ac.uk

December 2013

**Abstract:** The analysis of networks and in particular the identification of communities, or clusters, is a topic of active research with application arising in many domains. Several models were proposed for its solution. In [Cafieri et al., Phys. Rev. E 81(2):026105, 2010], a criterion is proposed for a graph bipartition to be optimal: one seeks to maximize the minimum for both classes of the bipartition of the ratio of inner edges to cut edges (*edge ratio*), and it is used in a hierarchical divisive algorithm for community identification in networks. In this paper, we develop a VNS-based heuristic for hierarchical divisive edge ratio network clustering. A $k$-neighborhood is defined as move of $k$ entities, i.e., $k$ entities change their membership from one to another cluster. A local search is based on 1-changes and $k$-changes are used for shaking the incumbent solution. Computational results on datasets from the literature validate the proposed approach.

**Résumé :** L'analyse de réseaux et en particulier l'identification de communautés, ou classes, est un sujet de recherche très actif dont les applications sont nombreuses dans de multiples domaines. Plusieurs modèles ont été proposés pour sa résolution. Dans [Cafieri et al., Phys. Rev. E 81(2):026105, 2010], on propose un critère pour que la bipartition d'un graphe soit optimale : on cherche à maximiser le minimum pour les deux classes de la bipartition du rapport du nombre d'arêtes internes au nombre d'arêtes coupées (*edge-ratio*). Ce critère est utilisé dans un algorithme hiérarchique divisif pour l'identification de communautés dans les réseaux. Dans le présent article, nous développons une heuristique pour la classification hiérarchique descendante basée sur la métaheuristique de recherche à voisinage variable. Un $k$-voisinage est défini comme le mouvement de $k$ entités, c'est-à-dire que $k$ entités changent leur appartenance d'une classe à l'autre. Une recherche locale est basée sur les 1-échanges et les $k$-échanges sont utilisés pour perturber la meilleure solution connue. Les résultats de calcul sur des données de la littérature valident l'approche proposée.

# 1 Introduction

Complex systems in a variety of domains are represented by networks. The most prominent examples include social networks, describing individuals and their interactions and relationships, telecommunication networks, transportation networks, biological networks, and many more. A modular structure characterizes many complex systems, which contain subgroups of entities sharing some common properties. A topic of particular interest in the study of complex networks is therefore the identification of modules, also called *clusters* or *communities*. This is very useful to identify some properties of the system described by the studied network starting from its structural features. The reader is referred to Newman [1] for an introduction to networks and to Fortunato [2] for a recent and deep survey of network clustering.

Let us consider a graph $G = (V, E)$, with $V$ the set of vertices and $E$ the set of edges, used to represent a network. Several models and clustering criteria have been proposed. One often maximizes or minimizes a criterion function. The most used is *modularity* [3], based on the idea of comparing the fraction of edges falling within communities to the expected fraction of such edges. An alternative approach to the maximization of a criterion function is based on the satisfaction of *a priori* conditions to have a community. Radicchi et al. [4] proposed two such conditions defining communities in a strong and a weak sense, respectively. Recall that the degree $k_i$ of a vertex $i$ belonging to $V$ is the number of its neighbors (or adjacent vertices). Let $S \subseteq V$ be a subset of vertices. Then the degree $k_i$ can be separated into two components $k_i^{in}(S)$ and $k_i^{out}(S)$, i.e., the number of neighbors of $i$ inside $S$ and the number of neighbors of $i$ outside $S$. A set of vertices $S$ forms a community in the *strong sense* if and only if every one of its vertices has more neighbors within the community than outside:

$$k_i^{in}(S) > k_i^{out}(S), \quad \forall i \in S.$$

A set of vertices $S$ forms a community in the *weak sense* if and only if the sum of all degrees within $S$ is larger than the sum of all degrees joining $S$ to the rest of the network:

$$\sum_{i \in S} k_i^{in}(S) > \sum_{i \in S} k_i^{out}(S).$$

This is equivalent to the condition that the number of edges within $S$ is at least half the number of edges in the cut of $S$.

The weak condition was used [4] as a local stopping criterion in hierarchical clustering; also, it led Wang et al. [5] to define a community $S$ *indivisible* if there is no bipartition, $(S_1, S_2)$ of $S$, such that both $S_1$ and $S_2$ satisfy the weak condition.

These concepts inspired the definition of the *edge-ratio* criterion in [6]. More precisely, in [6] the definition of community in the weak sense is extended into a criterion for a bipartition to be optimal: one seeks to maximize the minimum for both classes of the bipartition of the ratio of inner edges to cut edges. Specifically, the ratio of the number of edges within a community to the number of cut edges which have one end point only within that community is considered:

$$r(S) = \sum_{i \in S} k_i^{in}(S) / \sum_{i \in S} k_i^{out}(S).$$

When dividing $S$ in $S_1$ and $S_2$, this ratio for both communities $S_1$ and $S_2$ is considered and the smallest value is maximized:

$$f(S_1, S_2) = \max_{S_1, S_2 \subset V} \min\left(r(S_1), r(S_2)\right). \tag{1}$$

Solving sequentially this problem yields a hierarchical divisive clustering algorithm, with a well defined criterion for bipartitioning. To build the divisive algorithm, the weak definition is first strengthen by quantifying how much the number of inner edges is larger than the number of cut edges. This is done by introducing a parameter $\alpha$ in the weak condition:

$$\sum_{i \in S} k_i^{in}(S) \geq \alpha \sum_{i \in S} k_i^{out}(S). \tag{2}$$

The coefficient $\alpha$, equal to the ratio of twice the number of edges within the community $S$ divided by the number of edges within the cut of that community (($2$) in case of equality), is called the *edge ratio* [6].

So, a divisive algorithm can be built performing each bipartition in such a way to seek the maximum value of $\alpha$ for which the current sub-network $S$ is divisible (i.e., twice the ratio of the number of edges within $S$ divided by the number of edges within the cut of $S$). Bipartitioning is the crucial point in a hierarchical divisive algorithm and usually gives rise to the most computationally demanding steps of the algorithm. The problem of maximizing the edge-ratio for bipartitioning is NP-hard [7]. This motivates studying heuristic methods as an alternative to exact methods, which combine computational efficiency with the ability to provide good quality results.

In this paper we propose an efficient Variable Neighborhood Search based heuristic for divisive hierarchical clustering with edge-ratio criterion. The paper is organized as follows. In Section 2 we present the Variable Neighborhood Search heuristic for the considered optimization problem. In Section 3 we validate our approach on a set of datasets from the literature and compare the obtained results with an exact algorithm for bipartition previously proposed. Section 4 concludes the paper.

## 2    Variable Neighbourhood Search for hierarchical divisive edge-ratio clustering

### 2.1    Variable Neighbourhood Search

Variable Neighbourhood Search (VNS) is a metaheuristic (i.e., a framework for building heuristics), aimed at solving combinatorial and global optimization problems. It relies on iteratively exploring neighbourhoods of the current (local) optimum to identify better local optima, the main idea consisting in a systematic change of neighbourhood combined with a local search [8, 9, 10, 11, 12]. More precisely, VNS iteratively (until a given termination condition is satisfied) escapes from the current local optimum $x^*$ by initiating other local searches from starting points sampled from a neighbourhood of $x^*$ which increases its size iteratively until a local minimum better than the current one is found.

Thus, the main ingredients of the VNS metaheuristic are:

– Definition of a neighbourhood of the current solution;
– Neighbourhood changes;
– Local search;
– Shaking, i.e., a procedure to perturb the current solution;
– Update of the current solution.

### 2.2    Divisive hierarchical clustering

Hierarchical divisive heuristics (see, e.g. [13]) proceed from an initial partition containing all the $n$ vertices of the graph and iteratively divide a community $S$ into two (bipartitioning step) in such a way to improve as much as possible the objective function value. Bipartitions are iterated until a partition into $n$ communities having each a single entity is obtained; in practice, bipartitions can be ended once they do not improve the objective function value anymore. Let $\ell$ be in $\{1, \ldots, n\}$ and $S_\ell$ the community selected for bipartition at each iteration of the algorithm (at the first iteration, $S_\ell = V$). The bipartition of $S_\ell$ gives two new communities, $S_1$ and $S_2$. We let $S_\ell = S_1$ and $S_{\ell+1} = S_2$, then $\ell$ is increased by one and all process repeated. In the present study, the objective function is based on the edge-ratio criterion ($1$) and bipartitioning is done by Variable Neighbourhood Search. A sketch of the divisive algorithm is given in Algorithm 1, where *threshold* is used to decide if the two subgraphs obtained by a bipartition have to be kept (i.e., the current community to be bipartitioned is divisible).

In the remainder of this section, we focus on the step 4 of Algorithm 1 and we present a VNS algorithm to perform community splitting.

**Algorithm 1:** Hierarchical divisive algorithm
**Input:** graph $G = (V, E)$, where $|V| = n$ and $|E| = m$
**Output:** a partition $P$ of $V$

  1    $\ell = 1$, $S_\ell = \{v_1, \ldots, v_n\}$, $P = S_\ell$
  2  **while** $\ell \leq |P|$ and $\exists\, S_\ell \in P$ not visited **do**
  3        select $S_\ell \in P$ (not visited)
  4        find a bipartition of $S_\ell$ into $S_1$ and $S_2$ according to the edge-ratio criterion
  5        **if** $f \geq threshold$
  6          **then**
  7              $P \leftarrow (P \cup \{S_1\} \cup \{S_2\}) \setminus \{S_\ell\}$
  8              $S_\ell \leftarrow S_1$;  $S_{\ell+1} \leftarrow S_2$
  9        **end if**
10        $\ell \leftarrow \ell + 1$
11  **end while**

## 2.3   VNS for edge-ratio

Let us introduce binary variables $x$ (a vector with components $x_i$) to denote to which of the two sets, obtained by the bipartition done at each hierarchy level, vertex $v_i$ belongs for all $i$. By convention, $x_i = 1$ if $v_i$ belongs to $S_1$ and $x_i = 0$ otherwise. To perform community splitting (step 4 of Algorithm 1), we solve an optimization problem with decision variables $x$ and objective function defined in (1).

**Neighborhoods.**   To build a VNS heuristic for this optimization problem, we first have to define a neighborhood $N(x)$ of $x$. In discrete optimization problems with binary variables, neighborhoods of $x$ are obtained from $x$ by some simple modification, for instance complementing one of its components $x_i$, or complementing two complementary components (i.e., setting one component from 1 to 0 and another from 0 to 1).

In a VNS framework, neighborhood changes typically correspond to various types of moves, or perturbations, of the current solution.

Thus, in our VNS algorithm, starting from a random allocation of entities $x_i$ to the two sub-communities of the current community $S$ to be bipartitioned, we use a basic *allocation move* to change neighborhood of $x$. Allocation move works by taking some entity $v_i$ from $S = S_1 \cup S_2$ and changing its allocation: if $v_i \in S_1$ (i.e., $x_i = 1$) then, after the allocation move, it will belong to $S_2$ (i.e., $x_i = 0$) and vice-versa.

When performing such a change (move), we need to check if it leads to some improvement of the objective function value. Notice that, after the move, the new objective function to be maximized for bipartition is:

$$f^{new} = \min \left\{ \frac{2m_1^{new}}{c_1^{new}}, \frac{2m_2^{new}}{c_2^{new}} \right\}, \tag{3}$$

where $m_1^{new}$ and $m_2^{new}$ are the number of edges in the first and in the second subgraph obtained by bipartition, and $c_1^{new}$ and $c_2^{new}$ are the number of cut edges with one end point in the first subgraph and the number of cut edges with one end point in the second subgraph respectively. If $v \in S_1$ changes its membership to $S_2$, then we have:

$$m_1^{new} = m_1 - k_v^{in}, \quad m_2^{new} = m_2 + k_v^{cut} \tag{4}$$

because the new number of inner edges $m_1$ of $S_1$ is reduced by $k_v^{in}$ (the inner degree of $v$) and the number of edges in $S_2$ is augmented by the cut degree of $v$. The number of cut edges $c_1$ and $c_2$ are updated as follows:

$$c_1^{new} = c_1 + k_v^{in} - k_v^{cut} - k_v^{rest} \tag{5}$$
$$c_2^{new} = c_2 + k_v^{in} - k_v^{cut} + k_v^{rest} \tag{6}$$

where $k_v^{rest} = k_v - k_v(S)$ (computed for each vertex at each hierarchy level). Indeed, since $v$ moves to $S_2$, the number of cut edges $c_1^{new}$ of $S_1$ is increased by its inner degree in $S_1$ ($+k_v^{in}$); also, it is reduced by $k_v^{cut}$,

since its cut degree contributes to inner degree of $S_2$; finally, all vertices from $S \setminus (S_1 \cup S_2)$ connected with $v$ after the move to $S_2$ do not produce cut edges of $S_1$ anymore $(-k_v^{rest})$. So, the new objective function value is

$$f^{new} = \min \left\{ \frac{2 \cdot (m_1 - k_v^{in})}{c_1 + k_v^{in} - k_v^{cut} - k_v^{rest}}, \frac{2 \cdot (m_2 + k_v^{cut})}{c_2 + k_v^{in} - k_v^{cut} + k_v^{rest}} \right\} \tag{7}$$

and similarly for the opposite case of $v$ moving from $S_2$ to $S_1$.

**Local search.** A local search procedure is used in the VNS to improve the objective function value. It consists of starting from an initial solution vector $x$ and iteratively moving to the best neighbor $\bar{x} \in N(x)$ such that the value of the objective function $f$ is improved ($f(\bar{x}) > f(x)$ in the case of maximization). The heuristic stops when no such neighbor exists anymore. In our VNS setting, a move is done changing vertex memberships. Thus, the local search is an iterative application of the above allocation move, which is repeated until there is no vertex whose membership change will give partition with larger value of $f(S_1, S_2)$. In that case, a local maximum has been found.

The following proposition holds.

**Proposition 1** *If a bipartition $(S_1, S_2)$ of a set $S = V \setminus (S_1 \cup S_2)$ is locally optimal with respect to allocation neighborhood structure then the following holds:*

$$[(\forall v \in S_1)(f_1^{new} < f_1 \ \lor \ f_1^{new} < f_2)] \ \land \ [(\forall v \in S_2)(f_2^{new} < f_1 \ \lor \ f_2^{new} < f_2)]$$

**Proof.** The fact that there is a move of an entity $v$ from $S_1$ to $S_2$, or from $S_2$ to $S_1$ such that $f^{new}$ is larger than both $f_1$ and $f_2$, may be expressed in the following way:

$$[(\exists v \in S_1)(f_1^{new} > f_1 \ \land \ f_1^{new} > f_2)] \ \lor \ [(\exists v \in S_2)(f_2^{new} > f_1 \ \land \ f_2^{new} > f_2)].$$

If there is no such improvement move, it can formally be presented as negation of the previous expression. Then the statement of the proposition follows from De Morgan's laws.      □

**Proposition 2** *The time complexity of one iteration of the local search algorithm is $O(n)$.*

**Proof.** The proposed local search is based on the move procedure, which is repeated $n$ times. The results follows from the observation that the procedure move is $O(1)$.      □

**Shaking.** Shaking is a procedure used to generate a point $x'$ at random from the k-th neighborhood $N(x)$ of the current solution $x$. A new local search will then started at $x'$.

The natural way to introduce a distance in the solution space of the considered problem is to use Hamming distance. We say that two solutions $x$ and $y$ are at distance $k$ if their Hamming distance is equal to $k$. As solution vectors are binary vectors, the Hamming distance between $x$ and $y$ measures the minimum number of changes of zeros to ones, or reciprocally, required to change $x$ into $y$. We build the shaking procedure on this concept, perturbing the current solution $x$ to a new vector which has Hamming distance $k$ from it. Thus, $k$ zeros or ones are switched to their opposite value, or, in other words, $k$ entities change their original cluster.

**Update of the current solution.** Update of the current solution vector leads to update a number of entities appearing in the problem description. We need indeed to update the number of vertices of $S_1$ and $S_2$, their number of inner edges and cut edges, and correspondingly the degree of vertices of the two subgraphs obtained by bipartition.

The basic VNS algorithm for edge-ratio clustering bipartition can finally be summarized in Algorithm 2.

**Algorithm 2:** Basic VNS for edge-ratio clustering bipartition
**Input:** graph $G_S = (S, E_S)$
**Output:** a bipartition of $S$

```
 1   Initialization
 2   while time ≤ t_max do
 3          k ← 1
 4          while k ≤ k_max do
 5                 apply Shaking
 6                 perform Local search
 7                 k ← k + 1  (next neighborhood)
 8                 if f improved
 9                    then
10                           perform a move
11                 end if
12          end while
13   end while
```

# 3   Computational results and analysis

In this section, we discuss the results obtained on 11 datasets from the literature, corresponding to various real-world applications and often used to test algorithms and heuristics for community detection. In particular, we consider Zachary's karate club dataset [14] describing friendship relationships in a karate club, Lusseau's dolphins dataset [15] describing communications between dolphins, Hugo's *Les Misérables* dataset describing characters in Victor Hugo's masterpiece and their interactions [16], Krebs' political books dataset [17], a dataset representing the schedule of football games between American college teams [18], a network dealing with connections between US airports [19], a dataset on a coauthorship network of scientists working on network theory and experiment [20], a network describing electronic circuits [21], a network representing e-mail interchanges between members of a university [22], a network giving the topology of the Western States Power Grid of the United States [23] and a network of authors collaborations [19]. These datasets are all undirected and unweighted networks without loops. They are listed Table 1 together with their number of vertices $n$ and number of edges $m$.

## 3.1   Comparison with modularity

In Table 1 we also report the number of communities $M$ in partitions obtained by applying the proposed divisive heuristic with the VNS algorithm for bipartitioning. We compare this number with the number of communities obtained when the known criterion *modularity* [3] is used. An exact algorithm for modularity maximization is used [24] for the first 8 datasets, and a heuristic [25] refined using the procedure in [26] is used for the last three datasets, where an exact solution is not known. In the VNS, the value of parameter $k_{max}$ is set to $\min\{50, n/2\}$ in all test problems. We compare results obtained using two treshold values for deciding about *indivisibility* of a community. The default value for this threshold is 1 and, specially for small-scale networks, it allows us to obtain a number of communities that is very close to that obtained using modularity. In other cases, a threshold equal to 1 leads to consider a high number of communities in the final partition, often containing very small communities. In such cases, we experimentally find different values for the threshold which give results very close to results obtained with the modularity criterion. Notice that the impact of the threshold for indivisibility in edge-ratio clustering was not studied in previous work, as only small-scale networks were previously analyzed.

## 3.2   Comparison with a locally optimal hierarchical divisive algorithm

In [6] a mathematical programming formulation for identification of optimal communities according to the edge ratio criterion is proposed, which has a linear objective (i.e., maximization of $\alpha$), but non linear and non convex constraints (products between $\alpha$ and binary variables arise). However, if $\alpha$ is fixed, a linear program

Table 1: Results obtained using VNS, on datasets from the literature

| dataset | $n$ | $m$ | $M(threshold)$ | | $M(threshold)$ | | $M$modularity |
|---|---|---|---|---|---|---|---|
| karate | 34 | 78 | 4 | (1) | - | - | 4 |
| dolphins | 62 | 159 | 8 | (1) | 5 | (1.5) | 5 |
| les miserables | 77 | 254 | 9 | (1) | 7 | (1.5) | 6 |
| political books | 105 | 441 | 10 | (1) | 5 | (2) | 5 |
| football | 115 | 613 | 12 | (1) | 10 | (1.5) | 10 |
| Usair97 | 332 | 2126 | 16 | (1) | 6 | (2) | 6 |
| netscience_main | 379 | 914 | 69 | (1) | 18 | (6.5) | 19 |
| s838 | 512 | 819 | 101 | (1) | 12 | (5) | 12 |
| email | 1133 | 5452 | 39 | (1) | 12 | (1.5) | 11 |
| power | 4941 | 6594 | 1322 | (1) | 48 | (6) | 46 |
| erdos02 | 6927 | 11850 | 268 | (1) | 32 | (1.8) | 34 |

in 0-1 variables is obtained. This suggests to solve the optimal bipartition problem with a dichotomous search on the values of $\alpha$. An initial value $\alpha$ equal to 1 can first be chosen. If there is no feasible solution for that value, the network is indivisible. Otherwise, the value of $\alpha$ may be doubled and feasibility checked until a value is attained for which the weak condition cannot be satisfied. This gives an upper bound $\bar{\alpha}$ and the previous value of $\alpha$ gives a lower bound $\underline{\alpha}$. Then the dichotomous search proceeds by considering the mid value of the interval $[\underline{\alpha}, \bar{\alpha}]$. The procedure stops when the length $\bar{\alpha} - \underline{\alpha}$ of the current interval is smaller than some given tolerance $\epsilon$. This basic procedure can be accelerated in several ways, including using an initial value of $\alpha$ corresponding to a solution obtained by some heuristic and removing symmetries by fixing a variable. The algorithm is locally optimal in the sense that an exact method is used to solve the bipartition problem. See [6] for more details.

In Table 2 we compare, in terms of execution time, the results obtained using VNS and the exact algorithm for bipartition in the splitting step ot the hierarchical divisive algorithm. Threshold 1 is used. Tests were run on AMD Opteron 2 GHz CPU, 128 GB RAM. We remark that VNS allows us to obtain results significantly faster than exact bipartitioning.

The computational time is reduced up to 7 orders of magnitude. Thus, we are able to solve larger problems with respect to problems that we can solve using an exact method. Furthermore, using VNS does not significantly affect the quality of the results. In the most of the cases, we obtain the same partition using VNS and the exact bipartitioning method. Two examples of obtained partitions, and the corresponding dendrogram illustrating the hierarchical divisive algorithm, are shown in Figures 1 and 2. These two partitions, obtained using VNS, are the same obtained using exact bipartitioning.

Table 2: Comparison of results obtained using VNS and an exact algorithm for bipartition, on datasets from the literature

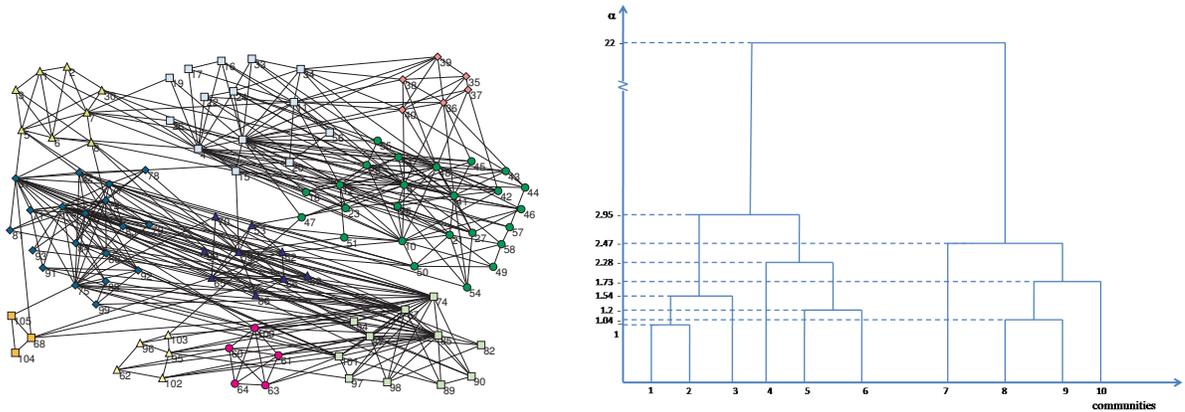| dataset | $n$ | $m$ | Time VNS (sec.) | Time exact (sec.) |
|---|---|---|---|---|
| karate | 34 | 78 | 0.0 | 62.10 |
| dolphins | 62 | 159 | 9.99e-3 | 172.2 |
| les miserables | 77 | 254 | 1.99e-2 | 283.49 |
| political books | 105 | 441 | 1.99e-2 | 716.45 |
| football | 115 | 613 | 3.99e-2 | 11780.79 |
| Usair97 | 332 | 2126 | 0.62 | 3752906.94 |
| netscience_main | 379 | 914 | 1.14 | 2347.24 |
| s838 | 512 | 819 | 0.64 | 12612.53 |
| email | 1133 | 5452 | 6.84 | − |
| power | 4941 | 6594 | 29.80 | − |
| erdos02 | 6927 | 11850 | 35.50 | − |

Figure 1: (Color online) Partition and dendrogram obtained for dataset `polbooks`
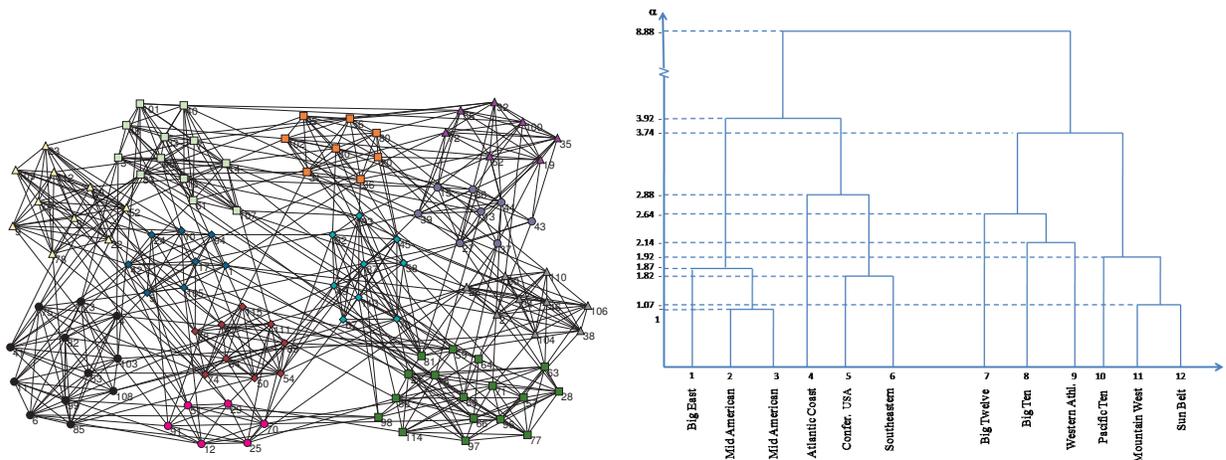


Figure 2: (Color online) Partition and dendrogram obtained for dataset `football`

# 4  Conclusions

In this paper we propose a heuristic based on (basic) Variable Neighborhood Search to perform the bipartitioning step in a hierarchical divisive algorithm based on the recent edge-ratio network clustering criterion, which was shown to be an alternative approach to modularity criterion. Neighborhoods to be used in the VNS are defined using the Hamming distance. Systematic perturbations of the current incumbent solution also use the Hamming distance between any two solution. We develop a local search for the addressed problem and evaluate its complexity.

Computational results show that VNS allows us to obtain good quality results significantly reducing the computational time needed to perform bipartitioning steps.

# References

[1] M. E. J. Newman, *Networks: An Introduction* (Oxford University Press, 2010)

[2] S. Fortunato, Physics Reports **486**, 75 (2010)

[3] M. Newman and M. Girvan, Physical Review E **69**, 026133 (2004)

[4] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, Proceedings of the National Academy of Sciences, USA **101**, 2658 (2004)

[5] J. Wang, Y. Qui, R. Wang, and X. Zhang, Journal of Systems Science and Complexity **21**, 637 (2008)

[6] S. Cafieri, P. Hansen, and L. Liberti, Physical Review E **81**, 026105 (2010)

[7] S. Nobe, P. Hansen, and N. Mladenović, Discrete Applied Mathematics **159**, 2276 (2011)

[8] N. Mladenović and P. Hansen, Computers and Operations Research **24**, 1097 (1997)

[9] P. Hansen and N. Mladenović, European Journal of Operations Research **130**, 449 (2001)

[10] P. Hansen and N. Mladenović, "Variable neighbourhood search," in *Handbook of Applied Optimization*, edited by P. Pardalos and M. Resende (Oxford University Press, Oxford, 2002)

[11] P. Hansen and N. Mladenović, "Variable neighbourhood search," in *Handbook of Metaheuristics*, edited by F. Glover and G. Kochenberger (Kluwer, Dordrecht, 2003)

[12] P. Hansen, N. Mladenović, and J. Pérez, 4OR **6**, 319 (2008)

[13] S. Cafieri, P. Hansen, and L. Liberti, Physical Review E **83**, 056105 (2011)

[14] W. Zachary, Journal of Anthropological Research **33**, 452 (1977)

[15] D. Lusseau, K. Schneider, O. Boisseau, P. Haase, E. Slooten, and S. Dawson, Behavioral Ecology and Sociobiology **54**, 396 (2003)

[16] D. Knuth, *The Stanford GraphBase: A Platform for Combinatorial Computing* (Addison-Wesley, Reading, MA, 1993).

> KEY: knuth
>
> ANNOTATION: Thus, I came to the conclusion that the designer of a new system must not only be the implementer and first large——scale user; the designer should also write the first user manual.
>
> The separation of any of these four components would have hurt TeX significantly. If I had not participated fully in all these activities, literally hundreds of improvements would never have been made, because I would never have thought of them or perceived why they were important.
>
> But a system cannot be successful if it is too strongly influenced by a single person. Once the initial design is complete and fairly robust, the real test begins as people with many different viewpoints undertake their own experiments.

[17] V. Krebs, `http://www.orgnet.com/` (unpublished)

[18] M. Girvan and M. Newman, Proceedings of the National Academy of Sciences, USA **99**, 7821 (2002)

[19] `http://vlado.fmf.uni-lj.si/pub/networks/data/`

[20] M. E. J. Newman, Physical Review E **74**, 036104 (2006)

[21] U. A. Lab, `http://www.weizmann.ac.il/mcb/UriAlon/`

[22] R. Guimerà, L. Danon, A. Diaz-Guilera, F. Giralt, and A. Arenas, Physical Review E **68**, 065103 (2003)

[23] D. Watts and S. Strogatz, Nature **393**, 409 (1998)

[24] D. Aloise, S. Cafieri, G. Caporossi, P. Hansen, L. Liberti, and S. Perron, Physical Review E **82**, 046112 (2010)

[25] A. Noack and R. Rotta, Lecture Notes in Computer Science **5526**, 257 (2009)

[26] S. Cafieri, P. Hansen, and L. Liberti, Discrete Applied Mathematics(2012), DOI 10.1016/j.dam.2012.03.030 (in press)