Les Cahiers du GERAD ISSN: 0711–2440

A Locally Optimal Heuristic for Modularity Maximization of Networks

S. Cafieri, P. Hansen
L. Liberti

G–2011–15

March 2011

A Locally Optimal Heuristic for Modularity Maximization of Networks

Sonia Cafieri

Dept. Mathématiques et Informatique
ENAC
7 Av. E. Belin
31055 Toulouse, France
sonia.cafieri@enac.fr

Pierre Hansen

GERAD
HEC Montréal
3000 Chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada, H3T 2A7
pierre.hansen@gerad.ca

Leo Liberti

LIX
École Polytechnique
91128 Palaiseau, France
liberti@lix.polytechnique.fr

March 2011

Les Cahiers du GERAD
G–2011–15

Abstract

Community detection in networks based on modularity maximization is currently done with hierarchical divisive or agglomerative as well as with partitioning heuristics, hybrids and, in a few papers, exact algorithms. We consider here the case of hierarchical networks in which communities should be detected and propose a divisive heuristic which is locally optimal in the sense that each of the successive bipartitions is done in a provably optimal way. This heuristic is compared with the spectral-based hierarchical divisive heuristic of Newman [Proceedings of the National Academy of Sciences, USA **103**, 8577 (2006)] and with the hierarchical agglomerative heuristic of Clauset, Newman and Moore [Phys. Rev. E **70**, 066111 (2004)]. Computational results are given for a series of problems of the literature with up to 4941 vertices and 6594 edges. They show that the proposed divisive heuristic gives better results than the divisive heuristic of Newman and than the agglomerative heuristic of Clauset et al.

Résumé

L'identification de communautés dans les réseaux se fait actuellement par des heuristiques hiérarchiques agglomératives ou divisives, ansi que par des heuristiques de partitionnement, des hybrides et, dans quelques articles, par des algorithmes exacts. Nous considérons ici le cas de réseaux hiérarchiques dans lequel les communautés doivent être détectées et nous proposons une heuristique divisive qui est localement optimale, dans le sens de ce que chacune des bipartitions successives se fait de manière optimale. Cette heuristique est comparée à l'heuristique hiérarchique divisive spectrale de Newman [Proceedings of the National Academy of Sciences, USA **103**, 8577 (2006)] et à l'heuristique hiérarchique agglomerative de Clauset et al. [Phys. Rev. E **70**, 066111 (2004)]. On donne des résultats de calcul sur une série de problèmes de la litérature. Ils montrent que l'heuristique divisive proposée donne de meilleurs résultats que la précédente heuristique divisive et que l'heuristique agglomérative cités.

1 Introduction

Networks, or graphs, are a powerful and versatile tool for the study of complex systems, with many applications in computer science, engineering, transportation, sociology, political science, biology, chemistry and other fields. A network consists of a set of vertices (or nodes) and a set of edges (or lines). Vertices are represented by points and associated with entities, such as customers, users of the World Wide Web, employees in an organization, transmitters, road-crossings, railway stations and atoms. Edges are pairs of vertices and represented by a line joining them. The shape of this line is irrelevant; only its presence or absence matters. Edges represent relationships between the entities associated with the vertices: communication, collaboration, existence of a connection such as a road or a railway line and chemical bonds. A detailed introduction to networks has recently been given by Newman [41].

A very important and much studied problem in network science and its applications is the detection of *communities* (also called *modules* or *clusters*). These are sets of entities, or vertices, which are likely to have some common function. Usually, the number of inner edges, i.e., edges joining two vertices of the same community, is larger than the number of outer edges, i.e., edges joining two vertices of different communities. An in-depth survey of the problem of community detection in graphs has recently been given by Fortunato [18]. There are several precise definitions of communities, and corresponding criteria, and many more heuristics as well as a few exact algorithms to find partitions or sets of nested partitions into communities. A heuristic finds a near optimal partition (or sometimes an optimal partition but without proof of its optimality) in moderate time. An exact algorithm finds an optimal partition, with proof of its optimality, hopefully in reasonable time. The most used definition for the quality of a community or of a partition into communities is that of *modularity*, proposed by Newman and Girvan [39]. Modularity of a community is defined as the difference between the number of edges it contains and the expected number of edges that it would contain if all edges were drawn at random, keeping the same distribution of degrees. The modularity of a partition is the sum of the modularities of its communities. See e.g. [18, 19, 9] for a discussion of the strengths and weaknesses of the modularity function. Given a network and a partition, modularity can be viewed as a measure of the extent to which the classes of the partition can be considered to be communities. Alternatively, given a network, modularity can be maximized to find an optimal partition, together with its number of clusters and their modularities.

As traditional in Data Analysis, given a set of n entities, clustering heuristics are either hierarchical, i.e., they aim at finding a set of nested partitions, or partitioning schemes, i.e., they aim at finding a single partition (or possibly several partitions into given numbers of clusters). In turn, hierarchical heuristics are divided into agglomerative and divisive ones. Hierarchical agglomerative heuristics [42, 11, 12, 53, 5] proceed from an initial partition with n communities each containing a single entity and iteratively merge the pair of entities for which this operation increases most the objective function (e.g., modularity), until all entities belong to the same community. Thus, they find $2n-1$ communities which are pairwise disjoint or included one into the other. Hierarchical divisive heuristics [38] proceed from an initial partition containing all entities and iteratively divide a community into two in such a way that the increase in the objective function value (e.g. modularity) is the largest possible, or the decrease in the objective value is the smallest possible. Bipartitions are iterated until a partition into n communities having each a single entity is obtained. Thus, once more, $2n-1$ communities are obtained, which are pairwise disjoint or included one into the other. Note that for some objectives, including modularity, mergings or bipartitions can be ended once they do not improve the objective function value any more.

The partitioning and hybrid heuristics rely upon simulated annealing [24, 34, 35], mean field annealing [32], genetic search [51], extremal optimization [16], linear programming followed by randomized rounding [1], dynamical clustering [6], multilevel partitioning [15], contraction-dilation [36], multistep greedy search [49], quantum mechanics [44] and many more sources of inspiration [10, 50, 48, 17, 31].

Hierarchical heuristics are in principle devised for finding a hierarchy of partitions implicit in the given network when it corresponds to some situation where hierarchy is observed or postulated. Such situations include the description of hierarchies in social organization and networks describing evolutionary processes. Results are presented on a dendrogram which displays visually mergings or divisions of communities together

with the values of a characteristic of each community (a variant of the dendrogram, called espalier, allows displaying simultaneously two characteristics of the communities [26]).

The subproblem of choosing at each iteration which pair of communities should be merged is easy. It suffices to consider all $O(n^2)$ merging of pairs of entities and compute each time the objective function value for the new community. Moreover, a careful use of data structures often reduces complexity. Low order polynomial hierarchical agglomerative heuristics have been obtained in several classical papers. These include $O(n^2)$ algorithms for single-linkage [21], complete linkage and minimum variance [4] and several others have a complexity of $O(n^2 \log n)$ [37]. However, in a divisive hierarchical heuristic, the subproblem of finding a bipartition locally optimizing the adopted criterion is more difficult. For some criteria there exists a polynomial algorithm for bipartitioning. For instance, this is the case for the minimum diameter criterion for which there is a $O(n^2)$ algorithm. With careful use of data structures, this gives a $O(n^2 \log n)$ locally optimal algorithm for hierarchical divisive clustering with the minimum diameter criterion [23]. The situation is less favorable for the maximum modularity criterion. Indeed, this problem is NP-hard even in the case of two clusters [8]. Nevertheless, as shown below, a non-polynomial algorithm can solve instances with up to 4941 vertices.

We consider only networks with unweighted and undirected edges in the present paper. Its purpose is to propose a locally optimal divisive heuristic for the maximum modularity criterion. To that effect, in the next section the bipartition subproblem is expressed as a quadratic mixed-integer program with a convex relaxation. This problem can then be solved by the `CPLEX` solver [27]. In Sect. 3 the full hierarchical divisive algorithm is described as well as the previous spectral hierarchical divisive heuristic of Newman [38] and the fast hierarchical agglomerative heuristic of Clauset, Newman and Moore [11]. A computational comparison of the three heuristics, detailing also the respective contributions of Newman's spectral results and the Kernighan-Lin heuristic, is given in Sect. 4. Conclusions are drawn in Sect. 5.

2 An exact algorithm for bipartition

We present in this section an exact algorithm for bipartition with maximization of modularity. We model this bipartitioning problem using binary variables to identify to which community each vertex and each edge belongs. In this respect, our model is similar to that of Xu et al. [55]. These authors proposed in 2007 a modularity maximization model to obtain a partition (generally with more than two communities) of a network. This model is expressed as a mixed integer convex quadratic program. Xu et al. were able to solve exactly instances with up to 104 vertices.

Let $G = (V, E)$ be a graph, or network, with set of vertices V of order $n = |V|$ and set of edges E of size $m = |E|$. We next recall two equivalent definitions of modularity Q. In the first one, it is expressed as a sum over communities of their modularities [39]:

$$Q = \sum_s [a_s - e_s],$$

where a_s is the fraction of all edges that lie within community s and e_s is the expected value of the same quantity in a graph in which the vertices have the same degrees but edges are placed at random. In the second one, modularity Q is expressed as a function, for each community, of its number of inner edges and of the sum of degrees of its vertices:

$$Q = \sum_s \left[\frac{m_s}{m} - \left(\frac{d_s}{2m} \right)^2 \right], \tag{1}$$

where m_s denotes the number of edges in community s, i.e., the subgraph $G_s = (V_s, E_s)$ with set of vertices $V_s \subset V$ and set of edges E_s having both vertices in V_s, and d_s denotes the sum of degrees k_i of the vertices of community s. Since we aim to find a bipartition, only two sub-modules of the original community have to be considered, i.e., $s \in \{1, 2\}$. We can express the sum of degrees d_2 of vertices belonging to the second community as a function of the sum of degrees d_1 of vertices belonging to the first one:

$$d_2 = d_t - d_1, \tag{2}$$

where d_t is the sum of degrees in the community to be bipartitioned and it is equal to $2m$ at the outset. We rewrite (1) for $s \in \{1, 2\}$, using (2):

$$\begin{aligned} Q &= \frac{m_1 + m_2}{m} - \frac{d_1^2}{4m^2} - \frac{d_2^2}{4m^2} = \\ &= \frac{m_1 + m_2}{m} - \frac{d_1^2}{4m^2} - \frac{d_t^2 + d_1^2 - 2d_t d_1}{4m^2} = \\ &= \frac{m_1 + m_2}{m} - \frac{d_1^2}{2m^2} - \frac{d_t^2}{4m^2} + \frac{d_t d_1}{2m^2}. \end{aligned} \tag{3}$$

We then introduce binary variables X_{r1}, X_{r2} and Y_{i1} to model the assignment of vertices and edges to the two communities of the bipartition. These variables are defined as follows:

$$X_{rs} = \begin{cases} 1 & \text{if edge } r \text{ belongs to community } s \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

for $r = 1, 2, \ldots m$ and $s = 1, 2$ and

$$Y_{i1} = \begin{cases} 1 & \text{if vertex } i \text{ belongs to community 1} \\ 0 & \text{otherwise, i.e. if vertex } i \text{ belongs to community 2} \end{cases} \tag{5}$$

for $i = 1, 2, \ldots n$. Two sets of variables X_{r1} and X_{r2} are needed as an edge may belong to the first community, or to the second one, or be a bridge between both of them. One set of variables Y_{i1} suffices as any vertex which does not belong to the first community must belong to the second.

Moreover, we impose for consistency that any edge $r = \{v_i, v_j\}$ with end vertices indiced by i and j can only belong to community s if both of its end vertices belong also to that community:

$$\begin{aligned} X_{r1} &\leq Y_{i1} \quad \forall r = \{v_i, v_j\} \in E \\ X_{r1} &\leq Y_{j1} \quad \forall r = \{v_i, v_j\} \in E \end{aligned} \tag{6}$$

and

$$\begin{aligned} X_{r2} &\leq 1 - Y_{i1} \quad \forall r = \{v_i, v_j\} \in E \\ X_{r2} &\leq 1 - Y_{j1} \quad \forall r = \{v_i, v_j\} \in E. \end{aligned} \tag{7}$$

Furthermore, we exploit the following expressions in terms of variables X and Y for the number of edges of each of the two communities and the sum of vertex degrees of the first one:

$$m_s = \sum_r X_{rs} \quad \forall s \in \{1, 2\}, \tag{8}$$

$$d_1 = \sum_{i \in V_1} k_i Y_{i1}. \tag{9}$$

The sum of vertex degrees of the first community only is needed, because of expression (2).

Maximizing modularity (3) subject to constraints (6)-(7) and (8)-(9) gives a quadratic convex mixed-integer program that can be solved by `CPLEX` [27]. Indeed, this model contains a single nonlinear but concave term, i.e., $-d_1^2/2m^2$, in the objective function, which is to be maximized. Hence, its continuous relaxation, obtained by removing the integrality constraints on the variables, is a convex quadratic program and easy to solve.

Note that in Xu et al.'s [55] model a number of other constraints are imposed. For instance, constraints are used to express that community s can be nonempty only if community $s - 1$ is so, and lower and upper bounds on the cardinality of the modules are added as an option. Furthermore, symmetry-breaking constraints avoid the computation of equivalent alternative optimal solutions.

3 Heuristics

In this section, we first recall the hierarchical agglomerative heuristic of Clauset, Newman and Moore [11] (*CNM*), and the hierarchical divisive spectral heuristic of Newman [38] (*divisive spectral*), to which we compare the heuristic of the present paper. We then describe the new heuristic itself.

The *CNM* heuristic fits into the general scheme for hierarchical agglomerative heuristics of cluster analysis [37]. It can therefore be implemented with a complexity of $O(n^2 \log n)$ in worst case. However, a careful exploitation of sparsity of the graph under study reduces its worst-case complexity to $O(m \log n)$ and its complexity in practice to $O(n \log^2 n)$, which is close to linear time. Results in terms of partitions and the dendrograms obtained are the same for both implementations. The *CNM* heuristic proceeds from an initial partition with n clusters each containing a single entity. Then, it merges iteratively the two clusters in the current partition for which the modularity increases the most. The formula

$$\Delta Q_{ij} = \begin{cases} 1/2m - k_i k_j/(2m)^2 & \text{if } v_i, v_j \text{ are connected} \\ 0 & \text{otherwise} \end{cases}$$

is used initially, then the ΔQ_{ij} for all edges are updated each in constant time. Mergings take place as long as the best of them increases modularity (or, in other words, there is a positive ΔQ_{ij}).

Two remarks are in order. First, in hierarchical agglomerative heuristics, errors, i.e., assignment of two entities to the same cluster while they should be in different clusters in the (or all) optimal partition(s), are never corrected. Second, as n is usually much larger than the number of clusters in the optimal partition, there are close to n mergings before reaching the best partition obtained by the heuristic and hence many occasions of error.

The hierarchical divisive heuristic of Newman proceeds from an initial partition containing all entities by iteratively splitting one of its clusters, as long as this operation increments modulatity. Two questions have to be answered to specify this heuristic: which cluster should be selected for splitting at each iteration and how should the splitting be made. The answer to the former question is unimportant as the best partition obtained does not depend on the order of the splittings but only on the way they are performed. The second question is difficult: indeed, finding the optimal, i.e., modularity maximizing, splitting of a cluster contains the problem of maximum modularity bipartition, which, as shown by Brandes et al. [8], is NP-hard.

For large instances, splitting will have to be done in a heuristic way. Newman [38] proposes two ways to do so. The first is based on spectral graph theory. The first eigenvector of the modularity matrix $B = (b_{ij})$ with

$$b_{ij} = a_{ij} - k_i k_j/2m$$

is computed. The entities corresponding to positive components of this eigenvector form one community and the remaining ones the other.

Results of splitting according to the first eigenvector can be improved by a variety of heuristics. Newman suggests the use of the Kernighan-Lin heuristic [28]. This heuristic proceeds from an initial bipartition to a sequence of re-assignments of one entity from a community to the other. At each step the re-assignment which improves most, or deteriorates least, the objective function value, is selected, performed and further re-assignments of the moved entity forbidden. Once no more re-assignments are allowed, the best partition found among the n partitions considered in the sequence is selected as new initial partition. The whole procedure stops when a full sequence of n re-assignments does not lead to any improvement.

Again two remarks are in order. First, errors, i.e., assigning two entities to different communities when both belong to the same community in the optimal partition are never corrected. Second, as the number of communities in the optimal partition tends to be small, few splittings will take place and thus there are few occasions for errors to be made.

The heuristic of the present paper proceeds along the same lines as the hierarchical divisive heuristic of Newman. The difference is that the exact bipartitioning method of Sect. 2 is used for the splitting step. Due to the difficulty of this subproblem and present limitations of nonlinear integer programming, the exact

bipartitioning could be applied only to small or medium-size networks with up to 4941 vertices. The proposed heuristic is locally optimal in that the splitting step is done optimally, but not globally optimal in that the new heuristic is a greedy one, i.e., each splitting step is done without considering the consequences on further steps, but better results might be obtained if this was done. Note that there are some cases where a greedy heuristic is optimal, the best known one in the field of cluster analysis being the single-linkage algorithm which maximizes the *split* (or *minimum dissimilarity* between pairs of entities in different communities) of the partitions obtained at all levels [21, 14]. This is not the case for the proposed hierarchical divisive heuristic, as shown by an example in Sect. 4.

4 Computational results

In all experiments, we considered a set of 12 well known problems from the literature. They are listed together with their order n and size m in the first three columns of Table 1. They are all undirected and unweighted networks without loops. All data are available on sites listed in [45]. Optimal solutions of these test problems are given in [2]. The bipartition subproblem is solved using `CPLEX` [27]. Other solvers for convex mixed-integer quadratic programs did not perform as well in preliminary tests.

In a first series of experiments, we compared the maximum modularity values and the corresponding number of clusters for the hierarchical agglomerative heuristic of Clauset et al. [11], and for the locally optimal divisive heuristic of the present paper. We also list maximum modularity and number of clusters obtained by an exact algorithm of [2], when possible, i.e., for datasets from 1 to 11. The modularity obtained by the proposed divisive heuristic for the 12th dataset is 0.9394. This appears to be the best value currently known for this dataset. Indeed, using the best heuristic from the extensive comparison by Noack and Rotta [45], gives a value of 0.93854.

In the following, we refer to the locally optimal divisive heuristic of the present paper as *divisive* CHL.

It appears that:

- the locally optimal hierarchical divisive heuristic CHL always gives partitions with smaller modularity than the exact algorithm. The difference, however, is moderate and goes from 0.18018% to 2.36603% of the optimal value. The average error is 0.82540%.
- The agglomerative heuristic always gives partitions with smaller modularity than the divisive one and, by transitivity, than the exact one. The difference this time is much more substantial and goes from 1.21494% to 12.9848%, relative to the exact solution. The average error is 5.52342%. The error between the divisive and the agglomerative heuristics goes from 0.780214% to 10.9013% of the value obtained by the divisive heuristic. The average error is 4.75179%, which is again substantial.
- The divisive heuristic CHL gives a partition with fewer communities than the exact one in 3 cases out of 11, the same number in 3 cases and a larger number in the remaining 5 cases. The average number of communities obtained with the divisive heuristic is 9.27273 vs. 8.81818 for the exact method, so it is slightly in excess.
- The agglomerative heuristic gives a partition with fewer communities than the exact one in 7 cases out of 11, a partition with the same number of communities than the exact one in 2 cases out of 11, and a partition with more communities in the remaining 2 cases. The average number of communities obtained with the agglomerative heuristic is 8 vs. 8.81818 for the exact method, so somewhat smaller.

In a second series of experiments we compared the divisive heuristic CHL of the present paper with the previous divisive heuristic of Newman [38]. To better understand the performance of Newman's heuristic, we consider three versions of it. In the first one, the bipartition at each iteration is done solely on the basis of the leading eigenvector of the modularity matrix. In the second one, the Kernighan-Lin heuristic is applied at each iteration to a randomly generated solution. In the third one, the Kernighan-Lin heuristic is applied at each iteration to the solution given by the leading eigenvector. Results are presented in Tables 2, 3, 4. It appears that

Table 1: Comparison of results of Clauset et al.'s heuristic (CNM), the proposed locally optimal divisive heuristic (*divisive* CHL), and an exact algorithm for modularity maximization [2] (*exact*) on real world datasets. M denotes the number of communities and Q the modularity value of the best solution found. *error*(%) denotes the percentage error for the two heuristics with respect to the exact algorithm. Average values are given for the first 11 datasets as the 12-th one cannot be solved exactly in reasonable time. n and m are the number of vertices and the number of edges of the networks. We consider Zachary's karate club dataset [56] describing friendship relationships between members of a club, Lusseau's dolphins dataset [33] describing communications between dolphins in Doubtful Sound New Zealand, Hugo's *Les Misérables* dataset describing characters in Victor Hugo's masterpiece and their interactions, compiled by Knuth [29], a dataset (A00_main) on classes and relationships from a software project related to Graph Drawing [22], a network dealing with protein interactions [13], Krebs' political books dataset [30], a dataset representing the schedule of football games between American college teams [20], another dataset on classes and relationships from a software project [22], a network dealing with connections between US airports [46], a dataset on a coauthorship network of scientists working on network theory and experiment, compiled by M. Newman [40], a network describing electronic circuits [52] and a network representing the topology of the Western States Power Grid of the United States [54].

dataset	n	m	*agglomerative* CNM			*divisive* CHL			*exact*	
			M	Q	*error*(%)	M	Q	*error*(%)	M	Q
karate	34	78	3	0.38067	9.31895	4	0.41880	0.23583	4	0.41979
dolphin	62	159	4	0.49549	6.24953	4	0.52646	0.38977	5	0.52852
les miserables	77	254	5	0.50060	10.6087	8	0.54676	2.36603	6	0.56001
A00_main	83	135	7	0.52394	1.31098	7	0.52806	0.53494	9	0.53090
p53 protein	104	226	8	0.52052	2.73018	7	0.52843	1.25203	7	0.53513
political books	105	441	4	0.50197	4.79288	4	0.52629	0.18018	5	0.52724
football	115	613	7	0.57728	4.51395	10	0.60091	0.60539	10	0.60457
A01_main	249	635	12	0.59908	5.34366	15	0.62877	0.65255	14	0.63290
usair97	332	2126	7	0.32039	12.9848	8	0.35959	2.33840	6	0.36820
netscience_main	379	914	19	0.83829	1.21494	20	0.84702	0.18619	19	0.84860
s838	512	819	12	0.80556	1.68904	15	0.81663	0.33805	12	0.81940
power	4941	6594	39	0.93402	–	40	0.93937	–	-	–
average			8	0.55125	5.52342	9.3	0.57525	0.82540	8.8	0.57957

- the first version always gives partitions with substantially smaller modularities than the divisive heuristic of the present paper or the exact algorithm. The error relative to the partition given by the exact algorithm goes from 5.81428% to 18.5189%. The average error is 10.7419%. The error relative to the partition given by the divisive heuristic of the present paper goes from 5.63859% to 18.0227%. The average error is 9.99712%.
- The second version gives results even worse than those of the first version. The error relative to the partition given by the exact algorithm goes from 5.95934% to 28.6442%. The average error is 14.5856%.
- The third version gives better results The error relative to the partition given by the exact algorithm goes from 0.09863% to 6.07995%. The average error is 3.02627%. The error relative to the partition given by the divisive heuristic of the present paper goes from −0.081704% to 5.46305%. The average error is 2.21592%. Observe that in the case of the `political books` instance, the modularity obtained with the *divisive spectral + KL* heuristic, i.e., 0.52672, is slightly better than that one obtained with the heuristic of the present paper, i.e, 0.52629. This is not a numerical error but illustrates, as mentioned above, that the proposed heuristic is locally but not globally optimal. The observation that version 3 is better than version 1 and that version 1 is better than version 2 was already made by Newman [38].
- Computing times of the heuristic proposed in this paper and an exact column generation algorithm for modularity maximization of [2] are given in Table 5. As the computers used are not the same for the two cases, these results should only be considered as indicative. Ratios between these computing times are given in the penultimate column and similar ratios, corrected by dividing those times by the clock frequency of the computer used, in the last column. The time of the heuristic CHL is less than the time of the exact algorithm in 9 cases over 12 and in several cases very substancially so. Moreover, the heuristic could solve the last problem, which is about 10 times larger than the penultimate, in reasonable time, while the exact algorithm could not.

- Currently, it is not possible to solve very large instances with the heuristic of the present paper or large ones with the exact column generation algorithm for modularity maximization of [2]. One cannot therefore be sure that the differences in performances between the heuristics described above and/or the exact algorithm extend to the resolution of large instances, except in a few cases. A comparison between the agglomerative heuristic CNM and the complete version of the divisive heuristic of Newman is part of Table 1 of [38]. The corresponding columns are reproduced in Table 6, together with the percentage errors in the modularities obtained. This confirms that the agglomerative heuristic of Clauset et al. gives poor results for small as well as for large instances.

Table 2: Comparison of results of Newman's spectral divisive heuristic (*divisive spectral*), the proposed locally optimal divisive heuristic (*divisive* CHL), and an exact algorithm for modularity maximization [2] (*exact*) on real world datasets. M denotes the number of communities and Q the modularity value of the best found solution. *error*(%) denotes the percentage error for the two heuristic with respect to the exact algorithm. Average values are given for the first 11 datasets as the 12-th one cannot be solved exactly in reasonable time. n and m are the number of vertices and the number of edges of the networks.

dataset	n	m	*divisive spectral*			*divisive* CHL			*exact*	
			M	Q	*error*(%)	M	Q	*error*(%)	M	Q
karate	34	78	4	0.39341	6.28409	4	0.41880	0.23583	4	0.41979
dolphin	62	159	5	0.49120	7.06123	4	0.52646	0.38977	5	0.52852
les miserables	77	254	9	0.51383	8.24628	8	0.54676	2.36603	6	0.56001
A00_main	83	135	4	0.46082	13.2002	7	0.52806	0.53494	9	0.53090
p53 protein	104	226	8	0.49152	8.14942	7	0.52843	1.25203	7	0.53513
political books	105	441	5	0.46718	11.3914	4	0.52629	0.18018	5	0.52724
football	115	613	8	0.49261	18.5189	10	0.60091	0.60539	10	0.60457
A01_main	249	635	8	0.53755	15.0656	15	0.62877	0.65255	14	0.63290
usair97	332	2126	8	0.31666	13.9978	8	0.35959	2.33840	6	0.36820
netscience_main	379	914	15	0.79926	5.81428	20	0.84702	0.18619	19	0.84860
s838	512	819	18	0.73392	10.432	15	0.81663	0.33805	12	0.81940
power	4941	6594	11	0.53516	–	40	0.93937	–	-	–
average			8.36	0.51710	10.7419	9.3	0.57525	0.82540	8.8	0.57957

Table 3: Comparison of results of the divisive Kernighan-Lin based heuristic (KL), the proposed locally optimal divisive heuristic (*divisive* CHL), and an exact algorithm for modularity maximization [2] (*exact*) on real world datasets. M denotes the number of communities and Q the modularity value of the best found solution. *error*(%) denotes the percentage error for the two heuristics with respect to the exact algorithm. Average values are given for the first 11 datasets as the 12-th one cannot be solved exactly in reasonable time. n and m are the number of vertices and the number of edges of the networks. Note that values of Q and error percentage are given with less digits that elsewhere in this paper due to the fact that the Kernighan-Lin heuristic depends upon the random partition to which it is applied.

dataset	n	m	KL			*divisive* CHL			*exact*	
			M	Q	*error*(%)	M	Q	*error*(%)	M	Q
karate	34	78	2	0.372	11.43	4	0.41880	0.23583	4	0.41979
dolphin	62	159	4	0.477	9.745	4	0.52646	0.38977	5	0.52852
les miserables	77	254	3	0.489	12.73	8	0.54676	2.36603	6	0.56001
A00_main	83	135	3	0.450	15.31	7	0.52806	0.53494	9	0.53090
p53 protein	104	226	11	0.402	24.83	7	0.52843	1.25203	7	0.53513
political books	105	441	4	0.496	5.959	4	0.52629	0.18018	5	0.52724
football	115	613	5	0.538	11.048	10	0.60091	0.60539	10	0.60457
A01_main	249	635	6	0.512	19.023	15	0.62877	0.65255	14	0.63290
usair97	332	2126	6	0.339	7.992	8	0.35959	2.33840	6	0.36820
netscience_main	379	914	8	0.606	28.644	20	0.84702	0.18619	19	0.84860
s838	512	819	5	0.707	13.726	15	0.81663	0.33805	12	0.81940
power	4941	6594	8	0.649	–	40	0.93937	–	-	–
average			5.2	0.48970	14.5856	9.3	0.57525	0.82540	8.8	0.57957

Table 4: Comparison of results of Newman's spectral divisive heuristic with the Kernighan-Lin refinement (*divisive spectral + KL*), the proposed locally optimal divisive heuristic (*divisive* CHL), and an exact algorithm for modularity maximization [2] (*exact*) on real world datasets. M denotes the number of communities and Q the modularity value of the best found solution. *error*(%) denotes the percentage error for the two heuristics *divisive spectral + KL* and *divisive* with respect to the exact algorithm. *error_div*(%) denotes the percentage error for the *divisive spectral + KL* with respect to the proposed locally optimal heuristic. Average values are given for the first 11 datasets as the 12-th one cannot be solved exactly in reasonable time. n and m are the number of vertices and the number of edges of the networks. Note that values of Q and error percentage are given with less digits that elsewhere in this paper due to the fact that the Kernighan-Lin heuristic depends upon the random partition to which it is applied.

dataset	n	m	*divisive spectral + KL*				*divisive* CHL			*exact*	
			M	Q	*error_div*(%)	*error*(%)	M	Q	*error*(%)	M	Q
karate	34	78	4	0.419	0	0.236	4	0.41880	0.23583	4	0.41979
dolphin	62	159	5	0.508	3.415	3.792	4	0.52646	0.38977	5	0.52852
les miserables	77	254	7	0.538	1.533	3.862	8	0.54676	2.36603	6	0.56001
A00_main	83	135	7	0.527	0.199	0.733	7	0.52806	0.53494	9	0.53090
p53 protein	104	226	6	0.518	1.930	3.158	7	0.52843	1.25203	7	0.53513
political books	105	441	4	0.527	-0.081	0.099	4	0.52629	0.18018	5	0.52724
football	115	613	8	0.579	3.638	4.221	10	0.60091	0.60539	10	0.60457
A01_main	249	635	16	0.594	5.463	6.080	15	0.62877	0.65255	14	0.63290
usair97	332	2126	7	0.358	0.501	2.827	8	0.35959	2.33840	6	0.36820
netscience_main	379	914	23	0.820	3.191	3.371	20	0.84702	0.18619	19	0.84860
s838	512	819	13	0.779	4.587	4.910	15	0.81663	0.33805	12	0.81940
power	4941	6594	8	0.791	–		40	0.93937	–	-	–
average			9.09	0.560731	2.21592	3.02627	9.3	0.57525	0.82540	8.8	0.57957

Table 5: Comparison of times for the proposed locally optimal heuristic and an exact algorithm for modularity maximization. All results are in seconds of CPU. Solutions given by the proposed heuristic were obtained on a 2.4 GHz Intel Xeon CPU of a computer with 8GB RAM shared by three other similar CPU running Linux. Solutions given by the exact algorithm were obtained on a dual processor computer Intel Pentium with 3.20 GHz, 3GB RAM running Linux. In the penultimate column, ratios between computing times of the proposed heuristic and the exact algorithm are reported. In the last column, ratios between these computing times are corrected by dividing CPU times by the clock frequency of the computer used.

dataset	n	m	*time divisive* CHL	*time exact*	CHL/*exact*	(CHL/*exact*)'
karate	34	78	0.42	0.34	1.2353	1.6471
dolphin	62	159	1.40	7.75	0.1806	0.2409
les miserables	77	254	4.52	7.26	0.6226	0.8301
A00_main	83	135	0.89	3.66	0.2432	0.3242
p53 protein	104	226	7.81	11.60	0.6733	0.8977
political books	105	441	12.09	45.65	0.2648	0.3531
football	115	613	338.19	249.41	1.3559	1.8979
A01_main	249	635	656.54	1014.48	0.6472	0.8629
usair97	332	2126	33157.85	16216.77	2.0447	2.7262
netscience_main	379	914	22.84	1615.14	0.0141	0.0188
s838	512	819	38.37	7655.56	0.0050	0.0067
power	4941	6594	4498.21	–	–	–

5 Conclusions

In this paper we presented a hierarchical divisive heuristic for modularity maximization which is locally optimal, i.e., such that the bipartition obtained at each iteration is guaranteed to be optimal. This heuristic can be used for two purposes: approximate modularity maximization of general networks or modularity maximization of networks which are known to correspond to some hierarchy either natural or man made. In the former case, experimental results show that the partitions obtained with the proposed divisive heuristic tend to have a modularity value close to that of optimal partitions (recall that the average error observed in our experiments is 0.82540%). So, the partition found can be considered as a fairly good approximation

Table 6: Comparison of results [38] of the Clauset et al.'s agglomerative heuristic (*agglomerative CNM*) and Newman's spectral divisive heuristic with the Kernighan-Lin refinement (*divisive spectral + KL*). Q denotes the modularity value of the best found solution. *error*(%) denotes the percentage error for the first heuristic with respect to the second one. n is the number of vertices. *Jazz* dataset describes musicians which worked together [47], *metabolic* dataset describes chemical reactions as well as the regulatory interactions that guide these reactions in C. elegans [3], *e-mail* dataset describes e-mail interchanges between members of a university [25], *key signing* (PGP) datadset describes the giant component of the network of users of the Pretty-Good-Privacy algorithm for secure information interchange [7], `physicists` (cond-mat) dataset describes a collaboration network of scientists posting preprints on the condensed matter archive [43].

dataset	n	Q *agglomerative CNM*	Q *divisive spectral + KL*	*error*(%)
`jazz`	198	0.439	0.442	0.67873
`metabolic`	453	0.402	0.435	7.58621
`e-mail`	1133	0.494	0.572	13.6364
`key signing`	10680	0.733	0.855	14.269
`physicists`	27519	0.668	0.723	7.6072
average		0.5195	0.5743	8.8078

of the optimal one, or at least as a tentative solution to be improved upon by various local improvement heuristics.

Our experiments also show that modularity maximization with the hierarchical agglomerative heuristic of Clauset, Newman and Moore tends to have a much larger error (with an average error of 5.52342% for our experiments).

It is for the latter case that our heuristic is tailored. We therefore compared it with three versions of the divisive heuristic of Newman. As observed by that author, a two-phase method in which a first bipartition is made in a splitting step on the basis of the first eigenvector of the modularity matrix followed by application of the Kernighan-Lin heuristic gives the best results. Still, the proposed heuristic appears to be even better, indeed the average error relative to the exact solution is reduced more than threefold, i.e., from 3.02627% to 0.82540%.

References

[1] G. Agarwal and D. Kempe. Modularity-maximizing graph communities via mathematical programming. *The European Physical Journal B*, 66(3):409–418, 2008.

[2] D. Aloise, S. Cafieri, G. Caporossi, P. Hansen, L. Liberti, and S. Perron. Column generation algorithms for exact modularity maximization in networks. *Physical Review E*, 82(4):046112, 2010.

[3] `http://deim.urv.cat/ aarenas/data/welcome.htm`.

[4] J. P. Benzécri. Construction d'une classification ascendante hierarchique par la recherche en chaîne des voisins reciproques. *Les Cahiers de l'Analyse des Données*, VII:209–219, 1982.

[5] V.D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal Statistical Mechanics: Theory and Experiment*, page P10008, 2008.

[6] S. Boccaletti, M. Ivanchenko, V. Latora, A. Pluchino, and A. Rapisarda. Detecting complex network modularity by dynamical clustering. *Physical Review E*, 75:045102, 2007.

[7] M. Boguna, R. Pastor-Satorras, A. Diaz-Guilera, and A. Arenas. *Physical Review E*, 70:056122, 2004.

[8] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner. On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20(2):172–188, 2008.

[9] S. Cafieri, P. Hansen, and L. Liberti. Loops and multiple edges in modularity maximization of networks. *Physical Review E*, 81(4):046102, 2010.

[10] D. Chen, Y. Fu, and M. Shang. A fast and efficient heuristic algorithm for detecting community structures in complex networks. *Physica A*, 388(13):2741–2749, 2009.

[11] A. Clauset, M.E.J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70:066111, 2004.

[12] Leon Danon, Albert Diaz-Guilera, and Alex Arenas. The effect of size heterogeneity on community identification in complex networks. *Journal of Statistical Mechanics: Theory and Experiment*, page P11010, 2006.

[13] L. Dartnell, E. Simeonidis, M. Hubank, S. Tsoka, I.D.L. Bogle, and L.G. Papageorgiou. Self-similar community structure in a network of human interactions. *FEBS Letters*, 579:3037–3042, 2005.

[14] M. Delattre and P. Hansen. Bicriterion cluster analysis. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 2(4):277–291, 1980.

[15] H.N. Djidjev. A scalable multilevel algorithm for graph clustering and community structure detection. *Lecture Notes in Computer Science*, 4936:117–128, 2008.

[16] J. Duch and A. Arenas. Community identification using extremal optimization. *Physical Review E*, 72(2):027104, 2005.

[17] Y. Fan, M. Li, P. Zhang, J. Wu, and Z. Di. Accuracy and precision of methods for community identification in weighted networks. *Physica A*, 377(1):363–372, 2007.

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

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

[20] M. Girvan and M. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences, USA*, 99(12):7821–7826, 2002.

[21] J.C. Gower and G.J.S. Ross. Minimum spanning trees and single linkage cluster analysis. *Applied Statistics*, 18:54–64, 1969.

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

[23] A. Guénoche, P. Hansen, and B. Jaumard. Efficient algorithms for divisive hierarchical clustering with the diameter criterion. *Journal of Classification*, 8:5–30, 1991.

[24] R. Guimerà and A.N. Amaral. Functional cartography of complex metabolic networks. *Nature*, 433:895–900, 2005.

[25] R. Guimerà, L. Danon, A. Diaz-Guilera, F. Giralt, and A. Arenas. Self-similar community structure in a network of human interactions. *Physical Review E*, 68:065103, 2003.

[26] P. Hansen, B. Jaumard, and B. Simeone. Espaliers: A generalization of dendrograms. *Journal of Classification*, 13:107–127, 1996.

[27] ILOG. *ILOG CPLEX 11.0 User's Manual*. ILOG S.A., Gentilly, France, 2008.

[28] B.W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, pages 291–307, 1970.

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

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

[31] J.M. Kumpula, J. Saramaki, K. Kaski, and J. Kertesz. Limited resolution and multiresolution methods in complex network community detection. *Fluctuation and Noise Letters*, 7(3):L209–L214, 2007.

[32] S. Lehmann and L.K. Hansen. Deterministic modularity optimization. *European Physical Journal B*, 60:83–88, 2007.

[33] D. Lusseau, K. Schneider, O.J. Boisseau, P. Haase, E. Slooten, and S.M. Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. can geographic isolation explain this unique trait? *Behavioral Ecology and Sociobiology*, 54(4):396–405, 2003.

[34] C.P. Massen and J.P.K. Doye. Identifying communities within energy landscapes. *Physical Review E*, 71:046101, 2005.

[35] A. Medus, G. Acuna, and C.O. Dorso. Detection of community structures in networks via global optimization. *Physica A*, 358:593–604, 2005.

[36] J. Mei, S. He, G. Shi, Z. Wang, and W. Li. Revealing network communities through modularity maximization by a contraction-dilation method. *New Journal of Physics*, 11:043025, 2009.

[37] F. Murtagh. A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal*, 26:354–359, 1983.

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

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

[40] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74:036104, 2006.

[41] M. E. J. Newman. *Networks: an introduction*. Oxford University Press, Oxford, 2010.

[42] M.E.J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69:066133, 2004.

[43] `http://www-personal.umich.edu/ mejn/netdata/`.

[44] Y.Q. Niu, B.Q. Hu, W. Zhang, and M. Wang. Detecting the community structure in complex networks based on quantum mechanics. *Physica A*, 387(24):6215–6224, 2008.

[45] A. Noack and R. Rotta. Multi-level algorithms for modularity clustering. *Lecture Notes in Computer Science*, 5526:257–268, 2009.

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

[47] P.Gleiser and L. Danon. *Adv. Complex Syst*, 6:565, 2003.

[48] J. Ruan and W. Zhang. Identifying network communities with a high resolution. *Physical Review E*, 77:016104, 2008.

[49] P. Schuetz and A. Caflisch. Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement. *Physical Review E*, 77:046112, 2008.

[50] Y. Sun, B. Danila, K. Josic, and K. E. Bassler. Improved community structure detection using a modified fine-tuning strategy. *Europhysics Letters*, 86:28004, 2009.

[51] M. Tasgin, A. Herdagdelen, and H. Bingol. Community detection in complex networks using genetic algorithms. *arXiv:0711.0491*, 2007.

[52] `http://www.weizmann.ac.il/mcb/UriAlon/`.

[53] Ken Wakita and Toshiyuki Tsurumi. Finding community structure in mega-scale social networks. Technical Report 0702048v1, arXiv, 2007.

[54] D.S. Watts and S.H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):409–410, 1998.

[55] G. Xu, S. Tsoka, and L.G. Papageorgiou. Finding community structures in complex networks using mixed integer optimization. *European Physical Journal B*, 60:231–239, 2007.

[56] W.W. Zachary. An information flow model for conflict and fission in small group. *Journal of Anthropological Research*, 33:452–473, 1977.