Discrete Applied Mathematics 163 (2014) 65–72

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Improving heuristics for network modularity maximization using an exact algorithm

Sonia Cafieri [a,*], Pierre Hansen [b,c], Leo Liberti [c]

[a] Laboratoire MAIAA, École Nationale de l'Aviation Civile, 7 Av. E. Belin, 31055 Toulouse, France
[b] GERAD and HEC Montréal, 3000 Chemin de la Côte-Sainte-Catherine, Montréal, Canada, H3T 2A7
[c] LIX, École Polytechnique, 91128 Palaiseau, France

ARTICLE INFO

Article history:
Received 15 February 2011
Received in revised form 30 January 2012
Accepted 24 March 2012
Available online 25 April 2012

Keywords:
Clustering
Bipartition
Network
Graph
Community
Modularity
Heuristic
Exact algorithm
Matheuristic

ABSTRACT

Heuristics are widely applied to modularity maximization models for the identification of communities in complex networks. We present an approach to be applied as a post-processing to heuristic methods in order to improve their performances. Starting from a given partition, we test with an exact algorithm for bipartitioning if it is worthwhile to split some communities or to merge two of them. A combination of merge and split actions is also performed. Computational experiments show that the proposed approach is effective in improving heuristic results.

1. Introduction

The identification of communities in complex networks has become in recent years a very active research domain [37,18] because of the common representation of complex real-world systems arising in a variety of fields as networks. One then aims to find *communities*, or *clusters*, of entities grouped on the basis of some relationship holding among them. Telecommunication networks such as the World Wide Web, biological networks representing interactions between proteins and social networks representing collaborations or conflicts between people or countries are some examples of real-life applications (see [37] for a detailed introduction).

Intuitively, one would say that a set of vertices of a network form a community if edges joining two vertices of that set are frequent and edges joining a vertex of that set to a vertex outside are not. This concept has been refined in many ways, leading to the introduction of concepts of *modularity* [40], *modularity density* [32], *min-max cut* [14], *normalized cut* [48] and others. Among these concepts, modularity is by far the most popular.

Modularity of a community is defined in [40] as the difference between the fraction of edges it contains and the expected fraction of edges it would contain if they were placed at random, keeping the same degree distribution. Then, modularity of a partition of a network into communities is defined as the sum of the modularities of these communities. Modularity

* Corresponding author. Fax: +33 562174507.
E-mail addresses: sonia.cafieri@enac.fr (S. Cafieri), pierre.hansen@gerad.ca (P. Hansen), liberti@lix.polytechnique.fr (L. Liberti).

0166-218X/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2012.03.030

expresses not only that a community contains a large fraction of the edges, but also that it contains a larger fraction of the edges than would be expected. It can be viewed as a measure of the extent to which the classes of a partition of a graph can be considered to be communities. Alternatively, modularity can be maximized to find an optimal partition of a network.

Modularity maximization has spawned in recent years numerous methods for cluster identification in networks. Despite its popularity, the accuracy and the significance of modularity maximizing modules are not well understood for real-world networks [21]. Furthermore, some criticism have been raised in recent literature, see, e.g., [21,19,7,34,30]. The two main concerns are the existence of a resolution limit and the fact that modularity function exhibits a degeneracy. The resolution limit, identified by Fortunato and Barthelemy [19], implies that, in the presence of large clusters, some clusters smaller than a certain size which depends on the number of edges of the network can be undetectable. As a consequence, modular structures like small cliques can be hidden in larger clusters. This effect appears to be driven primarily by the assumption that inter-module connectivity follows a random graph model [21]. Degeneracy (see [21]) implies that there can be a large number of partitions, even very different from each other, having high modularity values. This makes it easy to find high-scoring partitions but difficult to identify the global optimum. To address these criticisms a few approaches have been proposed. Sales-Pardo et al. [46] address the problem of degeneracy combining information from many distinct partitions with high modularity. Multiresolution methods [28,3,44] allow us to specify a target resolution limit and identify clusters on such given scale, though they do not solve the problem in a fully satisfactory manner. Despite these criticisms, modularity maximization still appears to be a very popular technique for network clustering. It exhibits, in fact, some clear advantages: modularity function has a clear and simple mathematical description and does not depend on parameters being decided arbitrarily (as an example, maximizing the number of intracluster edges requires some other parameter, e.g. the minimum cluster size); modularity maximization gives an optimal partition together with the number of clusters not to be specified in advance. Interestingly, one can use mathematical programming to model the community detection problem and, using modularity maximization, the splitting of a cluster into two can be expressed as a quadratic programming problem. This paper discusses such a formulations and exploits it within a procedure used as a refinement of previously computed partitions.

Numerous heuristics and a few algorithms have been proposed to find near optimal or optimal partitions respectively for the maximum modularity criterion. Heuristics are either partitioning methods or hierarchical divisive or agglomerative ones. Partitioning heuristics are based on simulated annealing [24,34,35], mean field annealing [31], genetic search [50], extremal optimization [16], spectral clustering [38], linear programming followed by randomized rounding [1], dynamical clustering [5], multilevel partitioning [15], contraction–dilation [36], quantum mechanics [41] and several other approaches [4,9,49,45,17,28]. Agglomerative hierarchical clustering [39,10,12,51,4] proceeds from an initial partitions into communities each containing a single vertex to merging sequentially vertices or sets of vertices corresponding to communities. In [47] this approach is combined with a vertex mover routine which improves the partitions by changing the community of a vertex to that of one of its adjacent vertices. Divisive hierarchical clustering proceeds from an initial trivial partition in one community containing all vertices and sequentially selects a community and proceeds to its bipartitioning. Divisive heuristics are much less frequent than agglomerative ones. The best known of them is Newman's spectral heuristic [38], which uses the signs of the first eigenvector of the modularity matrix to perform successive bipartitions. In a companion paper [8], we propose a hierarchical divisive heuristic which is locally optimal, i.e., in which all successive bipartitions are done in an optimal way.

These heuristics are able to solve large instances with up to thousand or tens of thousands of vertices (and sometimes over a million) and therefore are often preferred to exact algorithms, even though they do not have a guarantee of optimality. Only a few papers propose exact algorithms for maximizing modularity. The first one, due to Xu et al. [53], uses quadratic mixed-integer programming with a convex relaxation. Networks with up to 104 vertices were addressed successfully. Brandes et al. [6] have shown that modularity maximization is NP-hard, even if there are only two communities. In addition, they propose to express modularity maximization as a clique partitioning problem. They maximize modularity of networks with up to 105 vertices. Their algorithm is close to that of Grötschel and Wakabayashi [22,23]. Aloise et al. [2] apply column generation to modularity maximization and solve exactly instances with up to 512 vertices.

Given a partition found by a heuristic, one can apply another heuristic or an exact algorithm to the subnetworks induced by the communities found. This will eventually lead to a new, better, partition. Moreover, this refinement can be based on splitting a community or merging a pair of communities. In the spirit of *matheuristics*, an exact algorithm for bipartition is applied in our approach first to the communities considered one at a time, then merging pairs of communities and applying again the bipartition algorithm.

We employ our approach as post-processing of some known heuristics for modularity maximization, obtaining improved solutions and, for some datasets, the optimal partition.

The paper is organized as follows. In the next section, the proposed approach to improve heuristic results for modularity maximization is described, presenting in particular an exact algorithm for bipartition. Section 3 presents the results of computational experiments carried out applying the proposed approach as post-processing to three of the best heuristics available for modularity maximization of networks, i.e., the agglomerative hierarchical heuristic of Clauset et al. [10], the partitioning heuristic of Noack and Rotta [42] and the multistep greedy with vertex move heuristic of Schuetz and Caflisch [47]. We also apply this approach to the locally optimal divisive hierarchical heuristic of [8]. Conclusions are given in Section 4.