Algorithms for network modularity maximization

Daniel Aloise, Sonia Cafieri, Gilles Caporossi, Pierre Hansen, Leo Liberti, Sylvain Perron

Algorithms for network modularity maximization

Daniel Aloise[1], Sonia Cafieri[2], Gilles Caporossi[1], Pierre Hansen[1,3], Leo Liberti[3], Sylvain Perron[1]

[1] GERAD, HEC Montréal, Canada
daniel.aloise,gilles.caporossi,pierre.hansen,sylvain.perron,@gerad.ca
[2] Dept. Mathématiques et Informatique, ENAC, 7 av. E. Belin, 31055 Toulouse, France
sonia.cafieri@enac.fr
[3] LIX, École Polytechnique, 91128 Palaiseau, France
liberti@lix.polytechnique.fr

Mots-Clés : *clustering, network, graph, community, modularity, row and column generation*

1 Introduction

Networks are often used to represent complex systems arising in a variety of fields. Social networks model interactions among people. Telecommunication networks model communications betwen them, such as in the World Wide Web. Transportation networks model movements of goods and passengers. Biological networks model interactions between organisms, such as in food networks. A network (or graph) $G = (V, E)$ is composed of a set of vertices, representing the entities of the system under study, and a set of edges joining pairs of vertices and representing a relation holding for such pairs. Identifying *communities*, or *clusters*, in complex networks is a topic of particular interest and is currently a very active research domain. A clustering criterion is chosen, in terms of presence or absence of edges, as well as a structure for the clusters. The corresponding mathematical programming problem is then solved exactly or, more often, heuristically. Clustering algorithms and heuristics can be divided into partitioning algorithms, which aim at finding the best partition into a (often but not always) given number of clusters, and hierarchical algorithms, which lead to a set of nested partitions and can be further divided into agglomerative and divisive ones. Algorithms for community identification need a precise definition of community. A very successful definition is that proposed by Newman [4], which leads to the introduction of the concept of *modularity* for a partition of a network. It is defined as the sum for all communities of the difference between the fraction of edges they contain and the expected fraction of edges if they are placed at random, keeping the same degree distribution. A few algorithms and many heuristics have been developed to maximize modularity. Heuristics rely upon agglomerative and divisive hierarchical clustering, the latter including a spectral approach, and partitioning schemes based on simulated annealing, genetic search, and a variety of other approaches. These heuristics can solve approximately very large instances with up to hundred or thousand entities or even more. However, they do not have either an a priori performance guarantee (e.g. in the form of finding always a solution with a value which is at least a given percentage of the optimal one), nor an a posteriori performance guarantee (e.g. that the obtained solution is at least a computable percentage of the optimal one). By contrast, algorithms provide an optimal solution together with the proof of its optimality. Very few papers propose exact algorithms for modularity maximization. Moreover, they can only solve small instances (with about a hundred entities) in reasonable time. We believe, by analogy with other problems such as the traveling salesman problem, this performance can, and probably will, be much increased. For that purpose, the present paper reviews and compares the

algorithms yet proposed. It also presents a new stabilised column generation approach which raised the size of instances solved exactly from 115 to 512 entities.

2 Row and column generation algorithms

As shown by Brandes et al. [1], modularity maximization in NP-complete, and can be expressed, when there are n entities, as a clique partitioning problem with $O(n^2)$ 0-1 variables expressing that pairs of entities belong to the same cluster or not, and $O(n^3)$ constraints expressing that if entities i and j belong to the same cluster and entities j and k belong to the same cluster, then entities i and k must also belong to the same cluster. These constraints are numerous and can be added by batches of unsatisfied ones, as already noted by Grötschel and Wakabayashi [3]. Such row-generation approach can be easily implemented with the "lazy constraints" feature of CPLEX.

Alternately, one can solve this clique partitioning problem by column generation. The auxiliary problem is then an unconstrained quadratic 0-1 program with a 100% dense matrix. It can be solved by a Variable Neighborood Search (VNS) heuristic as long as a column with positive reduced cost can be found and with an exact branch and bound algorithm when this is no more the case.

A different approach was recently proposed by Xu, Tsoka and Papageorgiou [5]. These authors introduced indicator variables for entities and for edges of $G = (V, E)$ to belong to the first, second,. . .cluster. When G is sparse and the number of clusters is small, the model has less variables and constraints than the clique partitioning one.

Again, a column generation approach can be applied, replacing the difficult problem of finding simultaneously all clusters by a sequence of problems of finding one improved cluster at a time. Column generation algorithms are known to have slow convergence, therefore stabilization is applied [2] or, more precisely, focussing. For that purpose, an initial heuristic solution is found by a VNS or other heuristic, from which a plausible interval for the optimal values of the dual variables of the master problem are deduced by sensitivity analysis. Then departures from these intervals are penalized, in a decreasing way.

The four exact algorithms described are compared on a series of problems from the literature. Each of them except that one of Xu et al. is best in some case. The new column generation algorithm outperforms the others for the larger instances.

Références

[1] 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.

[2] O. du Merle, D. Villeneuve, J. Desrosiers, and P. Hansen. Stabilized column generation. *Discrete Mathematics*, 194 :229–237, 1999.

[3] M. Grötschel and Y. Wakabayashi. A cutting plane algorithm for a clustering problem. *Mathematical Programming*, 45 :59–96, 1989.

[4] M. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69 -026133, 2004.

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