Eur. Phys. J. B **60**, 231–239 (2007)
DOI: 10.1140/epjb/e2007-00331-0

THE EUROPEAN PHYSICAL JOURNAL B

Finding community structures in complex networks using mixed integer optimisation

G. Xu[1], S. Tsoka[2], and L.G. Papageorgiou[1,a]

[1] Centre for Process Systems Engineering, Department of Chemical Engineering, UCL (University College London), Torrington Place, London WC1E 7JE, UK
[2] Centre for Bioinformatics, School of Physical Sciences and Engineering, King's College London, Strand, London WC2R 2LS, UK

Received 20 February 2007 / Received in final form 21 September 2007
Published online 8 December 2007 –

Abstract. The detection of community structure has been used to reveal the relationships between individual objects and their groupings in networks. This paper presents a mathematical programming approach to identify the optimal community structures in complex networks based on the maximisation of a network modularity metric for partitioning a network into modules. The overall problem is formulated as a mixed integer quadratic programming (MIQP) model, which can then be solved to global optimality using standard optimisation software. The solution procedure is further enhanced by developing special symmetry-breaking constraints to eliminate equivalent solutions. It is shown that additional features such as minimum/maximum module size and balancing among modules can easily be incorporated in the model. The applicability of the proposed optimisation-based approach is demonstrated by four examples. Comparative results with other approaches from the literature show that the proposed methodology has superior performance while global optimum is guaranteed.

PACS. 89.75.Hc Networks and genealogical trees – 02.60.Pn Numerical optimization – 87.23.Ge Dynamics of social systems

1 Introduction

Many complex systems such as the Internet, social and biological relations have been represented as networks consisting of a set of nodes joined in pairs by edges to reflect the number of components in the systems and connections among them. Statistical analysis of networks has revealed a number of properties such as small world effects, degree distribution and high network transitivity (see [1–3] for reviews). In this paper, we develop a mathematical framework for the identification and analysis of community structures in networks.

Community structures are often found in various types of networks where the vertices are naturally clustered into tightly connected modules with large number of within-module edges and few inter-module links. The ability to identify and analyse such structures could be of vital importance in practice. For example, groups within the World Wide Web may reveal the thematic relationships of websites on similar topics [4,5]; modules found in social networks may correspond to different local communities [6,7]; subgroups in metabolic and cellular networks may reflect distinct functions in biological systems and evolutionary properties of biological molecules and species [8,9]. Therefore, the modular view of networks provides a clearer understanding on how complex systems are constructed from a number of fundamental components and sheds light into the interactions of such components.

A number of computational approaches have been proposed by various research groups to detect community structures in networks. Traditional methods comprise graph partitioning [10] and hierarchical clustering [11]. Graph partitioning deals with the separation of a network into several groups with roughly equal sizes so as to minimise the inter-group communications [12,13]. In the area of parallel computing, graph partitioning is applied in order to distribute different tasks to several processors while minimising inter-processor communications. As partitioning a graph is NP-complete [10], most heuristic algorithms proposed were bisection-based where a network was divided into a number of communities by an iterative bisection procedure [14,15]. It should be noted that the optimal solutions to the graph partitioning problem cannot be guaranteed since both the number of communities as well as the sizes of each group are previously fixed by the user.

Hierarchical clustering has also been applied extensively in the investigation of community structures of social and biological systems [9,16–18]. It is an agglomerative procedure transforming a distance matrix of pair-wise

[a] e-mail: l.papageorgiou@ucl.ac.uk

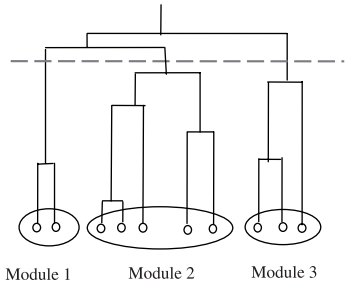

Fig. 1. A dendrogram of 10-node network generated by hierarchical clustering.

similarity measurements between all pairs of nodes into a hierarchical partition tree. Initially, each node forms an independent module and the number of modules is gradually reduced by merging the two most similar clusters iteratively until the whole network is included in one community. Any horizontal cut of the hierarchical tree splits the network into a number of subgroups (see Fig. 1 for a network of 10 nodes and 3 modules). Although hierarchical clustering does not require any specification of the size or number of modules, it cannot reveal which partition is the best one. Another problem associated with hierarchical clustering lies in its tendency to group only tightly connected nodes in the early stage of clustering because of their strong similarities. However, it cannot always classify nodes with few connectivities correctly since end solution depends on where the agglomerative procedure starts.

Apart from traditional methodologies proposed above, a number of local algorithms and physical models were applied to detect community structures. First, a set of self-contained local algorithms to detect network communities were proposed. The algorithms kept the same level of liability and outperformed other existing approaches with respect to computational costs [19,20]. Networks were also treated as electric circuits and communities were identified based on notions of voltage drops across networks [21]. Furthermore, an algorithm based on a modified q-state Potts model was presented [22]. Communities are considered as domains with equal spin values near the ground state of the system, which was approximated using Monte Carlo optimisation. Finally, Son et al. developed a random field Ising model to determine the community structure [23]. The ground state problem is equivalent to the maximum flow problems, which can be solved using combinatorial optimisation algorithms.

In more systematic investigations of network properties, the *modularity* metric [24] was introduced as a measure of network partition quality. Network modularity is the fraction of all edges that lie within communities minus the expected value of the same quantity in a graph in which the vertices have the same degrees but edges are placed at random. A modularity value of 0 indicates that the network considered is equivalent to random networks and no obvious community structures are observed; modularity approaching the maximum value of 1, indicates strong community structure.

Newman and Girvan [24] developed a series of divisive algorithms to discover community structures, involving the iterative removal of edges with the highest "betweenness" score to split the network into communities. These algorithms were highly effective at discovering community structures for many testing cases at the cost of very high computational resources when analysing large-scale networks. More computationally efficient algorithms were proposed to tackle networks with larger sizes [25,26]. Newman proved that network modularity can be rewritten as eigenvectors of a modularity matrix and this expression leads to a spectral algorithm for community detection resulting in higher quality solutions when compared to competing approaches [27].

Since proposing the concept of modularity, the community structure detection problem can be posed as an optimisation task which finds an optimal partition at the maximum value for modularity. Simulated annealing was first used to identify functional modules in metabolic networks of twelve organisms from three different superkingdoms by maximising their modularity values [8]. The same optimisation methodology was also applied to analyse and benchmark social networks [28] where a trade-off between quality of solutions and computational requirements was noted. Moreover, the applicability of extremal optimisation was demonstrated through a number of test cases of computer-simulated and real networks [29].

Recently, Fortunato and Barthelemy [30] reported the observation that the optimisation of modularity metric has a resolution limit, so submodules smaller than a certain scale in large networks may fail to be detected since the modularity optimisation procedure tends to combine small communities into larger ones. Kumpula et al. [31] showed that the q-state Potts model introduced by Reichardt and Bornholdt [22] also has a resolution threshold. Both findings raised major concerns of the reliability of modularity optimisation. However, Arenas et al. overcome such problems by proposing a systematic method to discover community structures at different resolution levels using the original modularity concept [32].

Although the presence of resolution limit of modularity maximisation makes some small modules in large networks invisible, modularity is still one of the most widely accepted metrics to detect community structures. All approaches mentioned above are able to achieve good quality modularity values when partitioning networks of various sizes. However, a major limitation is that global optimality of the solutions cannot be guaranteed. Here, a general mathematical programming formulation for the network community structure identification problem is presented where the objective function considered is maximisation of the modularity value and can be solved to global optimality. More importantly, the proposed optimisation model can easily be extended in the future to detect communities more accurately when alternative measures become available. Other additional features such as minimum/maximum module size and balancing among modules can also be incorporated using mathematical programming to aid accurate detection.

The paper is structured as follows: the problem statement for network community detection is defined in the next section. Section 3 presents an MIQP model to detect community structures of a network with the maximum modularity value. Symmetry breaking constraints are then proposed to avoid redundant equivalent solutions thus reducing the computational requirements significantly. The applicability of the proposed mathematical model is demonstrated in Section 4 through the use of four network examples and comparisons of the present methodology with other literature approaches. Finally, some concluding remarks are made in Section 5.

2 Problem statement

Networks are defined by a set of nodes and links connecting them. Each link is undirected and unweighted. Overall, the problem of network community structure identification can be stated as follows:
given:

- an undirected network consisting of N nodes and L links;

determine:

- optimal number of modules;
- node-module allocation;

so as to:

- maximise the network modularity metric.

3 Mathematical formulation

The indices, sets and parameters associated with the mathematical model are listed below:

Indices

n, e	nodes
l	links
m, k	modules

Parameters

N	total number of nodes
L	total number of links
M	total number of modules
d_n	degree of node n
α	minimum module size
β	maximum module size
ε	maximum size difference between any pair of modules

Sets

S	M most connected nodes
AM_n	allowed modules for assignment to node $n \in S$
ML_l	allowable modules for link l
Av_m	nodes allowed assignment to module m
B_n	nodes with higher connectivity than node n

The mathematical formulation is based on the following key variables:

Binary variables

E_m	1 if module m exists; 0 otherwise
X_{lm}	1 if link l belongs to module m; 0, otherwise
Y_{nm}	1 if node belongs to module m; 0, otherwise

Positive variables

L_m	number of links among nodes within module m
D_m	degree of module m

3.1 Objective function

The objective function considered here is the maximisation of the network *modularity* metric as proposed by Newman and Girvan [24]:

$$Q = \sum_m \left[\frac{L_m}{L} - \left(\frac{D_m}{2L} \right)^2 \right]. \tag{1}$$

3.2 Allocation constraints

Each node should be allocated to exactly one module:

$$\sum_m Y_{nm} = 1 \qquad \forall n. \tag{2}$$

Link l belongs to module m if both nodes associated with l (i.e. n and e) are allocated to module m. This logical condition can be written mathematically as:

$$2X_{lm} \leq Y_{nm} + Y_{em} \qquad \forall m, l = \{n, e\}. \tag{3}$$

Constraint (3) can alternatively be disaggregated into two tighter inequalities:

$$X_{lm} \leq Y_{nm} \qquad \forall m, l = \{n, e\}, \tag{4}$$

$$X_{lm} \leq Y_{em} \qquad \forall m, l = \{n, e\}. \tag{5}$$

3.3 Definition of L_m and D_m

L_m is defined as the total number of links within module m:

$$L_m = \sum_l X_{lm} \qquad \forall m \tag{6}$$

D_m is defined similarly to be equal to the sum of the degrees of nodes allocated to module m:

$$D_m = \sum_n d_n Y_{nm} \qquad \forall m. \tag{7}$$

3.4 Additional constraints

One of the key advantages of using mathematical programming approaches is the ease of accommodating user-defined conditions. Here, a number of additional features are formulated mathematically.

First, we describe how minimum and/or maximum module sizes can be incorporated. A binary variable, E_m, is introduced to determine the existence or not of module m. A degeneracy constraint is proposed to enforce that module m is allowed only when the previous module exists (i.e. $E_{m-1} = 1$):

$$E_m \leq E_{m-1} \qquad \forall m = 2, ..., M. \tag{8}$$

Note that if module $m-1$ does not exist (i.e. $E_{m-1} = 0$), then module m, does not exist as well (i.e. $E_m = 0$). Module m is not empty when the following two constraints are active at the same time:

$$\sum_l X_{lm} \geq \alpha \qquad \forall m \tag{9}$$

$$\sum_l X_{lm} \geq \beta \qquad \forall m. \tag{10}$$

The above constraints (9, 10) should be activated only if module m exists and therefore, they should be rewritten as:

$$\sum_l X_{lm} \geq \alpha E_m \qquad \forall m \tag{11}$$

$$\sum_l X_{lm} \leq \beta E_m \qquad \forall m. \tag{12}$$

It is worth mentioning that the above constraints (8, 11, 12) safeguard that all occupied modules are first ranked to avoid equivalent solutions and then module sizes within prespecified bounds are enforced.

Next, we demonstrate how balancing issues among modules, if required, can easily be accommodated in the current optimisation approach. By balancing, we denote that any two non-empty modules m and k, (i.e. $E_m = E_k = 1$) cannot differ by more than a user-defined number of links, ε:

$$|L_m - L_k| \leq \varepsilon \qquad \forall m, k > m. \tag{13}$$

The above absolute-value inequality can mathematically be written as:

$$L_m - L_k \leq \varepsilon \qquad \forall m, k > m \tag{14}$$

$$L_k - L_m \leq \varepsilon \qquad \forall m, k > m. \tag{15}$$

It should be added that the above constraints are activated only if both modules m and k are selected (i.e. $E_m = E_k = 1$). Thus, constraints (14, 15) can be rewritten as:

$$L_m - L_k \leq \varepsilon + \beta(2 - E_m - E_k) \qquad \forall m, k > m \tag{16}$$

$$L_m - L_k \leq \varepsilon + \beta(2 - E_m - E_k) \qquad \forall m, k > m. \tag{17}$$

Table 1. Equivalent solutions for a three-module problem.

	Module 1	Module 2	Module 3
Solution 1	n_1, n_2	n_3, n_7, n_8	n_4, n_5, n_6
Solution 2	n_1, n_2	n_4, n_5, n_6	n_3, n_7, n_8
Solution 3	n_4, n_5, n_6	n_1, n_2	n_3, n_7, n_8
Solution 4	n_4, n_5, n_6	n_3, n_7, n_8	n_1, n_2
Solution 5	n_3, n_7, n_8	n_1, n_2	n_4, n_5, n_6
Solution 6	n_3, n_7, n_8	n_4, n_5, n_6	n_1, n_2

The degeneracy constraint (8) indicates that the value of E_m can be forced to 1 when module k is non-empty (i.e. $E_k = 1$), so constraints (16) and (17) can be simplified as:

$$L_m - L_k \leq \varepsilon + \beta(1 - E_k) \qquad \forall m, k > m \tag{18}$$

$$L_m - L_k \leq \varepsilon + \beta(1 - E_k) \qquad \forall m, k > m. \tag{19}$$

3.5 Symmetry-breaking constraints

It is widely believed that when a set of objects is clustered into a number of modules, any renumbering of the modules generates an equivalent solution [33]. Specifically, if a network ends up with M optimal communities, there are $M!$ equivalent solutions. Table 1 enumerates equivalent solutions for a network example with 8 nodes and 3 modules. Here, two symmetry breaking constraints are proposed to eliminate equivalent solutions and thus to reduce the number of nodes explored during a branch-and bound solution procedure.

Suppose we seek to partition all nodes into M modules. In order to avoid equivalent solutions through the renumbering of modules, each node is allowed to be allocated to one of a particular set of modules, AM_n. First, all nodes are sorted based on their connectivities. For the example shown in Table 1, let us assume that n_1 is the most connected node, n_2 is the second most connected node and so on. The AM_n set is then constructed as: n_1 is allocated to module 1 only; n_2 can be assigned to either module 1 or 2. All other nodes can be allocated to any of the three available modules. Therefore, constraint (2) can be rewritten as the following equality:

$$\sum_{m \in AM_n} Y_{nm} = 1 \qquad \forall n. \tag{20}$$

By activating constraint (20), solutions 3 to 6 in Table 1 can be eliminated as node n_1 is allocated to module 1. It should be mentioned that similar constraints as in (20) have also been reported by Klein and Aronson [33] in the case of cluster analysis.

Since each node n has its own allowable set of modules (AM_n), link l that connects nodes n and e can be allocated to the modules that appear in both AM_n and AM_e. Here, we define set ML_l (allowable modules for link l) as $AM_n \cap AM_e$, where $l = \{n, e\}$. Consequently, constraints (4) and (5) can be replaced by:

$$X_{lm} \leq Y_{nm} \qquad \forall l = \{n, e\}, m \in ML_l \tag{21}$$

$$X_{lm} \leq Y_{em} \qquad \forall l = \{n, e\}, m \in ML_l \tag{22}$$

$$X_{lm} = 0 \qquad \forall l, m \notin ML_l. \tag{23}$$

Table 2. Computational results for illustrative examples.

	OptMod			Hierarchical clustering			Literature approaches	
Examples	OBJ[a]	N_{modu}[b]	CPU (s)	OBJ	N_{modu}	CPU (s)	OBJ	N_{modu}
Zachary	0.4198	4	1.03	0.4198	4	0.33	0.4190 [27]	4
							0.3724 [24]	5
Dolphin	0.5285	5	197.89	0.5084	5	1.10	0.5200 [24]	5
Les Miserables	0.5600	6	55.58	0.5000	19	1.82	0.5400 [24]	11
							0.5460 [28]	5
p53	0.5351	7	1844.31	0.4580	9	3.58	N/A	N/A

[a] Best modularity value found; [b] number of modules.

Another logical condition can be imposed by not allowing node n to be allocated to module m (assuming $m \in AM_n$) if all previous nodes e ($e \in B_n \cap Av_{m-1}$) have not been assigned to module $m-1$ (i.e. $\sum_{e \in (B_n \cap Av_{m-1})} Y_{e,m-1} = 0$, then $Y_{nm} = 0$). Note that B_n denotes the set of nodes e with larger number of connections than n and Av_m denotes the set of nodes that can be assigned to module m. Considering the example shown in Table 1, we then have: $B_{n_1} = \phi$, $B_{n_2} = \{n_1\}$, $B_{n3} = \{n_1, n_2\}$; $Av_1 = \{n_1, n_2, ..., n_8\}$, $Av_2 = \{n_2, n_3, ..., n_8\}$, $Av_3 = \{n_3, n_4, ..., n_8\}$ and so on. As a consequence, if nodes n_1 and n_2 are allocated to module 1, then node n_3 should not be placed to module 3; if nodes n_2 and n_3 are allocated to module 2, then node n_4 should not be assigned to module 4; if nodes n_1, n_2 and n_3 are assigned to module 1, then node n_4 is also excluded from module 3 etc.

Based on the above description, the following logical constraint is proposed:

$$Y_{nm} \leq \sum_{e \in B_n \cap Av_{m-1}} Y_{em-1} \qquad \forall n \geq 3, m = 3..., |AM_n|. \tag{24}$$

When applying the above constraint to the example shown in Table 1, we do not allow node n_3 to be assigned to module 3 as node n_2 appears in module 1 together with node n_1. Thus, solution 2 is eliminated and only solution 1 is feasible.

Symmetry breaking constraints (20) and (24) avoid all other $M! - 1$ equivalent solutions. From our experience, significant computational enhancements are also achieved by only considering the M most connected nodes (defined as set S). Both symmetry breaking constraints are active for all nodes in S:

$$\sum_{m \in AM_n} Y_{nm} = 1 \qquad \forall n \in S \tag{25}$$

$$\sum_{m} Y_{nm} = 1 \qquad \forall n \in S \tag{26}$$

$$Y_{nm} \leq \sum_{e \in (B_n \cap Av_{m-1})} Y_{em-1} \quad \forall n \in S, n \geq 3, m = 3..., |AM_n|. \tag{27}$$

Overall, the resulting mathematical model (OptMod) for determining community structures based on the modularity metric incorporating the above symmetry breaking constraints for network community identification is formulated as follows:

[OptMod]:

$$\text{Maximise } Q = \sum_{m} \left[\frac{L_m}{L} - \left(\frac{D_m}{2L} \right)^2 \right]$$

subject to

constraints $(6-8, 11, 12, 18, 19, 21-23, 25-27)$.

$E_m, X_{lm}, Y_{nm} \in \{0, 1\} \qquad \forall n, m, l$

$L_m, D_m \geq 0 \qquad \forall m.$

The resulting mathematical formulation is a mixed integer quadratic programming (MIQP) model comprising a concave quadratic objective function which is maximised with a set of linear constraints and mixed binary/continuous optimisation variables. The CPLEX mixed integer optimisation solver [34] is used to solve the proposed model to global optimality, due to its convexity, through the branch-and-bound procedure (see, for example, [35]).

4 Computational results

The proposed mathematical model (OptMod) is applied to four network examples from different research areas. All examples are implemented in GAMS (General Algebraic Modeling System) [36] using the CPLEX mixed integer optimisation solver with 0% margin of optimality and 36000 seconds CPU limit. The computational statistics and the optimal modularity values obtained by the proposed MIQP are reported in Table 2. The optimal modularity is then compared with other literature approaches for community structure identification (see Tab. 2). As an alternative, the computational requirements and the best modularity value for each partition from hierarchical clustering are reported. The hierarchical clustering runs are performed by the cluster package using the statistical computing language *R* (`www.r-project.org`). The community structures for all networks are displayed through the Pajek network analysis program (`http://vlado.fmf.uni-lj.si/pub/networks/pajek/`). In each figure, dotted lines are used to reveal the modules obtained.

The first example considers a social network compiled by Zachary [37], who spent two years in observing the social communications between members in a karate club

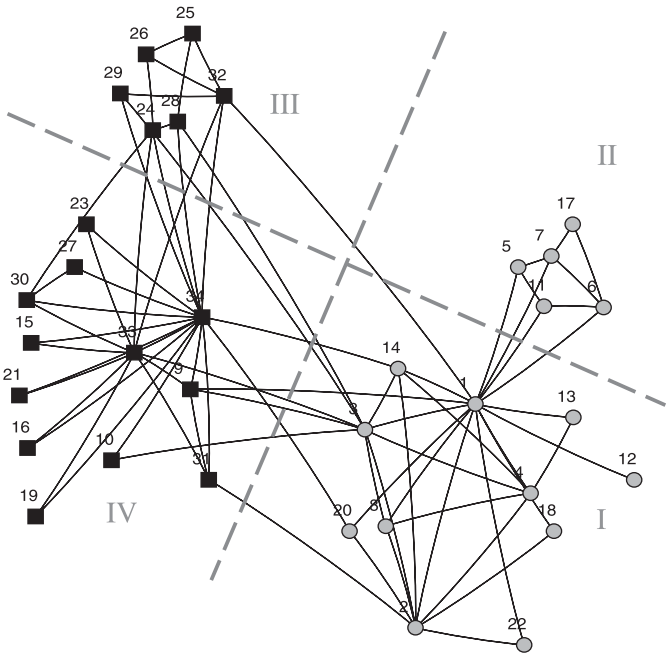

Fig. 2. Optimal community structure for the Zachary's karate club network using OptMod.

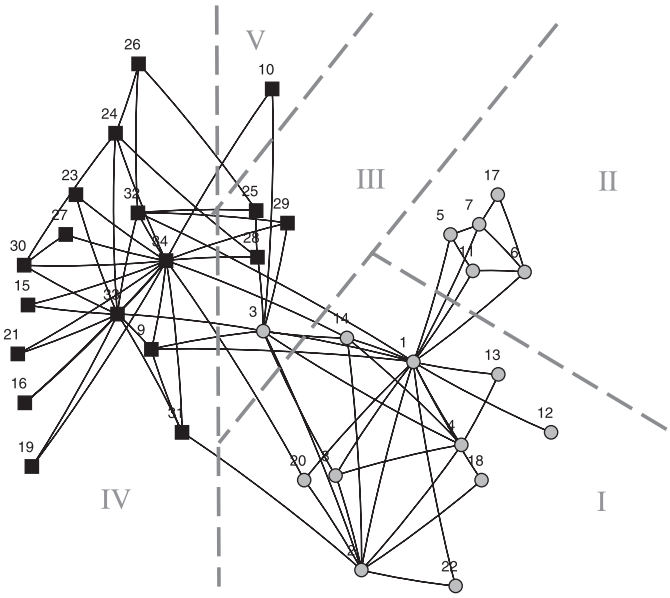

Fig. 3. Community structures identified through betweenness-based iterative algorithm [24] for the Zachary's karate club network.

at an American University. Nodes in the network stand for club members and the links reflect the social relations between them (see Fig. 2). According to the literature [37], the club naturally split in two smaller communities because of a dispute between the club's administrator (around node 1) and the karate teacher (around node 34). This actual division is visualised in Figures 2 and 3 where squares and circles denote the members of each community. Our approach shows that the optimal partition is found at a modularity value of 0.4198 when splitting the network into four independent modules (see Fig. 2 for the optimal partition). It can been seen clearly that the optimal partition from the proposed model perfectly reflects the real community structure (Nodes of modules I and II stand for members around the administrator and modules III and IV belong to the teacher's group). Similar results were produced by hierarchical clustering [16], greedy optimisation algorithm [26], simulated annealing [28], extremal optimisation [29] and the betweenness-based iterative algorithms [24] (see Tab. 2). It is noted that hierarchical clustering identifies the same community structures as the optimal partition. The betweenness-based algorithm [24] finds 5 modules with modularity of 0.3724 (see Fig. 3). Node 10, which is considered as an independent module through that algorithm, should be allocated together with node 34. The betweenness-based algorithm also resulted in one misclassification (node 3) when compared with the actual community structure according to observations by Zachary [37].

The sensitivity of the network modularity value with respect to different values of user-defined link difference between modules, ε (from 1 to 30) in balancing constraints is investigated for the Zachary example. It is observed that the network is partitioned to two subgroups with equal sizes resulting to a modularity value of 0.3718 when ε is less than 6. As the value of ε is further increased, the Zachary network is then divided into 3 or 4 modules with better modularity values. The optimal partition is finally obtained with the maximum modularity value of 0.4198 when the value of ε is larger or equal to 17. It can clearly be seen from Figure 4 that low ε values enforce nodes to distribute evenly within modules while sacrificing the solution quality. Large ε relaxes the balancing constraints thus leading to the optimal partition achieved by the proposed MIQP model. Consequently, user criteria can prioritise the prevalence of either module size balancing or network partitioning optimality. We note that only this example has been analysed with balancing constraints in order to showcase their use. It is obvious that this type of constraint can be used at will in any other examples as required by the user.

In the second example, we present a community of 62 bottlenose dolphins living in Doubtful Sound, New Zealand, constructed by Lusseau [38,39] after seven years of field studies. Each node represents a dolphin and the links in the network are identified based on the significantly frequent communications among them. Using this network as input to the proposed MIQP model, 5 communities are found. Module I and modules II-V reflect the real division observed by Lusseau [39] with zero misclassification and the MIQP model indicates the existence of four smaller communities in the second group (see Fig. 5 for the optimal division by our approach; squares and circles denote the actual partition reported by Lusseau). Comparing with the division from hierarchical clustering (see Figs. 5 and 6), the optimal community structure merge groups I and II of Figure 6, while partition module IV from hierarchical clustering into two groups (see modules II and IV of Fig. 5). Hierarchical clustering also results in 2

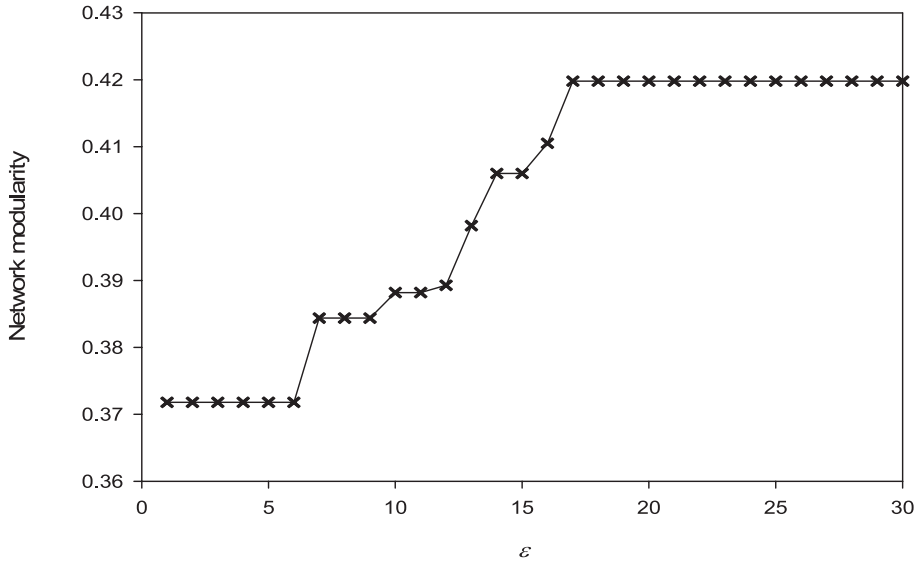

Fig. 4. Sensitivity of optimal modularity values with parameter ε.

misclassification (nodes 8 and 20) when compared with the real partition. It can be seen from Table 2 that all methods partition the dolphin network into 5 communities. Hierarchical clustering and the betweenness-based iterative algorithms algorithm achieved a modularity value of 0.5084 and 0.5200, respectively. Our approach results in a maximum value of 0.5285, which is 3.80% and 1.61% more efficient than the other two literature approaches.

The third example considered here is the network showing the connections between major characters in Victor Hugo's novel of crime and redemption in post-restoration France, Les Miserables. This network was constructed by Knuth [40] where nodes represent characters and edges reveal the coappearence of the corresponding characters in one or more scenes. A modularity value of 0.5400 was reported by Newman and Girvan [24] when partitioning the network into 11 communities and hierarchical clustering results in a modularity value of 0.5000 with 19 communities. However, a number of modules identified by both approaches show few modular characteristics as they contain only one node. According to our model, optimal community presence is identified when the number of modules is optimised to 6 with the maximum modularity value of 0.5600 (better than previous approaches). The optimal partition shown in Figure 7 clearly reflects the plot structure of the novel and the importance of each character in this book: each module corresponds to the stories that the characters are involved in and a number of dominant characters such as Jean Valjean (node 12) and Javert (node 49) act as hubs of their communities (modules I and VI, respectively).

Finally, we apply our model to the p53 protein-protein interaction network constructed by Dartnell et al. [41]. This network involves an annotated protein interaction map in mammalian cell cycle, DNA repair and apoptosis. As a key element in maintaining genomic stability, protein p53 lies in the centre of the network and controls the

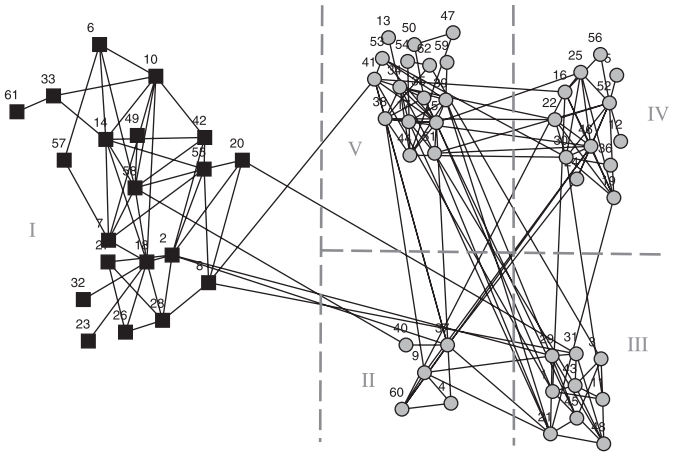

Fig. 5. Optimal community structure for the bottlenose dolphins of Doubtful Sound using OptMod.

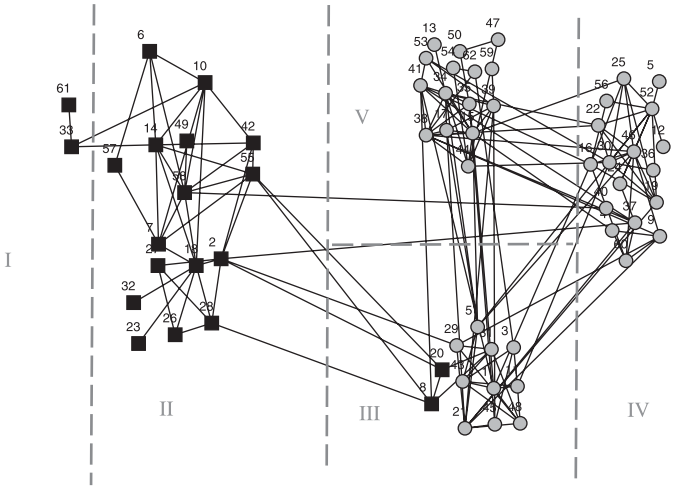

Fig. 6. Community structures identified through hierarchical clustering for dolphins of Doubtful Sound.

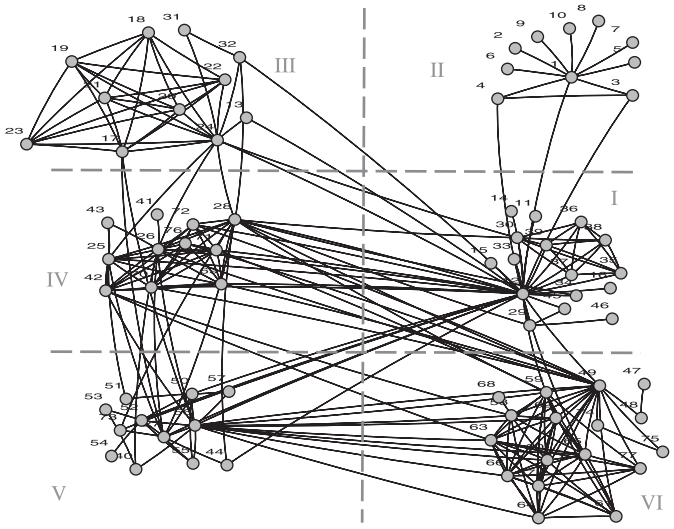

Fig. 7. Optimal community structures for the Les Miserables network.

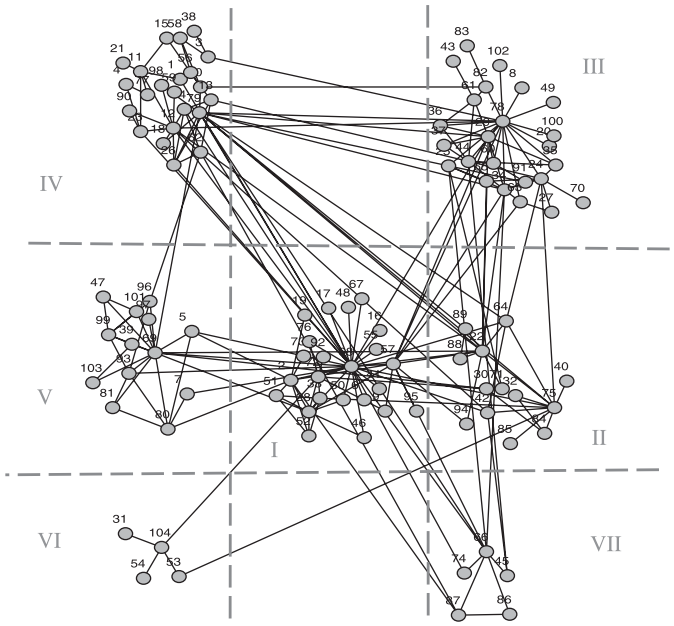

Fig. 8. Optimal community structure for the p53 protein-protein interaction network.

intra- and intercellular signals with gene transcription. The p53 network consisting of 104 proteins and 226 interactions has been proven to have a scale-free topology [41] showing that a vast majority of the nodes are poorly connected while few of them act as hubs with a high centrality. Hierarchical clustering found 9 modules with modularity of 0.4580 with 9 modules. The maximum modularity value (0.5351) is again reported by our model partitioning the p53 network into 7 communities (see Fig. 8). It is not surprising that module I lies in the center of the network and communicates with all other six modules. Node 68 (protein p53), the most central protein to the network, is included in this module.

5 Concluding remarks

Many social, technical and biological systems can be represented as networks of interacting components. Community structures are usually found in those systems where nodes are naturally divided into subgroups with dense within-module connections. Detection of such structures can be vitally beneficial to the study of various complex systems since nodes within the same module may share similar functional properties and novel patterns or functions can be deducted through the analysis of the interacting modules.

In this paper, a rigorous MIQP model has been proposed to identify optimal community structure in complex networks. The objective function considered is maximisation of the network modularity proposed previously [24]. Symmetry breaking constraints have been introduced to avoid the generation of equivalent solutions thus enhancing the computational performance of the proposed model. Our results have shown that global optimal solutions have been achieved for all examples studied.

The search for the optimal network partition with the maximum modularity value is difficult since the solution space grows faster than any power of network size [29]. Danon et al. [42] compared several recent network community identification approaches in terms of sensitivity and efficiency. Computational results indicated that the most accurate methods tend to be computationally expensive and may become prohibitive at large network sizes. It is well understood that the development of faster and more accurate methodologies might be the focus of future research. The main contribution of our proposed model lies in its suitability to find optimal community structures of networks with small and medium sizes. More importantly, the power of mathematical programming is demonstrated by easily incorporating other additional features such as minimum/maximum module size and balancing among modules in the proposed optimisation model. Future work involves the investigation of alternative solution approaches so as to detect communities in larger scale complex networks with optimal or near-optimal modularity values.

The authors thank M.E.J. Newman for providing the Zachary, Dolphin and Les Miserables datasets through his website: `http://www-personal.mich.edu/~mejn`. GX acknowledges support from ORSAS (Overseas Research Students Awards Scheme) and the Centre for Process Systems Engineering.

References

1. A. Barabasi, R. Albert, Science **286**, 509 (1999)
2. M.E.J. Newman, SIAM Rev. **45**, 167 (2003)
3. S. Boccaletti, V. Latora, Y, Moreno, M. Chavez, D.-U. Hwang, Phys. Rep. **424**, 175 (2006)
4. G.W. Flake, S.R. Lawrence, C.L. Giles, F.M. Coetzee, IEEE Comput. **35**, 66 (2002)

5. J.P. Eckmann, E. Moses, Proc. Natl. Acad. Sci. U.S.A. **99**, 5825 (2002)
6. M. Girvan, M.E.J. Newman, Proc. Natl. Acad. Sci. U.S.A. **99**, 7821 (2002)
7. R. Guimera, L. Danon, A.D. Guilera, F. Giralt, A. Arenas, Phys. Rev. E **68**, 065103 (2003)
8. R. Guimera, L.A.N. Amaral, Nature **433**, 895 (2005)
9. P. Holme, M. Huss, H. Jeong, Bioinformatics **19**, 532 (2003)
10. M.R. Garey, D.S. Johnson, *Computers and intractability, A Guide to the theory of NP-completeness* (W.H. Freeman, San Francisco, 1979)
11. D. Fisher, J. Artif. Intell. Res. **4**, 147 (1996)
12. M.E.J. Newman, Eur. Phys. J. B **38**, 321 (2004)
13. S. Boettcher, A.G. Percus, Phys. Rev. E **64**, 026114 (2001)
14. B.W. Kernighan, S. Lin, Bell Syst. Tech. J. **49**, 291 (1970)
15. A. Pothen, H. Simon, K.P. Liou, SIAM J. Matrix. Anal. A **11**, 430 (1990)
16. M. Gustafsson, M. Hornquist, A. Lombardi, Physica A **367**, 559 (2006)
17. A.W. Rives, T. Galitski, Proc. Natl. Acad. Sci. U.S.A. **100**, 1128 (2003)
18. C.V. Mering, E.M. Zdobnov, S. Tsoka, F.D. Ciccarelli, J.B. Pereira-Leal, C.A. Ouzounis, P. Bork, Proc. Natl. Acad. Sci. U.S.A. **100**, 15428 (2003)
19. F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, D. Parisi, Proc. Natl. Acad. Sci. U.S.A. **101**, 2658 (2004)
20. C. Castellano, F. Cecconi, V. Loreto, D. Parisi, F. Radicchi, Eur. Phys. J. B **38**, 311 (2004)
21. F.Wu, B.A. Huberman, Eur. Phys. J. B **38**, 331 (2004)
22. J. Reichardt, S. Bornholdt, Phys. Rev. Lett. **93**, 218701 (2004)
23. S.W. Son, H. Jeong, J.D. Noh, Eur. Phys. J. B **50**, 431 (2006)
24. M.E.J. Newman, M. Girvan, Phys. Rev. E **69**, 026113 (2004)
25. M.E.J. Newman, Phys. Rev. E **69**, 066133 (2004)
26. A. Clauset, M.E.J. Newman, C. Moore, Phys. Rev. E **70**, 066111 (2004)
27. M.E.J. Newman, Proc. Natl. Acad. Sci. U.S.A. **103**, 8577 (2006)
28. A. Medus, G. Acuna, C.O. Dorso, Physica A **358**, 593 (2005)
29. J. Duch, A. Arenas, Phys. Rev. E **72**, 027104 (2005)
30. S. Fortunato, M. Barthelemy, Proc. Natl. Acad. Sci. U.S.A. **104**, 36 (2007)
31. J.M. Kumpula, J. Saramaki, K. Kaski, J. Kertesz, Eur. Phys. J. B **56**, 41 (2007)
32. A. Arenas, A. Fernandez, S. Gomez, e-print `arXiv:physics/0703218` (2007)
33. G. Klein, J.E. Aronson, Nav. Res. Log. **38**, 447 (1991)
34. Ilog, *ILOG CPLEX 10.0 User's Manual* (2006)
35. C.A. Floudas, *Nonlinear and Mixed-Integer Optimisation* (Oxford University Press, New York, 1995)
36. A. Brooke, D. Kendrick, A. Meeraus, R. Raman, *GAMS: A user's guide* (GAMS development Corp. Washington, DC. 1998)
37. W.W. Zachary, J. Anthropol. Res. **33**, 452 (1977)
38. D. Lusseau, Proc. R. Soc. London. Ser. B (Suppl.) **270**, S186 (2003)
39. D. Lusseau, Behav. Ecol. Sociobiol. **54**, 396 (2003)
40. D.E. Knuth, *The Stanford graphbase: a platform for combinatorial computing* (Addison-Wesley, Reading, MA, 1993)
41. L. Dartnell, E. Simeonidis, M. Hubank, S. Tsoka, I.D.L. Bogle, L.G. Papageorgriou, FEBS. Lett. **579**, 3037 (2005)
42. L. Danon, A. Diaz-Guilera, J. Duch, A. Arenas, J. Stat. Mech. P09008 (2005)