European Journal of Operational Research 218 (2012) 316–326

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

Identifying large robust network clusters via new compact formulations of maximum *k*-club problems

Alexander Veremyev[a], Vladimir Boginski[a,b,*]

[a] Department of Industrial and Systems Engineering, University of Florida, 303 Weil Hall, Gainesville, FL 32611, United States
[b] University of Florida Research and Engineering Education Facility (UF-REEF), 1350 N Poquito Road, Shalimar, FL 32579, United States

ARTICLE INFO

Article history:
Received 23 February 2010
Accepted 22 October 2011
Available online 6 November 2011

Keywords:
Combinatorial optimization
Graph theory
Robust network clusters
k-clubs
R-robust *k*-clubs
Compact 0–1 formulations

ABSTRACT

Network robustness issues are crucial in a variety of application areas. In many situations, one of the key robustness requirements is the connectivity between each pair of nodes through a path that is *short enough*, which makes a network cluster more robust with respect to potential network component disruptions. A *k-club*, which by definition is a subgraph of a diameter of at most k, is a structure that addresses this requirement (assuming that k is small enough with respect to the size of the original network). We develop a new compact linear 0–1 programming formulation for finding maximum k-clubs that has substantially fewer entities compared to the previously known formulation ($O(kn^2)$ instead of $O(n^{k+1})$, which is important in the general case of $k > 2$) and is rather tight despite its compactness. Moreover, we introduce a new related concept referred to as an *R-robust k-club* (or, (k,R)-club), which naturally arises from the developed k-club formulations and extends the standard definition of a k-club by explicitly requiring that there must be at least R distinct paths of length at most k between all pairs of nodes. A compact formulation for the maximum R-robust k-club problem is also developed, and error and attack tolerance properties of the important special case of R-robust 2-clubs are investigated. Computational results are presented for multiple types of random graph instances.

1. Introduction

Large-scale complex networks play a crucial role in a great variety of areas nowadays. Although a significant amount of work has been done on studying structural properties of networks in terms of their connectivity, the research on various characteristics of complex networks is far from complete. In addition to the large-scale and complex nature of networks, one often needs to deal with uncertain potential disruptions that can interfere with the operation of a networked system. These issues can be caused by a variety of factors, including man-made and natural disruptions, which may result in failures of components (nodes and/or edges) in the network.

A natural approach to taking into account potential multiple network component failures is to consider *robust network clusters* that ensure a sufficient degree of “robust connectivity” between the nodes. Note that the conventional definition of connectivity (e.g., the existence of a path between every two nodes) may not provide the required robustness characteristics, since a long path between a pair of nodes can make the connection vulnerable, especially if every node and/or edge in the path can potentially fail. In this context, the shorter the path between every pair of nodes is, the more “robust” the corresponding network structure becomes (although special cases of “vulnerable” networks with short connectivity paths can still be constructed). However, the robustness characteristics can be substantially improved if there are multiple distinct paths between every pair of nodes. This would ensure that a network cluster stays connected even if some nodes and/or edges are deleted from the network.

Clearly, a *clique* is a very robust network structure in the aforementioned context, since every two nodes in a clique are directly connected by an edge. It is easy to see that the deletion of any number of nodes from a clique would not violate the clique structure of the remaining nodes. Moreover, a clique with q nodes is guaranteed to remain a connected network if at most $q - 2$ edges are randomly deleted.

There has been a lot of work related to various aspects of finding large cliques in networks (for an extensive review of the maximum clique problem see Bomze et al., 1999). However, in most practical situations, cliques are *overly restrictive* structures, since it is challenging to construct a network with all the possible connections in the presence of obstacles and other limitations in real-life situations. Therefore, several concepts referred to as *clique relaxations* have been introduced. The main idea behind these concepts is to “relax” certain properties of a clique while still maintaining

[*] Corresponding author at: University of Florida Research and Engineering Education Facility (UF-REEF), 1350 N Poquito Road, Shalimar, FL 32579, United States. Tel.: +1 850 833 9355; fax: +1 850 833 9366.

E-mail address: boginski@reef.ufl.edu (V. Boginski).

0377-2217/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2011.10.027

sufficient connectivity and robustness characteristics of the obtained network structures. Note that in many cases the maximum size of these clique relaxations is substantially larger than the maximum size of cliques in the same network, which provides a significant advantage in situations when *large robust clusters* need to be identified.

The ideas for these clique relaxations originally come from the study of social networks; however, these definitions can be utilized in a variety of other application areas, such as communication/information exchange networks, energy networks, etc. There are three main directions for possible relaxations of the clique definition:

1. *Density-based relaxations* – relaxing the requirement for the edge density of a clique to be 1: *quasi-cliques* (γ – *dense subgraphs*) (Abello et al., 2002);
2. *Degree-based relaxations* – relaxing the requirement for the degree of each node in a clique of size q to be $q - 1$: *k-plexes* (Seidman and Foster, 1978);
3. *Path (diameter)-based relaxations* – relaxing the requirement for the length of the path between any two nodes in a clique to be 1: *k-cliques* (Luce, 1950) and *k-clubs* (Mokken, 1979).

A *quasi-clique* (also referred to as a γ-dense subgraph) is a subgraph that has the edge density of at least γ, where $\gamma \in (0,1]$. Clearly, a quasi-clique becomes a clique if $\gamma = 1$. A *k-plex* is a subgraph in which the degree of each node is at least $q - k$ (assuming that q is the number of nodes in this subgraph). A *k-clique* is a subgraph where the length of the path between any two nodes is at most k (note that in this definition other nodes in this path are *not required to belong to the k-clique*), whereas a *k-club* is a subgraph that has a *diameter* of at most k (in this definition, *all the nodes* in the shortest path connecting any given pair of nodes within a k-club have to also belong to this k-club). Obviously, for $k = 1$, a k-plex, a k-clique or a k-club would also be a clique.

Although these definitions are rather straightforward and intuitively clear, mathematically rigorous studies on related optimization aspects (e.g., mathematical programming formulations for finding the largest clique relaxations in graphs) have started to appear only within the past few years. The compact linear 0–1 formulation for the maximum quasi-clique problem with $O(n)$ variables and constraints has been developed by Pattillo et al. (2011). The maximum k-plex problem has been addressed by Balasundaram et al. (2011), where they provide the most compact formulation with n variables and constraints.

In this paper, we will concentrate on the third type of clique relaxations mentioned above and develop new compact formulations for the *maximum k-club problem*. Due to the above considerations, a k-club can be viewed as a "tighter" structure than a k-clique; therefore, it is more applicable for connectivity and clustering problems on networks where robustness issues play an important role. For instance, in the context of real-life sparse networks (e.g., communication networks), identifying the maximum k-club would mean identifying the largest possible cluster in a network that can serve as a system of communication "hubs" that are connected and have short transmission paths between each other. In many other applications (e.g., network-based data mining), maximal k-clubs would denote large tightly connected clusters.

In the next sections, we will present more rigorous definitions of the concepts used in this paper and describe the new compact formulations of the maximum k-club problem. As an alternative to the approach of directly formulating the linear 0–1 problem with $O(n^{k+1})$ entities used by Bourjolly et al. (2002) and Balasundaram et al. (2005), our approach will be based on formulating a preliminary non-linear 0–1 problem for finding the maximum k-club, and then utilizing the structure of the problem to formulate it as a linear 0–1 problem with $O(kn^2)$ entities. Approaches of linearizing non-linear 0–1 problems can potentially be efficient for certain classes of problems. Specifically, linearization techniques for quadratic/polynomial 0–1 (Adams and Forrester, 2007; Chaovalitwongse et al., 2004; Glover and Woolsey, 1974) and fractional 0–1 (Wu, 1997; Prokopyev et al., 2005) problems have been addressed in the literature. In this paper, we will demonstrate that for the considered class of maximum k-club problems efficient linearization techniques that use the special structure of k-clubs can be developed.

Further in this paper, we will also define the new concept of an *R-robust k-club*, which naturally arises from the developed k-club formulation and provides an additional degree of robustness for the considered network clusters. In addition, we will consider an important special case of an R-robust 2-club and demonstrate its attractive properties of error and attack tolerance. We will also present computational results for different types of networks that can be represented by power-law, uniform and other random graph models.

2. Notations, definitions, and previous work

To facilitate further discussion, we will use the following definitions and notations. Denote by $G = (V,E)$ a simple undirected graph with the set of n vertices (nodes) V and the set of edges (links) E. Let $A = \{a_{ij}\}_{i,\ j=1,\ldots,n}$ be the adjacency matrix of G, which is an $n \times n$ 0–1 matrix, where an element $a_{ij} = 1$ if there is an edge (undirected arc) between nodes i and j, and $a_{ij} = 0$ otherwise. Let $d_G(i,j)$ be the length of a shortest path between vertices i and j in G and $d(G) = \max_{i,j\in V} d_G(i,j)$ be the *diameter* of G.

For a subset of vertices $S \subseteq V$, $G(S)$ denotes the subgraph induced by S on G, $G(S) = (S, S \times S \cap E)$. A *clique* C is a subset of V such that the subgraph $G(C)$ induced by C on G has all possible edges. Clearly, the diameter of any clique is 1, since all pairs of vertices are directly connected.

As mentioned in the previous section, a *k-club* is a diameter-based clique relaxation, and in terms of the above notations, it can be formally defined as a subset of vertices $S \subseteq V$ such that the diameter of induced subgraph $G(S)$ is at most k. The maximum k-club problem is computationally challenging, and the decision version of this problem has been shown to be *NP*-complete (Balasundaram et al., 2005).

The only previously known mathematical programming formulation for the general case ($k > 2$) of the maximum k-club problem has been proposed in Bourjolly et al. (2002) and Balasundaram et al. (2005). Since it will be referred to in the analysis later in this paper, we briefly outline this formulation before proceeding with further discussion. For any two vertices $i,j \in V$, let $\mathcal{C}^k_{ij}$ be the set of all paths of length at most k linking i and j, and V_t is the vertex set of path t. For every $i \in V$, let x_i be a binary variable equal to 1 if and only if it belongs to solution. Let y_t be an auxiliary binary variable associated with every path $t \in \cup_{i,j\in V} \mathcal{C}^k_{ij} = \mathcal{C}$. Note that $\left|\mathcal{C}^k_{ij}\right| = O(n^{k-1})$, therefore $|\mathcal{C}| = O(n^{k+1})$. The problem is then formulated as follows:

$$\begin{aligned}
\max \quad & \sum_{i=1}^{n} x_i && (1)\\
\text{subject to} \quad & \sum_{t\in\mathcal{C}^k_{ij}} y_t \geqslant x_i + x_j - 1,\ \forall (i,j) \notin E,\ \mathcal{C}^k_{ij} \neq \emptyset,\\
& y_t \leqslant x_r, \quad \forall t \in \mathcal{C}, \forall r \in V_t,\\
& x_i + x_j \leqslant 1, \quad \forall (i,j) \notin E,\ \mathcal{C}^k_{ij} \neq \emptyset,\\
& x_i, y_t \in \{0,1\}, \quad \forall i \in V, \forall t \in \mathcal{C}.
\end{aligned}$$

For $k > 2$, this linear 0–1 formulation has significant drawbacks: it is rather extensive and generally requires $O(n^{k+1})$ entities; moreover, formulating the constraints requires explicit enumeration of all possible paths of length at most k between all pairs of nodes. This makes solving the maximum k-club problem using this formulation rather challenging and in many cases computationally intractable even for small values of k (e.g., $k = 3,4,5,\ldots$). Clearly, as k increases, the number of entities in this formulation would become extremely large.

In the next sections, we present a new, substantially more compact, linear 0–1 formulation for the maximum k-club problem, which requires $O(kn^2)$ entities. Computational experiments with the corresponding LP relaxations have also shown that the proposed formulation is also rather tight, as the relative gap between exact and LP relaxation objective values does not exceed 1% in many instances, especially those with larger k (this will be discussed in more detail in Section 3.4).

In addition, will also extend the proposed formulation to require the paths between all pairs of nodes to be distinct, which adds extra robustness properties to the considered k-club structures. This consideration will motivate us to introduce the new definition of an *R-robust k-club* and to develop a compact linear integer formulation for the maximum R-robust k-club problem.

3. Compact maximum *k*-club formulation

For the purposes of consistency and clarity, we will start the discussion with the description of the preliminary formulations of several special cases, namely, the maximum 2 and 3 – clubs. We will then proceed with the formulation for the general case of the maximum k-club problem and show how to reduce the number of entities in the linear formulation to $O(kn^2)$.

3.1. Preliminary formulations: maximum 2, 3, k-club problems

3.1.1. Preliminary formulation of the maximum 2-club problem

Consider a simple undirected graph $G = (V,E)$ with n nodes as discussed in the previous section, and let A be the adjacency matrix of G.

Now, consider a problem of finding a maximum 2-club in this graph. Suppose we pick some subgraph G_s, and we want to check whether this subgraph is a 2-club. For these purposes we define $x = (x_1,\ldots,x_n)$ to be a 0–1 vector with $x_i = 1$ if node i belongs to G_s, and $x_i = 0$ otherwise.

The subgraph G_s is a 2-club if its diameter is less than or equal to 2. In other words, every pair of nodes (i,j) is connected directly, or through some other node k. Such a connection of nodes (i,j) can be easily formulated in terms of the following constraint:

$$a_{ij} + \sum_{k=1}^{n} a_{ik}a_{kj}x_k \geqslant x_ix_j,$$

Using the simple linearization, the problem of finding the maximum 2-club in the graph G can be formulated as

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} x_i \\ \text{subject to } & a_{ij} + \sum_{k=1}^{n} a_{ik}a_{kj}x_k \geqslant x_i + x_j - 1, \\ & x_i \in \{0,1\}, \end{aligned}$$

where $i = 1,\ldots,n$; $j = i + 1,\ldots,n$.

Let $\delta(i) = \{j : a_{ij} = 1\}$ be a neighborhood of node i. Using this definition, the problem formulation can be rewritten as

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} x_i \\ \text{subject to } & \sum_{k\in\delta(i)\cap\delta(j)} x_k \geqslant x_i + x_j - 1, \\ & x_i \in \{0,1\}, \end{aligned}$$

where $i = 1,\ldots,n$; $j = i + 1,\ldots,n$; $j \notin \delta(i)$.

The number of constraints in this formulation depends on the edge density of the graph G, and all the formulations presented below can also be easily rewritten in terms of $\delta(i)$ notations to reduce the number of entities. This might be useful for computational purposes, but for simplicity of understanding we keep formulations in the presented format.

3.1.2. Preliminary formulation of the maximum 3-club problem

Using the same logic, we can formulate the maximum 3-club problem as follows:

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} x_i \\ \text{subject to } & a_{ij} + \sum_{k=1}^{n} a_{ik}a_{kj}x_k + \sum_{k=1}^{n}\sum_{m=1}^{n} a_{ik}a_{km}a_{mj}x_kx_m \geqslant x_i + x_j - 1, \\ & x_i \in \{0,1\}, \end{aligned}$$

where $i = 1,\ldots,n$; $j = i + 1,\ldots,n$.

This is the problem with a linear objective and quadratic constraints. In a standard and straightforward linearization approach (a more efficient alternative to this approach will be proposed later in the paper), one can introduce new variables $w_{ij} = x_ix_j$ to linearize this problem as follows:

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} x_i \\ \text{subject to } & a_{ij} + \sum_{k=1}^{n} a_{ik}a_{kj}x_k + \sum_{k=1}^{n}\sum_{m=1}^{n} a_{ik}a_{km}a_{mj}w_{km} \geqslant x_i + x_j - 1, \\ & w_{ij} \leqslant x_i, \quad w_{ij} \leqslant x_j, \ w_{ij} \geqslant x_i + x_j - 1, \\ & x_i, w_{ij} \in \{0,1\}, \end{aligned}$$

where $i = 1,\ldots,n$; $j = i + 1,\ldots,n$. This formulation is linear and contains $O(n^2)$ 0–1 variables and $O(n^2)$ constraints.

3.1.3. Preliminary formulation of the maximum k-club problem

Using the similar logic and notations as above, the maximum k-club problem can be represented as:

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} x_i \\ \text{subject to } & a_{ij} + \sum_{k=1}^{n} a_{ik}a_{kj}x_k + \sum_{k=1}^{n}\sum_{m=1}^{n} a_{ik}a_{km}a_{mj}x_kx_m \\ & + \sum_{k=1}^{n}\sum_{m=1}^{n}\sum_{t=1}^{n} a_{ik}a_{km}a_{mt}a_{tj}x_kx_mx_t + \cdots \\ & + \sum_{i_1=1}^{n}\sum_{i_2=1}^{n}\cdots\sum_{i_{k-2}=1}^{n}\sum_{i_{k-1}=1}^{n} a_{ii_1}a_{i_1i_2}\ldots a_{i_{k-2}i_{k-1}}a_{i_{k-1}j}x_{i_1}\ldots x_{i_{k-1}} \geqslant x_i + x_j - 1, \\ & x_i \in \{0,1\}, \end{aligned}$$

where $i = 1,\ldots,n$; $j = i + 1,\ldots,n$.

This formulation can be also linearized and simplified using the same standard approaches as above, and the resulting linear formulation will have $O(n^{k-1})$ variables and constraints. However, in the next section, we will show that the size of this formulation can be substantially reduced by applying a more efficient linearization technique that employs the special structure of k-clubs.

3.2. Compact linear integer programming formulation of the maximum k-club problem

In this section, we describe how to linearize the maximum k-club problem in order to obtain a more compact formulation. The idea of this linearization is to define new variables $w_{ij}^{(l)}(i,j = 1,\ldots,n; l = 2,\ldots,k)$, representing the number of distinct paths of distance l from node i to j in the subgraph $G_s(x)$ defined by vector $(x_1,\ldots,x_n)$. If node i or j does not belong to $G_s(x)$, then $w_{ij}^{(l)} = 0$. We show that there are only $O((k-1)n^2)$ variables and constraints.

For $l = 2$ we can write

$$w_{ij}^{(2)} = x_i x_j \sum_{k=1}^{n} a_{ik} a_{kj} x_k.$$

Since $w_{ij}^{(2)} \leqslant n$, we can linearize it as follows:

$$w_{ij}^{(2)} \leqslant \sum_{k=1}^{n} a_{ik} a_{kj} x_k + n(2 - x_i - x_j), \quad w_{ij}^{(2)} \geqslant \sum_{k=1}^{n} a_{ik} a_{kj} x_k - n(2 - x_i - x_j),$$
$$w_{ij}^{(2)} \leqslant n x_i, \quad w_{ij}^{(2)} \geqslant -n x_i, \quad w_{ij}^{(2)} \leqslant n x_j, \quad w_{ij}^{(2)} \geqslant -n x_j.$$

Other additional variables $w_{ij}^{(l)}$ can be found recursively, since

$$w_{ij}^{(l)} = x_i \sum_{k=1}^{n} w_{kj}^{(l-1)} a_{ik}.$$

Similarly, using the fact that $w_{ij}^{(l)} \leqslant n^{l-1}$ we can linearize it as

$$w_{ij}^{(l)} \leqslant \sum_{k=1}^{n} w_{kj}^{(l-1)} a_{ik} + n^{l-1}(1 - x_i), \quad w_{ij}^{(l)} \geqslant \sum_{k=1}^{n} w_{kj}^{(l-1)} a_{ik} - n^{l-1}(1 - x_i),$$
$$w_{ij}^{(l)} \leqslant n^{l-1} x_i, \quad w_{ij}^{(l)} \geqslant -n^{l-1} x_i.$$

Putting all these constraints together, the maximum k-club problem can be formulated as

$$\begin{aligned}
\max \quad & \sum_{i=1}^{n} x_i \\
\text{subject to} \quad & \sum_{l=2}^{k} w_{ij}^{(l)} \geqslant x_i + x_j - 1, i = 1, \ldots, n, j \notin \delta(i), \\
& w_{ij}^{(2)} \leqslant \sum_{k=1}^{n} a_{ik} a_{kj} x_k + n(2 - x_i - x_j), \\
& w_{ij}^{(2)} \geqslant \sum_{k=1}^{n} a_{ik} a_{kj} x_k - n(2 - x_i - x_j), \\
& w_{ij}^{(2)} \leqslant n x_i, \quad w_{ij}^{(2)} \geqslant -n x_i, \quad w_{ij}^{(2)} \leqslant n x_j, \quad w_{ij}^{(2)} \geqslant -n x_j
\end{aligned}$$

and for $l = 3, \ldots, k$

$$w_{ij}^{(l)} \leqslant \sum_{k=1}^{n} w_{kj}^{(l-1)} a_{ik} + n^{l-1}(1 - x_i), \quad w_{ij}^{(l)} \geqslant \sum_{k=1}^{n} w_{kj}^{(l-1)} a_{ik} - n^{l-1}(1 - x_i),$$
$$w_{ij}^{(l)} \leqslant n^{l-1} x_i, \quad w_{ij}^{(l)} \geqslant -n^{l-1} x_i,$$
$$x_i \in \{0, 1\}, \quad w_{ij} \in Z^{+}.$$

where $i,j = 1, \ldots, n$.

3.3. Compact linear 0–1 formulation of the maximum k-club problem

In this subsection we further refine the compact linear integer programming formulation described above and transform that formulation to an equivalent linear 0–1 programming problem. Recall that $w_{ij}^{(l)}, (i,j = 1, \ldots, n; l = 2, \ldots, k)$ represents the number of distinct paths of length l from node i to j in the subgraph $G_s(x)$ defined by the 0–1 vector $(x_1, \ldots, x_n)$. It means that $w_{ij}^{(l)}$ is a non-negative integer variable with the upper bound of n^{l-1}. However, in the context of the standard maximum k-club problem, we do not need to know the number of distinct paths of distance l from node i to j in the subgraph $G_s(x)$. In this sense, these variables contain a lot of "unnecessary" information. Since we only need to check if there is *at least one path* of length l between nodes i and j, the only information we need to know about the variable $w_{ij}^{(l)}$ is whether it has a zero or a nonzero value.

To address this consideration, we define 0–1 variables $\upsilon_{ij}^{(l)}, (i,j = 1, \ldots, n; l = 2, \ldots k)$ as follows: $\upsilon_{ij}^{(l)} = 1$ if there exists *at least one path of length l* from node i to j in the subgraph $G_s(x)$ defined by vector $(x_1, \ldots, x_n)$, and $\upsilon_{ij}^{(l)} = 0$ otherwise.

For $l = 2$, we can write

$$\upsilon_{ij}^{(2)} = \min\left\{ x_i x_j \sum_{k=1}^{n} a_{ik} a_{kj} x_k, 1 \right\}.$$

This equality can be linearized as follows:

$$\upsilon_{ij}^{(2)} \leqslant x_i, \quad \upsilon_{ij}^{(2)} \leqslant x_j,$$
$$\upsilon_{ij}^{(2)} \leqslant \sum_{k=1}^{n} a_{ik} a_{kj} x_k, \quad \upsilon_{ij}^{(2)} \geqslant \frac{1}{n}\left(\sum_{k=1}^{n} a_{ik} a_{kj} x_k \right) + (x_i + x_j - 2),$$

where $\upsilon_{ij}^{(2)}$ is a 0–1 variable for any $1 \leqslant i < j \leqslant n$. Other additional variables can be found recursively, since

$$\upsilon_{ij}^{(l)} = \min\left\{ x_i \sum_{k=1}^{n} \upsilon_{kj}^{(l-1)} a_{ik}, 1 \right\}.$$

Similarly, we can linearize it as

$$\upsilon_{ij}^{(l)} \leqslant x_i, \quad \upsilon_{ij}^{(l)} \leqslant \sum_{k=1}^{n} a_{ik} \upsilon_{kj}^{(l-1)}, \quad \upsilon_{ij}^{(l)} \geqslant \frac{1}{n}\left(\sum_{k=1}^{n} a_{ik} \upsilon_{kj}^{(l-1)} \right) + (x_i - 1),$$

where $\upsilon_{ij}^{(l)}$ is a 0–1 variable for any $2 \leqslant l \leqslant k$ and $1 \leqslant i < j \leqslant n$.

Putting all these constraints together, the maximum k-club problem can now be formulated as the following linear 0–1 problem:

$$\max \sum_{i=1}^{n} x_i$$

subject to

$$\sum_{l=2}^{k} \upsilon_{ij}^{(l)} \geqslant x_i + x_j - 1, \quad i = 1, \ldots, n, j \notin \delta(i)$$

for $j > i = 1, \ldots, n$,

$$\upsilon_{ij}^{(2)} \leqslant x_i, \quad \upsilon_{ij}^{(2)} \leqslant x_j, \quad \upsilon_{ij}^{(2)} \leqslant \sum_{k=1}^{n} a_{ik} a_{kj} x_k,$$
$$\upsilon_{ij}^{(2)} \geqslant \frac{1}{n}\left(\sum_{k=1}^{n} a_{ik} a_{kj} x_k \right) + (x_i + x_j - 2),$$

and for $l = 3, \ldots, k$; $j > i = 1, \ldots, n$,

$$\upsilon_{ij}^{(l)} \leqslant x_i, \quad \upsilon_{ij}^{(l)} \leqslant \sum_{k=1}^{n} a_{ik} \upsilon_{kj}^{(l-1)}, \quad \upsilon_{ij}^{(l)} \geqslant \frac{1}{n}\left(\sum_{k=1}^{n} a_{ik} \upsilon_{kj}^{(l-1)} \right) + (x_i - 1),$$
$$x_i, \upsilon_{ij}^{(l)} \in \{0, 1\}, \; i,j = 1, \ldots, n; \; l = 2, \ldots, k.$$

It should be noted that for tightness purposes $1/n$ in the above constraints can be substituted for

$$\frac{1}{\sum_{k=1}^{n} a_{ik} a_{kj}}.$$

Although we will use these tighter constraints for computational studies, for the sake of readability we will continue further discussion in the next sections using the problem formulation in the previous format.

3.4. Tightness of maximum k-club linear 0–1 problem formulations

Before proceeding with further material, an important issue that needs to be discussed here is the tightness of different linear 0–1 formulations of the maximum k-club problem. As mentioned above, the previously developed linear 0–1 formulation (1) with $O(n^{k+1})$ entities was proposed in Bourjolly et al. (2002) and Balasundaram et al. (2005), and it would be interesting to compare the tightness of LP relaxations for that formulation and the one proposed in this paper (note that this comparison would be significant for larger values of k, since for $k = 2$ the two formulations are essentially the same, as formulation (1) can be simplified for this special case).

One can hypothesize that formulation (1) is tight due to the large number of constraints; however, it is computationally challenging to verify it directly by performing any meaningful computational experiments and obtain exact 0–1 and LP relaxation solutions for this formulation for non-trivial problem instances (i.e., connected graphs that are large enough, so that the maximum k-clubs for larger k do not coincide with the whole graph). For instance, for $k \geqslant 4$ and $n \geqslant 100$ (which is a reasonable "lower end" of the order of magnitude for n to produce non-trivial solutions for relatively small values of k), the corresponding problems would contain at least around 10^{10} entities, and for slightly larger values of n and k (e.g., $n = 100$, $k = 7$), the size of the problems would grow to around 10^{16} entities, which makes not only the 0–1 problems, but also their LP relaxations, computationally intractable even for the considered relatively small values of n and k.

It should be also mentioned that neither the linear 0–1 formulation (1) from Bourjolly et al. (2002) and Balasundaram et al. (2005), nor its LP relaxation, was ever implemented for any graph instances in the previous literature (note that one needs to explicitly enumerate all the paths of length no more than k between all pairs of nodes in order to just formulate the problem for each specific instance, which makes it rather challenging to implement this formulation even before solving the problem). Bourjolly et al. (2002) used an exact algorithm based on DROP (a heuristic developed in an earlier work by the same authors in Bourjolly et al. (2000)) for solving the maximum k-club problem in their computational experiments. The reported computational experiments suggested that this exact algorithm works well for $k = 2$ in graphs with up to 200 nodes; however, for $k = 3$ and $k = 4$ the algorithm was consistently able to handle only the graphs with up to $n = 100$ nodes, and no computational results were reported for for $k > 4$ (this may be partially due to the fact that the performance of the algorithm depends on the quality of bounds produced by the DROP heuristic, and this quality may be affected by larger values of n and k). Balasundaram et al. (2005) used the special case of both formulations with $k = 2$ for their computational experiments; however, for $k > 2$, another (simpler) formulation for the maximum k-clique problem was used in order to find the maximum 3-clique (which coincidentally was also a 3-club) in the S. Cerevisiae network (this network will be considered in Section 5).

Unlike formulation (1), the formulation proposed in this paper, as well as its LP relaxation, is easily implementable and contains a reasonable number of entities for the same ranges of n and k as mentioned above in this subsection. Detailed computational results summarizing the exact and LP relaxation solutions for multiple problem instances will be presented in Section 5. Interestingly, in many considered instances (especially for $k = 5,6,7$), the tightness of our formulation (i.e., the relative gap between exact and LP relaxation objective values) is within a few percent, which shows that the proposed formulation is not only compact, but also rather tight, especially for more computationally challenging cases with larger values of k.

In addition to the computational analysis, it is also important to observe some general properties of the LP relaxations of both considered linear 0–1 formulations, which hold for any graph, regardless of n and k. Specifically, one can easily verify that $x_i = 0.5$, $\forall i = 1,\ldots,n$ is a *feasible solution* for both the formulation proposed in this paper and the formulation (1) by Bourjolly et al. (2002), which implies that the optimal objective values for initial LP relaxations for both formulations are *at least* $n/2$. Therefore, it is clear that both formulations will generally be not very tight for all problem instances with the size of the maximum k-club *not exceeding* $n/2$ (i.e., for larger values of n and/or smaller values of k). For all

Table 1
Average exact size of the maximum k-club ($k = 2,\ldots,7$) for the considered random graph instances (10 instances for each combination of $n, \hat{p}$), the average CPU time for solving the linear 0–1 problem, and average tightness calculated as the relative gap between the exact and the LP relaxation objective values. An asterisk (*) next to the relative gap percentage means that the LP relaxation objective was exactly equal to $n/2$, as is the case for all considered problem instances with the exact maximum k-club size not exceeding $n/2$. As mentioned in Section 3.4, this gap is not worse than the relative LP relaxation gap in formulation (1) by Bourjolly et al. (2002).

Graph parameters		Maximum k-club size, CPU time, and LP gap (%), for $k = \cdots$					
		2	3	4	5	6	7
$n = 100$	$\hat{p} = 2\%$	**7.3**	**12.2**	**21.1**	**31.4**	**44.1**	**55.7**
		0.9 s	2.9 s	5.1 s	8.5 s	12.9 s	15.1 s
		85.4%*	75.6%*	57.8%*	37.2%*	16.4%*	1.6%
	$\hat{p} = 3\%$	**8.8**	**16.3**	**32.3**	**57.3**	**78.1**	**88.3**
		1.0 s	5.8 s	15.4 s	35.7 s	27.7 s	45.1 s
		82.4%*	67.4%*	35.4%*	2.7%	0.1%	0.0%
	$\hat{p} = 4\%$	**10.6**	**21.1**	**52.4**	**85.5**	**95.5**	**97.9**
		1.1 s	14.4 s	51.3 s	21.6 s	37.2 s	62.7 s
		78.8%*	57.8%*	7.9%	0.1%	0.0%	0.0%
$n = 200$	$\hat{p} = 1\%$	**7.9**	**12.6**	**23.8**	**35.6**	**55.4**	**80.5**
		6.5 s	15.3 s	19.5 s	77.9 s	151.4 s	364.6 s
		92.1%*	87.4%*	76.2%*	64.4%*	44.6%*	21.5%*
	$\hat{p} = 1.5\%$	**9.5**	**16.6**	**34.8**	**62.0**	**117.0**	**158.9**
		7.3 s	27.8 s	74.5 s	760.8 s	1022.8 s	554.4 s
		90.5%*	83.4%*	65.2%*	38.0%*	2.7%	0.1%
	$\hat{p} = 2\%$	**11.5**	**20.4**	**48.6**	**133.4**	**183.2**	**193.3**
		8.9 s	77.8 s	1259.7 s	940.1 s	472.5 s	879.9 s
		88.5%*	79.6%*	51.4%*	3.4%	0.0%	0.0%
$n = 300$	$\hat{p} = 0.5\%$	**7.0**	**10.0**	**15.9**	**21.3**	**29.2**	**36.5**
		20.1 s	36.9 s	48.0 s	56.7 s	83.4 s	111.3 s
		95.3%*	93.3%*	89.4%*	85.8%*	80.5%*	75.7%*
	$\hat{p} = 1.0\%$	**10.3**	**17.4**	**37.8**	**60.7**	**123.4**	**207.1**
		23.8 s	66.8 s	175.4 s	1633.2 s	26532.5 s	19934.1 s
		93.1%*	88.4%*	74.8%*	59.5%*	18.3%*	0.7%
	$\hat{p} = 1.5\%$	**12.8**	**25.1**	**62.0**	**187.4**	**275.4**	**292.5**
		30.6 s	285.6 s	5599.4 s	34475.1 s	2334.7 s	4782.4 s
		91.5%*	83.3%*	58.7%*	5.3%	0.0%	0.0%

such instances, the lower bound on the relative LP relaxation gap for both formulations is essentially determined by the structure of each specific graph and the corresponding maximum k-club size.

Furthermore, as it will be indicated in Section 5, computational experiments suggest that the optimal objective value for the LP relaxation of the proposed formulation is always *exactly equal* to $n/2$ for all considered instances with the maximum k-club not exceeding $n/2$. Although it is not clear if this holds for all such instances, since a formal proof is challenging due to a rather complex recursive nature of the constraints in this formulation, the conducted computational study (90 graph instances and 540 problem instances, as outlined in Section 5) suggests that our formulation is generally at least as good (or, at the very least, not substantially worse) than the one by Bourjolly et al. (2002) in terms of tightness due to the following summarizing arguments:

- For all instances with size of the maximum k-club *not exceeding* $n/2$, the optimal LP relaxation objective for our formulation ($n/2$, with $x_i = 0.5\ \forall i = 1,\ldots,n$) is equal to the lower bound for the optimal LP relaxation objective of the formulation by Bourjolly et al. (2002), since $x_i = 0.5\ \forall i = 1,\ldots,n$ is also feasible for that formulation;
- For all instances with the size of the maximum k-club *exceeding* $n/2$, the average relative gap between the exact and LP relaxation objectives for our formulation is very small, and for some instances it is exactly equal to 0% (see Table 1).

As a concluding remark of this section, we mention that although it is generally beneficial to have a tight formulation for a problem, solving the LP relaxation by itself will not necessarily help one to identify the exact set of nodes that are included into the maximum k-club, as it will only provide an upper bound on the maximum k-club cardinality. Therefore, the main emphasis of this paper is still on providing a linear 0–1 formulation for the considered problem that would be computationally tractable and able to obtain *exact* optimal solutions in a reasonable time at least for moderate-size graph instances. Later in this paper, we present the results of computational experiments that demonstrate that this is indeed the case, even for larger values of k (up to $k = 7$). As it will be discussed in Section 5, the CPU time does not significantly increase with k for the considered graphs. This attractive characteristic allowed us to perform the the first known computational experiments that produced exact solutions to the maximum k-club problems for $k = 5,6,7$ in graphs with up to 300 nodes, which is a substantial improvement over all previously reported computational results in terms of both n and k.

4. *R*-robust *k*-clubs

Before presenting the computational results for the proposed formulation and its LP relaxation, we define and analyze an important extension of the standard k-club concept, which naturally follows from the above material.

Due to the network robustness considerations discussed in the introductory sections of this paper, we note that the existence of a "short" path between any two nodes in a k-club (for relatively small values of k) is a useful property in terms of robustness characteristics; however, the main drawback of the standard definition of a k-club is that the considered paths are not required to be distinct, which means that k-clubs can still be vulnerable to targeted attacks that destroy appropriate network components. To address this drawback, we propose to define another type of network clusters, which have a property that *multiple* short paths exist between any pair of nodes. More formally, we define a subgraph G_s to be an *R-robust k-club* (or, a *(k,R)-club*) if there are *at least R internally node-disjoint paths of length at most k* between every pair of nodes in the subgraph G_s. It should be noted that although network clusters that have this property have good attack tolerance characteristics, developing mathematical programming techniques for finding the exact solution of the maximum (k,R)-club problem is not an easy task. To relax this definition, one can introduce alternative requirements for disjoint paths, such as: (1) *internally edge-disjoint* paths that may share common nodes; or (2) paths that have a difference in *at least one edge*.

Here and further in the paper, we will consider R-robust k-clubs in the context of the latter (relaxed) definition, that is, two paths are considered distinct if they have a difference in *at least one edge*. As it will be shown below, the aforementioned formulation for the maximum k-club problem can be directly generalized to this definition of an R-robust k-club; however, it cannot be extended to R-robust k-clubs with internally node-disjoint paths. Despite the difficulties in dealing with the general case of the problem, it is important to note that for the special case of an *R-robust 2-club*, all of the above definitions of disjoint paths are equivalent, which means that in this case one can just assume that two paths are distinct if they have a difference in *at least one edge*, and this would automatically imply that these paths are node-disjoint and edge-disjoint.

In this section, we introduce a compact formulation for the maximum R-robust k-club problem in the relaxed form. We will show that it can be derived from the maximum k-club problem formulation presented above. Furthermore, we will formally discuss certain robustness properties of the aforementioned special case of *R-robust* 2-*clubs*, which address the issues of *error/attack tolerance* (i.e., the resiliency to potential multiple failures of nodes and/or edges in a network).

4.1. Maximum R-robust k-club problem

Here we provide a compact linear integer formulation for the problem of finding a maximum R-robust k-club in the context of the relaxed definition mentioned above. Recall that when we formulated the maximum k-club problem we used variables $w_{ij}^{(l)} (i,j = 1,\ldots,n; l = 2,\ldots,k)$, which represent the number of distinct paths of distance l from node i to j in the subgraph $G_s(x)$ defined by vector $(x_1,\ldots,x_n)$. Thus, the problem formulation of finding the maximum R-robust k-club is very similar to the problem of finding the maximum k-club with the only difference that we require

$$a_{ij} + \sum_{l=2}^{k} w_{ij}^{(l)} \geqslant R(x_i + x_j - 1), \quad i < j = 1,\ldots,n,$$

instead of

$$\sum_{l=2}^{k} w_{ij}^{(l)} \geqslant x_i + x_j - 1, \quad i = 1,\ldots,n, j \notin \delta(i).$$

Thus, the problem formulation can be written as follows:

$$\begin{aligned}
\max \quad & \sum_{i=1}^{n} x_i \\
\text{subject to } & a_{ij} + \sum_{l=2}^{k} w_{ij}^{(l)} \geqslant R(x_i + x_j - 1), \\
& w_{ij}^{(2)} \leqslant \sum_{k=1}^{n} a_{ik}a_{kj}x_k + n(2 - x_i - x_j), \\
& w_{ij}^{(2)} \geqslant \sum_{k=1}^{n} a_{ik}a_{kj}x_k - n(2 - x_i - x_j), \\
& w_{ij}^{(2)} \leqslant nx_i, \quad w_{ij}^{(2)} \geqslant -nx_i, \quad w_{ij}^{(2)} \leqslant nx_j, \quad w_{ij}^{(2)} \geqslant -nx_j
\end{aligned}$$

and for $l = 3,\ldots,k$

$$w_{ij}^{(l)} \leqslant \sum_{k=1}^{n} w_{kj}^{(l-1)} a_{ik} + n^{l-1}(1-x_i), \quad w_{ij}^{(l)} \geqslant \sum_{k=1}^{n} w_{kj}^{(l-1)} a_{ik} - n^{l-1}(1-x_i),$$
$$w_{ij}^{(l)} \leqslant n^{l-1}x_i, \quad w_{ij}^{(l)} \geqslant -n^{l-1}x_i,$$
$$x_i \in \{0,1\}, \quad w_{ij} \in Z^{+},$$

where $i,j = 1,\ldots,n$.

Note that the recursive method of calculating variables does not allow us to extend this formulation and require all paths to be node-disjoint. Clearly, the existence of certain number of node-disjoint paths is more desirable in practice, since it guarantees that this cluster has the certain level of attack tolerance. In the next section we will consider a special case with $k = 2$ where the node disjoint requirement is satisfied.

4.2. Important special case: R-robust 2-clubs

As it was noted above, in the case of *R*-robust 2-clubs, any two nodes will have at least *R* completely distinct paths connecting them; that is, these paths will not have any edges/nodes in common. As it will be shown in the next subsection, *R*-robust 2-clubs have very attractive error and attack tolerance properties. Before we proceed with these considerations, we present the formulation of the maximum *R*-robust 2-club problem, which in this special case will have only 0–1 variables, rather than integer variables for the general case of the maximum *R*-robust *k*-club problem (in the relaxed form) considered above.

The formulation of this problem is rather compact and can be written as follows:

$$\max \quad \sum_{i=1}^{n} x_i$$
$$\text{subject to } a_{ij} + \sum_{k=1}^{n} a_{ik}a_{kj}x_k \geqslant R(x_i + x_j - 1),$$
$$x_i \in \{0,1\},$$

where $i = 1,\ldots,n$; $j = i+1,\ldots,n$.

4.3. Error and attack tolerance properties of R-robust 2-clubs

In this subsection, we consider in detail the properties of an important special case of *R*-robust *k*-clubs – an *R*-robust 2-club. As it has been mentioned before, the main attractive feature of *R*-robust 2-clubs is the fact that all *R* paths between any two nodes will be *completely distinct*, that is, they will not have any edges/nodes in common.

An illustrative example of a 2-robust 2-club is given in Fig. 5(b). If one compares the structure of this 2-robust 2-club to the structure of the regular 2-club in the same network (see Fig. 5(a)), it can be easily seen that the deletion of the central node from the 2-club will completely destroy the connectivity of this cluster; however, the deletion of any one node or edge from the 2-robust 2-club will not only preserve the *connectivity* of this cluster, but it also will not violate its 2-*club structure* (i.e., all the remaining nodes will still be connected through a path of at most two edges).

This observation leads us to some interesting considerations regarding the *error and attack tolerance* of *R*-robust 2-clubs. *Attacks* on networks can be defined as "targeted" disruptions that attempt to destroy certain components of the network (nodes or edges) in order to interfere with network connectivity. A related notion of *errors*, which essentially represent random (not targeted) disruptions of network components, can also be considered. The ability of a network to maintain certain connectivity characteristics in the presence of errors and/or attacks is referred to as the *error and/or attack tolerance of a network*. A well-known experimental study of error and attack tolerance of power-law and uniform random networks with respect to node failures is presented in Albert et al. (2000).

Clearly, the issue of error and attack tolerance of a network is important in a variety of applications; moreover, these issues need to be generalized and considered with respect to both *node failures* and *edge failures*. In this context, the definition of an *R*-robust 2-club is attractive, since it explicitly addresses the error and attack tolerance properties of these network clusters. Specifically, the following facts can be easily established.

Proposition 1. *The deletion of any one node from an R-robust 2-club guarantees that the remaining nodes and edges form at least an (R − 1)-robust 2-club.*

Proposition 2. *The deletion of any one edge from an R-robust 2-club guarantees that the remaining nodes and edges form at least an (R − 1)-robust 2-club.*

From these observations, a more general statement characterizing error and attack properties of *R*-robust 2-clubs immediately follows.

Proposition 3. *The deletion of any (R − 1) network components (nodes and/or edges) from an R-robust 2-club guarantees that the remaining nodes and edges form a 2-club.*

These robustness characteristics are attractive due to the following considerations:

1. Error and attack tolerance properties of *R*-robust 2-clubs are similar to those of cliques (the deletion of multiple network components does not violate the connectivity of a cluster); however, the size of *R*-robust 2-clubs is usually larger than the size of cliques in real-world networks (this is especially true for power-law networks, which is illustrated by Figs. 2 and 3(a)– (c));
2. The connectivity pattern that is preserved after the deletion of (*R* − 1) network components is a 2-club (rather than just a regular connected component), which ensures that all nodes are connected by a short path even after multiple network component failures;
3. The number of network components that can be deleted without violating the 2-club structure of the considered *R*-robust 2-club is determined solely by the user-defined parameter *R* and does not depend on any other parameters, such as the size of this *R*-robust 2-club or the size of the original network.

5. Computational experiments

In this section, we present the summary of computational experiments conducted using the formulations developed above, as well as demonstrate that the developed maximum *k*-club formulation is tight, especially as *k* increases.

We have performed computational experiments with different types of random graphs. First, we have considered power-law and uniform random graph instances and investigated how the size of the maximum *k*-clubs and *R*-robust *k*-clubs depends on the parameters of these random graphs. In addition, we have performed the experiments that demonstrate that the computational tractability of the maximum *k*-club problem for any specific graph instance *does not significantly depend on k* (assuming that $k \ll n$), since the values of $k > 2$ do not substantially increase the size of the problem ($O(kn^2)$). This has been demonstrated using random graph instances consistent with those used in Bourjolly et al. (2002). As mentioned above, for the previously developed

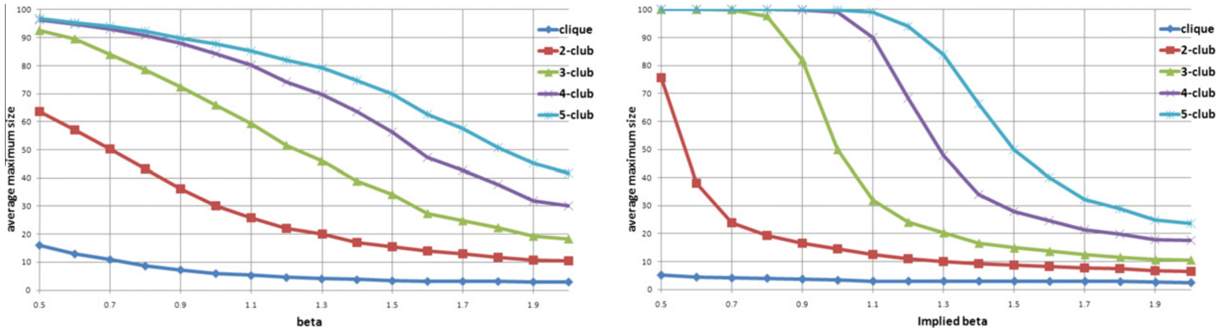

Fig. 1. Average maximum k-club size for different values of k in a power-law (left) and uniform (right) random graphs with the same number of nodes ($n = 100$) and the same edge densities for equal beta and implied beta.

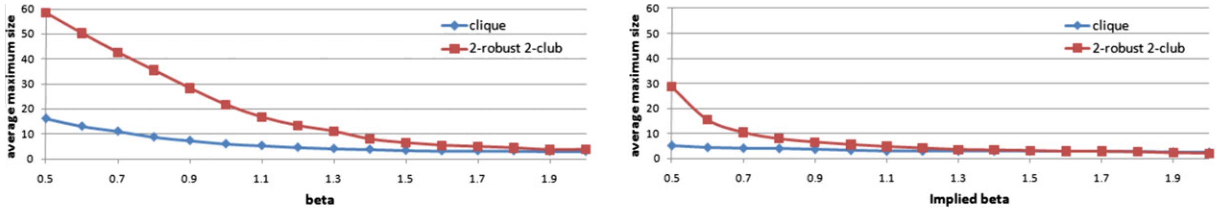

Fig. 2. Comparison of the maximum clique and maximum 2-robust 2-club size in a power-law (left) and uniform (right) random graphs with the same number of nodes ($n = 100$) and the same edge densities for equal beta and implied beta.

formulation of the maximum k-club problem ($O(n^{k+1})$ entities in the general case), the size of the problem would increase drastically for $k > 2$, and it is challenging and impractical to implement and use that formulation for any problem instances other than those with $k = 2$ (in the special case of $k = 2$, both formulation approaches produce the same set of constraints). FICO Xpress-IVE Version 1.21.02 solver was used for finding exact solutions for the considered problems on random graph instances presented in the plots and the table below.

The results of a series of computational experiments on uniform and power-law random graph instances are presented in Figs. 1 and 2. These plots present the average size of the maximum k-clubs and 2-robust 2-clubs compared to the maximum clique size in the corresponding graphs. Note that power-law and uniform random graph instances were generated so that they would have the same edge density. Recall that a uniform random graph $G(n,p)$ has n nodes, where each pair of nodes is connected by an edge independently with the probability p, whereas in a power-law graph the probability that a node has a degree k is proportional to $k^{-\beta}$. Clearly, the parameters p and β determine the edge density of the corresponding uniform and power-law random graph instances; therefore, assuming that the graphs have the same number of nodes, the parameters β and p can be chosen to ensure that the power-law and uniform random graph have the same edge density. That is, for any value of p, there exists a value of β (referred to as the *implied* β) that would produce a power-law graph with the same edge density.

A brief formal description of how an *implied* β is calculated is presented below. Given parameters α and β, a power-law graph has y nodes with degree of x, where y and x satisfy

$$y = \frac{e^{\alpha}}{x^{\beta}}.$$

Obviously, the maximum degree of that graph cannot be greater than $e^{\alpha/\beta}$. Then, the number of nodes (n) in that graph can be computed as follows

$$n = \sum_{x=1}^{e^{\alpha/\beta}} \frac{e^{\alpha}}{x^{\beta}}.$$

Thus, knowing the values of n and β, we can calculate α. The number of edges can be calculated as

$$|E| = 1/2 \sum_{x=1}^{e^{\alpha/\beta}} x \frac{e^{\alpha}}{x^{\beta}}.$$

Note, that in the uniform random graph, the expected number of edges is

$$|E| = p\frac{n(n-1)}{2}.$$

Therefore, using the last two formulas, if p is fixed, we can compute β to ensure that the power-law and uniform random graphs have the same edge density, and vice versa. More details on power-law random graphs can be found in Chung et al. (2004, 2001).

The figures demonstrate that k-clubs and 2-robust 2-clubs are generally significantly larger than cliques in both uniform and power-law random graphs, which motivates the practical significance of these types of robust network clusters. Also, an interesting observation is that the size of the maximum k-clubs decays at a higher rate for uniform random graphs (the rate appears to be exponential for uniform random graphs, and power-law with a thicker tail for power-law random graphs). Fig. 3 shows the maximum clique and the maximum 2-robust and 3-robust 2-clubs in the same power-law graph, and it can be seen that 2-robust and 3-robust 2-clubs are substantially larger than the maximum clique.

In the next set of computational experiments we demonstrate the advantages of the proposed model for solving the maximum k-club problem for different values of k. As the proposed formulation shows, the number of entities grows linearly as k increases. Therefore, the computational time of solving the maximum k-club problem should not vary drastically when $k > 2$ is still reasonably small (as is often the case in practical settings). We considered $k = 2, 3, 4, 5, 6, 7$ and tested the proposed formulation on randomly generated 100-, 200-, and 300-node graphs controlled by two density parameters $0 \leqslant a \leqslant b \leqslant 1$. We used the graph generation procedure from Gendreau et al. (1993), which was also used by Bourjolly et al. (2002), in order to generate graphs with greater

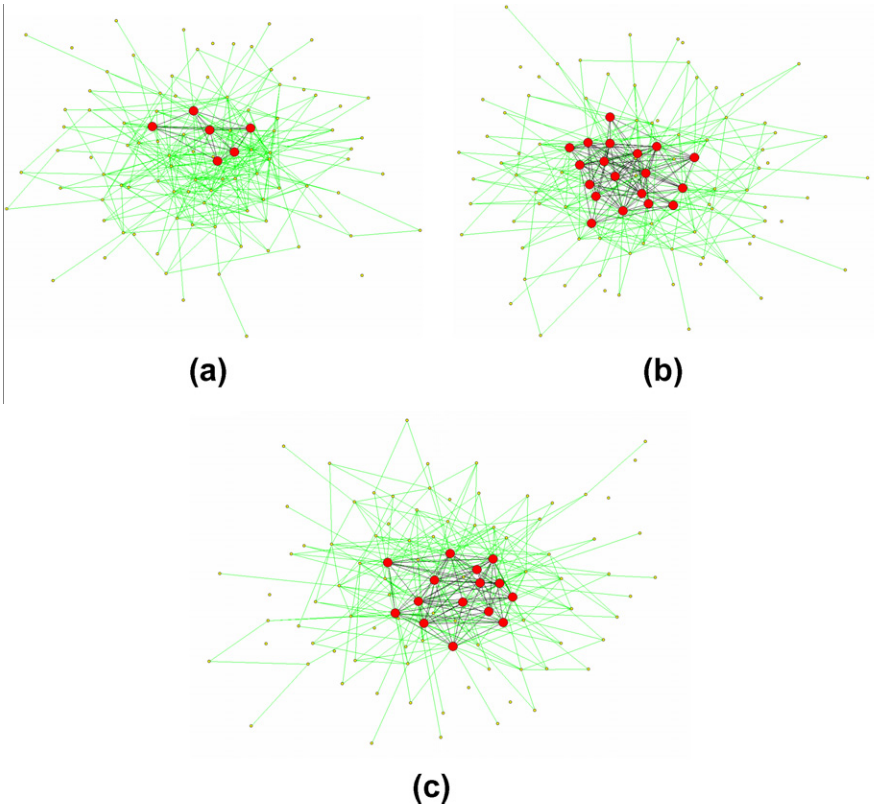

Fig. 3. Graphical representation of (a) maximum clique (6 nodes), (b) 2-robust 2-club (19 nodes), and (c) 3-robust 2-club (15 nodes) in the same power-law network with 100 nodes and $\beta = 1.2$. The maximum 2-robust and 3-robust 2-clubs are substantially larger than the maximum clique, while they still have good robustness characteristics. The figures were obtained using Pajek Version 1.26 (2010).

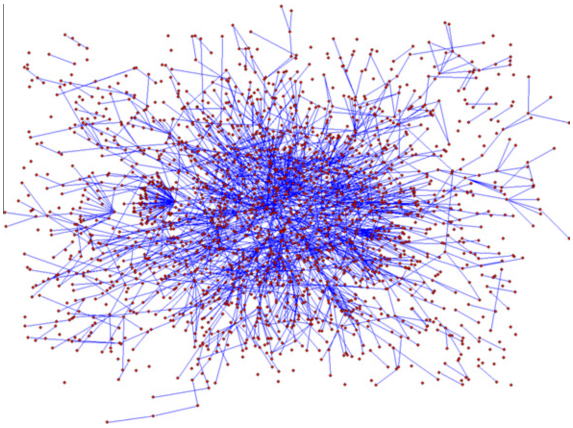

Fig. 4. General view of the protein–protein interaction network of the yeast S. Cerevisiae.

variance in node degrees compared to classical uniform random graphs. The generation procedure is as follows. First, for every value of $i = 1,\ldots,n$, the numbers p_i are randomly chosen from the interval $[a,b]$. Then, the probability $p_{ij} = (p_i + p_j)/2$ is defined and an edge (i,j) between nodes i and j in the graph is generated with probability p_{ij}. Therefore, the larger the difference $(b - a)$, the greater the degree variance in the generated graph. The average edge density of the generated graph is $\hat{p} = (a + b)/2$. These experiments were performed on a machine with Intel Core i7 CPU X 940 2.13 GHz processor, 8 GB RAM, running 64-bit Windows 7 Professional operating system.

Table 1 summarizes the computational results. For every pair $(n, \hat{p})$, we generated 10 graph instances (90 graph instances and 540 problem instances total) and reported the average maximum k-club size, the average CPU time for solving the proposed linear 0–1 formulation, as well as the average tightness for each group of problem instances (i.e., the average relative gap between the exact 0–1 and the corresponding LP relaxation objective values). The edge density of the considered graphs was chosen to make sure that the optimal solutions for all $k = 2,\ldots,7$ are non-trivial and do not coincide with the entire graph (as it was also mentioned in Section 3.4).

Next, we considered a real-world network instance that represents protein–protein interactions of the yeast S. Cerevisiae. This is a sparse power-law network with approximately 2,000 nodes and edges (see Fig. 4). Note that this network has been considered in (Balasundaram et al., 2005), where the maximum 2-club and the maximum 3-club (which was obtained using the maximum 3-clique problem formulation) were identified in this network. In this study, we conducted computational experiments for this network and found the maximum 2, 3, and 4 – clubs, as well as the maximum 2-robust 2-club using the proposed formulations. It is worth mentioning that to our knowledge, this study is the first one that produced the exact optimal solution of the maximum 4-club problem in the S. Cerevisiae network.

For computational purposes, we used the following preprocessing procedure to decrease the size of the considered optimization problems for the S. Cerevisiae network.

1. *Identify connected components.* Since any k-club is a connected cluster, then any two nodes in the graph which are disconnected cannot belong to any k-club. Thus, any k-club is a subset of a connected cluster. The connected clusters can be identified in a polynomial time.

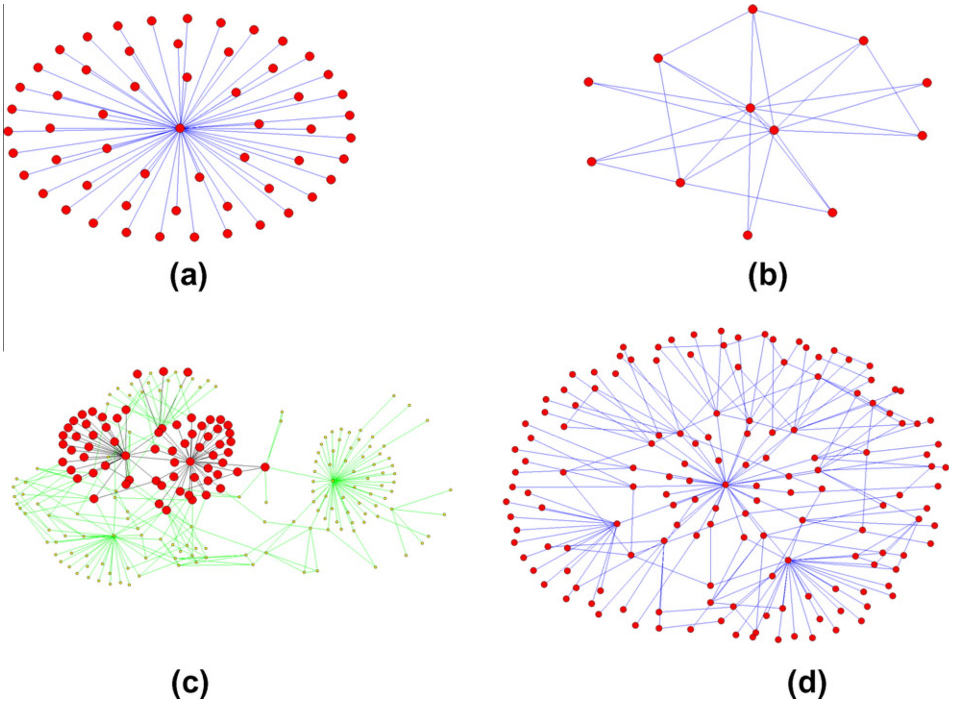

Fig. 5. Graphical representation of maximum 2, 3, and 4 – clubs ((a), (c), and (d)), as well as the maximum 2-robust 2-club (b), in the S. Cerevisiae network. The figures were obtained using Pajek Version 1.26 (2010).

2. *Ignore any node that cannot belong to a large k-club.* This network is a typical example of a sparse power-law graph. In these graphs, there are few nodes with large degrees and many nodes with very low degrees. One can find some large k-club (assume that its size is N_k) which has a node with the largest degree in this network and use it as a lower bound on the maximum k-club. Therefore, any node which has less than N_k k-distant neighbors (i.e., nodes that are connected with the considered node through a path of at most k edges) cannot belong to the maximum k-club. We can ignore these nodes and reduce the size of the optimization problem.
3. *Ignore any node with a degree less than R in the maximum R-robust 2-club problem.* Since any node in an R-robust 2-club should have a degree of at least R, then we can ignore any node which has a degree less than R and also reduce the size of the optimization problem.

The results of these computations are presented in Fig. 5. It can be easily observed from this figure that the maximum 2, 3, and 4-clubs are very vulnerable to node deletions (e.g., the deletion of only one node would violate the cluster connectivity); however, the maximum 2-robust 2-club would remain a 2-club if any one node or edge is deleted.

6. Conclusion

In this paper, we have developed a new linear 0–1 programming formulation with $O(kn^2)$ entities that allows one to find exact solutions of the maximum k-club problem in the general case of $k > 2$ substantially more efficiently than the previously known approaches. To the best of our knowledge, no previous studies on exact algorithms for this problem have reported any computational experiments for problem instances with $k \geqslant 5$, or any computational experiments with 200- and 300-node graphs for $k > 2$. Besides the fact that the proposed formulation is compact, the conducted computational study on a total of 540 problem instances suggests that this formulation is at least as good in terms of tightness as the formulation described in Bourjolly et al. (2002) and Balasundaram et al. (2005).

In addition, we have introduced the new concept of an R-robust k-club and developed the corresponding compact formulations for certain special cases of the maximum R-robust k-club problem. Moreover, we have shown that in the special case of R-robust 2-clubs, one can guarantee and theoretically justify important robustness characteristics of these network clusters, in particular, their error and attack tolerance.

Directions of potential further research include more detailed theoretical analysis of tightness of the two formulations, as well as developing formulations and solution algorithms for the general case of the maximum R-robust k-club problem with internally node-disjoint paths.

Acknowledgements

V. Boginski's research is partially supported by DTRA and AFOSR grants. The authors thank the referees for their valuable comments that helped to improve the paper.

References

Abello, J., Resende, M.G.C., Sudarsky, S., 2002. Massive quasi-clique detection. Lecture Notes in Computer Science, LATIN 2002: Theoretical Informatics. Springer, pp. 598-612.

Adams, W.P., Forrester, R.J., 2007. Linear forms of nonlinear expressions: New insights on old ideas. Operations Research Letters 35, 510–518.

Albert, R., Jeong, H., Barabási, A-L., 2000. Error and attack tolerance of complex networks. Nature 406, 378–382.

Balasundaram, B., Butenko, S., Trukhanov, S., 2005. Novel approaches for analyzing biological networks. Journal of Combinatorial Optimization 10, 23–39.

Balasundaram, B., Butenko, S., Hicks, I., 2011. Clique relaxations in social network analysis: The maximum k-plex problem. Operations Research 59, 133–142.

Bomze, I.M., Budinich, M., Pardalos, P.M., Pelillo, M., 1999. The maximum clique problem. In: Du, D.-Z., Pardalos, P.M. (Eds.), Handbook of Combinatorial Optimization. Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 1–74.

Bourjolly, J.-M., Laporte, G., Pesant, G., 2000. Heuristics for finding k-clubs in an undirected graph. Computers and Operations Research 27, 559–569.

Bourjolly, J.-M., Laporte, G., Pesant, G., 2002. An exact algorithm for the maximum *k*-club problem in an undirected graph. European Journal of Operational Research 138, 21–28.

Chaovalitwongse, W., Pardalos, P.M., Prokopyev, O.A., 2004. A new linearization technique for multi-quadratic 0-1 programming problems. Operations Research Letters 32, 517–522.

Chung, F., Lu, L., Vu, V., 2004. The spectra of random graphs with given expected degrees. Internet Mathematics 1, 257–275.

Chung, F., Aiello, B., Lu, L., 2001. A random graph model for power law graphs. Experimental Mathamatics 10, 53–66.

Gendreau, M., Soriano, P., Salvail, L., 1993. Solving the maximum clique problem using a tabu search approach. Annals of Operations Research 41, 385–403.

Glover, F., Woolsey, E., 1974. Converting the 01 polynomial programming problem to a 01 linear program. Operations Research 22, 180–182.

Luce, R.D., 1950. Connectivity and generalized cliques in sociometric group structure. Psychometrika 15, 169–190.

Mokken, R.J., 1979. Cliques, clubs and clans. Quality and Quantity 13, 161–173.

Pajek Version 1.26, 2010 (http://vlado.fmf.uni-lj.si/pub/networks/pajek/).

Pattillo, J., Veremyev, A., Butenko, S., Boginski, V., 2011. On the maximum quasiclique problem. Submitted for publication.

Prokopyev, O.A., Meneses, C., Oliveira, C.A.S., Pardalos, P.M., 2005. On multiple-ratio hyperbolic 0–1 programming problems. Pacific Journal of Optimization 1, 327–345.

Seidman, S.B., Foster, B.L., 1978. A graph theoretic generalization of the clique concept. Journal of Mathematical Sociology 6, 139–154.

Wu, T.-H., 1997. A note on a global approach for general 0–1 fractional programming. European Journal of Operational Research 101, 220–223.