Chapter 5
Clique Relaxation Models in Social Network Analysis

Jeffrey Pattillo, Nataly Youssef, and Sergiy Butenko

Abstract Clique relaxation models that were originally introduced in the literature on social network analysis are not only gaining increasing popularity in a wide spectrum of complex network applications, but also keep garnering attention of mathematicians, computer scientists, and operations researchers as a promising avenue for fruitful theoretical investigations. This chapter describes the origins of clique relaxation concepts and provides a brief overview of mathematical programming formulations for the corresponding optimization problems, algorithms proposed to solve these problems, and selected real-life applications of the models of interest.

5.1 Introduction

Social networks represent certain types of social interaction, such as acquaintance, friendship, or collaboration between people or groups of people that are referred to as *actors*. In social networks, vertices (nodes, dots) usually stand for actors, and edges (arcs, links, lines) represent the pairwise relations or interactions between the actors. As an illustration of a social network, an example of a terrorist network is given in Fig. 5.1, which describes the connections between terrorists associated with the September 11, 2001 attack on the World Trade Center. The links were identified by Krebs [49] after the terrorist attack; however, the network is reconstructed

J. Pattillo (✉)
Department of Mathematics, Texas A&M University, College Station, TX 77843, USA
e-mail: jptlo@math.tamu.edu

N. Youssef • S. Butenko
Department of Industrial and Systems Engineering, Texas A&M University,
College Station, TX 77843-3131, USA
e-mail: nyoussef@tamu.edu; butenko@tamu.edu

M.T. Thai and P.M. Pardalos (eds.), *Handbook of Optimization in Complex Networks: Communication and Social Networks*, Springer Optimization and Its Applications 58,
DOI 10.1007/978-1-4614-0857-4_5,

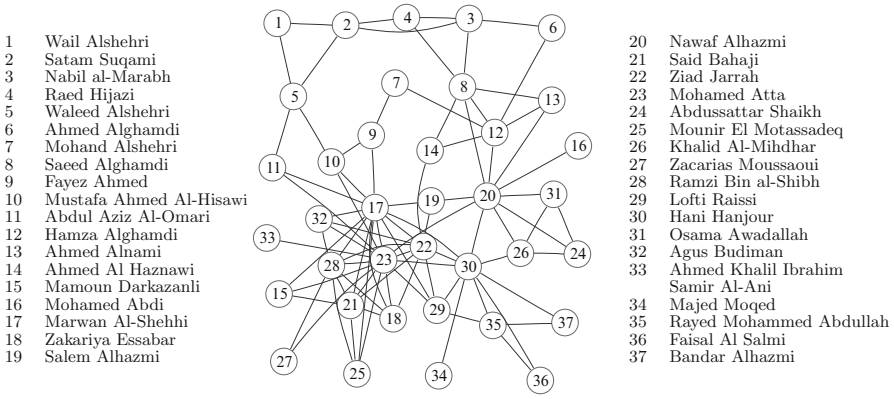

Fig. 5.1 The network representation of the data describing connections between terrorists associated with September 11 events

based on the information that was publicly available *before* September 11. Analysis of such networks, even though performed post-factum, may still be useful for understanding the structure of terrorist organizations and detecting similar structures to prevent potential terrorist attacks in the future.

One of the central concepts in *social network analysis* is the notion of a *cohesive subgroup*, which is a "tightly knit" subgroup of actors in a social network. During the last several decades, social network analysis methods in general, and social cohesion concepts in particular, have been employed in various branches of sociology in the contexts of crime detection/prevention and terrorist network analysis [12, 27, 31, 56, 68, 69], studies on the sociology of political systems and historical revolutionary movements [41, 65], epidemiology of sexually transmitted diseases [67], organizational management [32], and the sociology of worker solidarity [42].

Study of social cohesion can be traced back to one of the founding fathers of sociology, Emile Durkheim (1858–1917), who wrote [35]:

> The totality of beliefs and sentiments common to the average members of a society forms a determinate system with a life of its own. It can be termed the collective or common consciousness.

Durkheim found the notion of cohesiveness difficult to define rigorously,

> ...social solidarity is a wholly moral phenomenon which by itself is not amenable to exact observation and especially not to measurement.

However, a rigorous mathematical definition became possible with the introduction of sociometric analysis and graph theoretic methods into sociology that have led to the association of the notion of social cohesion with certain graph theoretic concepts. In particular, Luce and Perry [53] used *complete subgraphs* to model *social cliques*, which are defined as "an exclusive circle of people with a common

purpose" [58]. It should be noted that the term "clique" was used in Hawthorne and Warner studies in the 1930s (see, e.g., [66, 70]), however, it referred to "socially perceived subgroups" and was not strictly defined [70].

While the notion of a clique embodies a "perfect" cohesive group, in which every two entities are connected to each other, this definition is overly conservative in many practical scenarios. Indeed,

1. One may not need to require every possible link to exist between elements of a cohesive subgroup.
2. The social network of interest may be built based on empirical data, which are prone to errors, so, even if a completely connected cohesive subgroup is sought for, it may be impossible to detect due to erroneous data.

To overcome this impracticality of the clique model, other graph-theoretic formalizations of the cohesive subgroup concept have been proposed in the literature. Not surprisingly, all these alternative definitions can be viewed as clique generalizations, each of which relaxes one of the elementary clique properties, such as familiarity, reachability, or robustness [4, 52, 59, 72]. Hence, we use the term "clique relaxations" in reference to such models. Since their introduction in the social network literature in the 1970s, the clique relaxation models received very little attention from researchers in graph theory, computer science, and operations research during the decades that followed. At the same time, methodology-wise, the social network literature dealing with these concepts was largely limited to giving their formal definition within a given context and providing illustrative examples on graphs with several vertices [83].

Recent progress in Internet and telecommunication technologies makes it possible to collect tremendous amounts of social interaction data dynamically and at virtually any level of detail, providing unprecedented opportunities to researchers interested in areas related to social network analysis. The availability of massive amounts of data that needs to be analyzed spurred the interest in developing effective algorithms capable of handling problems of practical scale. The need for advanced computational tools urged the recent progress in developing theoretical foundations for the models of interest and effective algorithms utilizing the theoretical achievements. The objective of this chapter is to provide a brief survey of results concerning optimization problems seeking to find maximum size clique relaxation structures in networks. More specifically, in this chapter we are primarily interested in mathematical programming formulations of the corresponding optimization problems and algorithms proposed for solving these problems. While in social network applications one may also be interested in detecting cohesive subgroups of sizes smaller than maximum, computing the largest cohesive subgroup is of special interest, since its size provides a global measure of cohesiveness of the corresponding network. We will also discuss selected applications of clique relaxation models outside of social network analysis and highlight some of the promising directions for future research in this active area.

The remainder of this chapter is organized as follows. Section 5.2 provides formal definitions of the clique relaxation concepts introduced in the context of

social network analysis and discusses some of their basic properties. Mathematical programming formulations of the corresponding optimization problems, that aim to find clique relaxation structures of largest possible size in the given network, are outlined in Sect. 5.3. Section 5.4 surveys existing algorithms for solving the optimization problems of interest either exactly or approximately and discusses successful examples of application of some of these algorithms in practice. Extensions of the discussed methodologies to applications outside of the sociological realm are the focus of Sect. 5.5. Finally, Sect. 5.6 concludes the chapter.

5.2 Definitions and Properties of Clique Relaxation Models

Throughout this chapter, we consider a simple undirected graph $G = (V,E)$, where $V = \{1,...,n\}$ and $E \subseteq V \times V$, respectively, denote the sets of vertices and edges in G, with $|V| = n$ and $|E| = m$. G is said to be *complete* if for every $u,v \in V$ such that $u \neq v$, $(u,v) \in E$; in other words, $G = (V,E)$ is *complete* if $E = V \times V$ with $|E| = \binom{n}{2}$. Given a subset of vertices $S \subseteq V$, $G[S] = (S, E \cap (S \times S))$ denotes the *subgraph induced by* S, obtained by deleting all vertices in $V \setminus S$ and their corresponding incident edges from G.

A clique C is a subset of V such that the subgraph $G[C]$ induced by C on G is complete. A clique is called *maximal* if it is not contained in a larger clique, and it is called *maximum* if there is no larger clique in G. The size of the maximum clique in G is referred to as the clique number and is denoted by $\omega(G)$.

The notion of clique defined above embodies ideal properties of *reachability*, *familiarity*, *density*, and *robustness* discussed below in this chapter. Such structural properties are indeed very useful in modeling group cohesiveness within social networks. The idealized properties of cliques are, however, overly restrictive in practical applications. This observation has called for the development of clique relaxations, aiming at relaxing particular properties of cliques. We next introduce the definitions of the major graph properties that are needed in order to characterize the clique relaxations of interest. A path between two vertices $u,v \in V$ of length k is a sequence of distinct vertices $u = v_0, v_1, ..., v_k = v$, such that $(v_i, v_{i+1}) \in E, 0 \leq i \leq k-1$. Two paths are called *independent* if they only intersect at their ends. The length of the shortest path between two vertices $u,v \in V$ in G is represented by $d_G(u,v)$, referred to as the *distance* between u and v. The largest of the pairwise distances between vertices define the *diameter* of the graph, i.e., $\text{diam}(G) = \max_{u,v \in V} d_G(u,v)$. G is said to be *connected* if every two vertices in G are linked by a path in G. For any vertex $v \in V$, the *open neighborhood* $N(v) = \{u \in V | (u,v) \in E)\}$ defines the set of vertices adjacent to v in G, whereas $N[v] = \{v\} \cup N(v)$ denotes the *closed neighborhood* of v. The *degree* of a vertex $v \in V$, given by $\deg_G(v) = |N(v)|$, represents the number of edges emanating from v. Considering a subgraph $G[S]$ induced by $S \subseteq V$, the degree of a vertex $s \in S$ is given by $\deg_{G[S]}(s) = |N(s) \cap S|$. The minimum degree in a graph is denoted by $\delta(G)$. A subset of vertices D is called a *dominating set* if every vertex in the graph is either in D or has at least one neighbor

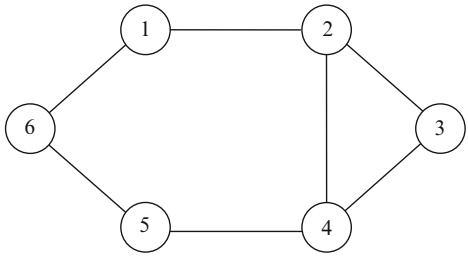

Fig. 5.2 A graph illustrating the difference between 2-cliques and 2-clubs, originally used in [4]

in D. The size of a smallest dominating set, also known as the domination number of G, is denoted by $\gamma(G)$. The *edge density* of G is the ratio of the number of edges to the total number of possible edges, i.e., $m/\binom{n}{2}$.

The concepts of k-cliques and k-clubs were among the first clique relaxations to be studied in the literature and were first introduced in [72] and [71]. They both relax the property of *reachability* desirable for a cohesive subgroup, that is, each member of a cohesive subgroup should be able to easily reach any other member of the subgroup. In graph-theoretic terms, this property can be expressed using the notions of distance and diameter. While pairwise distances between members of the clique is equal to one, k-cliques ensure that any two vertices within the subgraph are at most distance k from each other in the original graph. On the other hand, k-clubs require the diameter of the induced subgraph to be at most k. Next, we formally define these concepts.

Definition 5.1. A k-clique S is a subset of V such that, for all vertices $u, v \in S$, $d_G(u,v) \leq k$. The size of the largest k-clique is called the k-clique number and is denoted by $\tilde{\omega}_k(G)$.

Definition 5.2. A k-club S is a subset of V such that the induced subgraph $G[C]$ has a diameter of at most k. The size of the largest k-club is called the k-club number and is denoted by $\overline{\omega}_k(G)$.

Note that any k-club is a k-clique but the converse is not true. For example, the graph in Fig. 5.2 contains a 2-clique $C = \{1,2,3,4,5\}$, which is not a 2-club, since the distance between vertices 1 and 5 in the subgraph induced by C is 3. Based on this observation, Alba [4] concluded that the concept of k-clique lacks the "tightness" essential to applications in social networks, which motivated him to introduce the concept of a "sociometric clique", later renamed to "k-clan" by Mokken [59]. According to their definition, a k-clique C is called an *k-clan* if the diameter of the induced subgraph $G(C)$ is no more than k. A considerable drawback in the k-clan definition is that for some graphs a k-clan may not exist [10]. Indeed, Fig. 5.3 shows a graph with two 2-cliques $\{1,2,3,4,5,6,7\}$ and $\{1,2,3,5,6,7,8\}$, neither of which is a 2-clan.

Familiarity is another important property one wants to have in a cohesive subgroup; ideally, every member of the group should be familiar with every other member of the group, which is the case for cliques. Known familiarity-based models either relax the minimum number of neighbors or the maximum number of non-neighbors within the group. The k-core concept, first introduced in

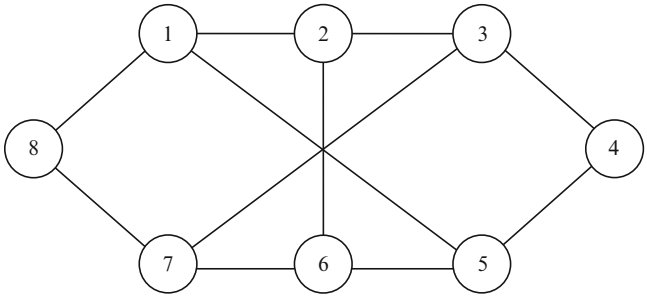

Fig. 5.3 A graph with no 2-clans proposed in [10]

[71], imposes a lower bound on the minimum degree within the subgraph, ensuring that each vertex in the group is connected to at least k other vertices in the group. The k-plex model introduced in [72], on the other hand, requires any subset of k vertices to dominate to entire group, hence restricting the number of non-neighbors per vertex to be at most $k-1$. The formal definitions of these familiarity-based relaxations are given next.

Definition 5.3. A k-core S is a subset of V such that, for all vertices $u, v \in S$, $\deg_{G[S]}(v) = |N(v) \cap S| \geq k$. The size of the k-core in the graph G is denoted by $\omega'_k(G)$.

Definition 5.4. A k-plex S is the maximum subset of V such that any set of k vertices in S dominates the group. Alternatively, a k-plex S is a subset of V such that, for all vertices $u, v \in S$, $\deg_{G[S]}(v) = |N(v) \cap S| \geq |S| - k$. The size of the largest k-plex is denoted by $\omega_k(G)$.

High *density* of connections is yet another defining characteristic of a cohesive subgroup that is perfectly reflected in the notion of clique, which can be alternatively defined as a subset of vertices inducing a subgraph with edge density of 1. This rather strict requirement on edge density could also be relaxed. Namely, instead of opting for cliques with density one, γ-quasi-cliques ensure a subgraph with a density *at least* γ, where $0 \leq \gamma \leq 1$.

Definition 5.5. A γ-quasi-clique or, simply, a γ-clique is a subset S of V such that the induced subgraph $G[S]$ has the edge density $|E \cap (S \times S)| / \binom{|S|}{2}$ of at least γ. The size of the largest γ-quasi-clique is denoted by $\omega_\gamma(G)$.

Finally, *robustness* of a group may be measured in terms of the number of vertices that need to be deleted to completely disconnect the group. Since any vertex within a clique is directly linked to any other vertex of the clique, disconnecting a clique would require removing all its vertices. A model known as the k-vertex connected subgraph relaxes the connectivity property, whereby removal of at least k vertices destroys connectivity. By Menger's theorem [33], a k-vertex connected group can be also defined as a subgraph such that any two of its vertices can be joined by k independent paths.

Definition 5.6. A k-vertex connected subgraph $G[S]$ is defined by a subset S of V such that the induced subgraph $G[S \setminus R]$ is connected for all subsets $R \subset S$, with $|R| < k$. The size of the largest k-connected subgraph is denoted by $\omega_{kc}(G)$.

The corresponding concept in social networks literature is the so-called *structural cohesion*, which is defined as the minimum number of members whose removal from a group would disconnect the group [60].

While, by definition, the above mentioned clique relaxations emphasize a single clique structural aspect, other models have been created ensuring different combinations of properties. Examples include the (λ, γ)-quasi-clique relaxing both density and degree requirements and the r-robust k-club additionally requiring r-connectivity in a k-club. For further readings, consult [23, 81]. Surveys [18] and [63] provide further information and references on the maximum clique and related problems.

5.3 Mathematical Programming Formulations

All the optimization problems of interest to this chapter can be formulated as (mixed) integer programs (MIP). Below in this section we provide some of the known formulations for all problems except for the maximum k-core and maximum k-vertex connected subgraph. No k-core formulation is given due to the fact that, as we will discuss in Sect. 5.4, the maximum k-core problem can be solved to optimality by a simple greedy algorithm that runs in polynomial time, whereas all other problems considered in this chapter are NP-hard. Thus, even though the maximum k-core problem can be easily formulated as an integer program, this is not the way this problem is typically approached. On the other hand, we are not aware of any publications proposing mathematical programming formulations or effective algorithms for the maximum k-vertex connected subgraph problem. Developing such formulations and algorithms is an interesting and important direction for future research, especially since the corresponding notion of structural cohesion is enjoying rapidly increasing popularity in the literature on social network analysis.

5.3.1 Maximum k-Clique

The maximum k-clique problem consists of finding the k-clique in the graph with largest cardinality $\omega_k(G)$. This problem can be formulated as a binary integer linear problem:

$$\tilde{\omega}_k(G) = \max \sum_{i \in V} x_i \tag{5.1}$$

$$\text{subject to} \quad x_i + x_j \leq 1, \ \forall i, j \in V \ d_G(i,j) > k, \tag{5.2}$$

$$x_i \in \{0,1\}, \ i \in V. \tag{5.3}$$

Each binary variable corresponds to a vertex in the graph and assumes unity if the corresponding vertex is included in the k-clique. Constraint (5.2) ensures that two vertices whose pairwise distance in the original graph G exceeds k are not both considered in the k-clique.

It should be noted that the maximum k-clique problem in graph G can be equivalently formulated as the maximum clique problem in graph G^k representing the k-th power of graph G, which is defined as follows. G^k has the same set of vertices as G, with edges connecting all pairs of vertices that are distance at most k from each other in G. Thus, the numerous mathematical programming formulations for the maximum clique problem available in the literature [18] can be adopted to obtain analogous formulations for the maximum k-clique problem.

5.3.2 Maximum k-Club

The maximum k-club problem looks for the largest k-club in the graph. This problem can be formulated as a binary integer linear problem:

$$\overline{\omega}_k(G) = \max \sum_{i \in V} x_i \tag{5.4}$$

$$\text{subject to} \quad x_i + x_j \leq 1 + \sum_{l : P^l_{i,j} \in \mathbb{P}_{i,j}} y^l_{ij}, \ \forall (i,j) \notin E, \tag{5.5}$$

$$x_p \geq y^l_{ij}, \ \forall p \in V(P^l_{ij}), P^l_{ij} \in \mathbb{P}_{ij}, (i,j) \notin E, \tag{5.6}$$

$$x_i \in \{0,1\}, \ i \in V, \tag{5.7}$$

$$y^l_{ij} \in \{0,1\}, \ \forall P^l_{ij} \in \mathbb{P}_{ij}, (i,j) \notin E. \tag{5.8}$$

$\mathbb{P}_{ij}$ represents an index set of all paths of length at most k, whereas P^l_{ij} denotes the path between i and j indexed at position $l \in \mathbb{P}_{ij}$. $V(P^l_{ij})$ denotes the set of vertices included inn path P^l_{ij}. Constraint (5.5) allows both vertices i and j into the solution if no edge but a path of length at most k exists between them, and only if y^l_{ij} is equal to 1 for at least one path between i and j. Hence, this formulation makes sure to include in the k-club all vertices along any one path of length at most k between any two vertices in the k-club.

More recently, Veremyev and Boginski [81] have developed a more compact formulation for the maximum k-club problem with $O(kn^2)$ constraints. We first present the special case with $k = 2$ and then generalize to any $k \geq 2$. Let $A = [a_{ij}]^n_{i,j=1}$, where $a_{ij} = 1$ if there exists an edge between vertices i and j, and 0 otherwise, denote the adjacency matrix of G. Vertices in a 2-club are either connected directly or through at most one other vertex l, which can be expressed

using the following non-linear constraint:

$$a_{ij} + \sum_{l=1}^{n} a_{il}a_{lj}x_l \geq x_i x_j. \tag{5.9}$$

Simplifying the aforementioned constraint results in the following formulation:

$$\overline{\omega}_k(G) = \max \sum_{i\in 1}^{n} x_i \tag{5.10}$$

$$\text{subject to} \quad a_{ij} + \sum_{l=1}^{n} a_{il}a_{lj}x_l \geq x_i + x_j - 1, \forall i = 1,...,n; j = i+1,...,n, \tag{5.11}$$

$$x_i \in \{0,1\},\ i = 1,...,n. \tag{5.12}$$

The above formulation can be generalized for any $k \geq 2$, by allowing any two vertices in the k-club to be either directly linked or connected through at most $k-1$ vertices within the subgraph:

$$\overline{\omega}_k(G) = \max \sum_{i\in 1}^{n} x_i \tag{5.13}$$

$$\text{subject to} \quad a_{ij} + \sum_{l=1}^{n} a_{il}a_{lj}x_l + \sum_{l=1}^{n}\sum_{m=1}^{n} a_{il}a_{lm}a_{mj}x_l x_m + \cdots + \tag{5.14}$$

$$\sum_{i_1=1}^{n}\sum_{i_2=1}^{n}\cdots\sum_{i_{k-2}=1}^{n}\sum_{i_{k-1}=1}^{n} a_{ii_1}a_{i_1i_2}\ldots a_{i_{k-2}i_{k-1}}a_{i_{k-1}j}x_{i_1}\ldots x_{i_{k-1}} \tag{5.15}$$

$$\geq x_i + x_j - 1, \forall i = 1,...,n; j = i+1,...,n, \tag{5.16}$$

$$x_i \in \{0,1\},\ i = 1,...,n. \tag{5.17}$$

5.3.3 Maximum k-Plex

The maximum k-plex problem finds the k-plex in the graph with maximum cardinality. This problem can be formulated as a binary integer linear problem, where $\deg_{\overline{G}}(i) = |V \backslash N[i]|$ denotes the degree of vertex i in the complement graph $\overline{G} = (V, \overline{E})$ [9]:

$$\omega_k(G) = \max \sum_{i\in V} x_i \tag{5.18}$$

$$\text{subject to} \quad \sum_{j\in V\backslash N[i]} x_j \leq (k-1)x_i + \deg_{\overline{G}}(i)(1-x_i),\ \forall i \in V, \tag{5.19}$$

$$x_i \in \{0,1\},\ i \in V. \tag{5.20}$$

Binary variables indicate whether or not a vertex is included in the maximum k-plex. Constraint (5.19) expresses the fact that if vertex i is included in the k-plex, then it has at most $k-1$ non-neighbors, otherwise the constraint becomes redundant.

5.3.4 Maximum Quasi-clique

The maximum γ-quasi-clique problem can be formulated as the following mixed-integer problem with linear objective and a single quadratic constraint [64]:

$$\omega_\gamma(G) = \max \sum_{i=1}^{n} x_i \tag{5.21}$$

$$\text{subject to} \quad \sum_{i=1}^{n} \sum_{j=i+1}^{n} a_{ij} x_i x_j \geq \gamma \sum_{i=1}^{n} \sum_{j=i+1}^{n} x_i x_j, \tag{5.22}$$

$$x_i \in \{0,1\},\ i = 1,\ldots,n. \tag{5.23}$$

Two linearizations of this formulation are proposed in [64]. The first, standard, linearization is based on defining a new variable $x_{ij} = x_i x_j$ and using an equivalent (for binary variables) representation of this quadratic equation in terms of three linear inequalities. This results in the following mixed integer linear programming formulation with $n(n-1)/2$ variables and $\frac{3}{2}n(n-1)+1$ constraints:

$$\omega_\gamma(G) = \max \sum_{i=1}^{n} x_i, \tag{5.24}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \sum_{j=i+1}^{n} (\gamma - a_{ij}) x_{ij} \leq 0, \tag{5.25}$$

$$x_{ij} \leq x_i,\ x_{ij} \leq x_j,\ x_{ij} \geq x_i + x_j - 1,\ j > i = 1,\ldots,n \tag{5.26}$$

$$x_{ij} \geq 0,\ x_i \in \{0,1\},\ j > i = 1,\ldots,n. \tag{5.27}$$

The second linearization is based on rewriting the single constraint of formulation (5.21)–(5.23) in the form

$$\sum_{i=1}^{n} x_i \left(\gamma x_i + \sum_{j=1}^{n} (a_{ij} - \gamma) x_j \right) \geq 0. \tag{5.28}$$

and introducing a new variable y_i for $i = 1,\ldots,n$ as follows:

$$y_i = x_i \left(\gamma x_i + \sum_{j=1}^{n} (a_{ij} - \gamma) x_j \right). \tag{5.29}$$

Again, replacing this quadratic equality constraint with several linear inequality constraints, we obtain the following MIP with $2n$ variables, n of which are binary and n continuous, and $4n+1$ constraints:

$$\omega_\gamma(G) = \max \sum_{i=1}^{n} x_i \tag{5.30}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} y_i \geq 0, \tag{5.31}$$

$$y_i \leq u_i x_i, \;\; y_i \geq l_i x_i, \;\; i = 1, \ldots, n, \tag{5.32}$$

$$y_i \geq \gamma x_i + \sum_{j=1}^{n} (a_{ij} - \gamma) x_j - u_i (1 - x_i), \;\; i = 1, \ldots, n, \tag{5.33}$$

$$y_i \leq \gamma x_i + \sum_{j=1}^{n} (a_{ij} - \gamma) x_j - l_i (1 - x_i), \;\; i = 1, \ldots, n, \tag{5.34}$$

$$x_i \in \{0, 1\}; \;\; y_i \in \mathbb{R}, \; i = 1, \ldots, n. \tag{5.35}$$

5.4 Algorithms for Detection of Clique Relaxation Structures

5.4.1 k-Clique

Balasundram et al. [10] have shown that the maximum k-clique problem is NP-hard for any positive integer k. A natural way of attacking this problem is by reducing it to the maximum clique problem in the k-th power of the graph, G^k. The maximum clique problem has been well studied in the literature [18], and all the methods available for this problem can be easily adopted for solving the maximum k-clique problem. Therefore, below we mention some of the numerous algorithms proposed for solving the clique problem. It should be noted that G^k may have a much higher edge density than G, and most of the known algorithms for the maximum clique problem tend to perform better on sparse graphs. Thus, the maximum k-clique problem appears to be more challenging than the maximum clique problem.

Some of the most notable exact combinatorial algorithms for the maximum clique problem have been proposed by Bron and Kerbosch [22], Balas and Yu [8], Applegate and Johnson [5], Carraghan and Pardalos [25], Babel [6], Balas and Xue [7], Wood [84], Sewell [73], Östergård [62], and Tomita and Kameda [78]. Due to its simplicity and effectiveness, especially on sparse graphs, implementation of Carraghan–Pardalos algorithm developed by Applegate and Johnson [5] was used as a benchmark exact algorithm for the maximum clique problem in the Second DIMACS Implementation Challenge [44]. Östergård [62] proposed an enhanced version of the Carraghan–Pardalos algorithm, which uses approximate coloring for

the upper bound. The resulting algorithm implemented in software called *cliquer* is considered one of the fastest exact approaches for solving the maximum clique problem. More recently, Tomita and Kameda [78] claim that for randomly generated test instances and DIMACS benchmark instances [34] the algorithm they propose outperforms all other competitive algorithms published literature at the time of this writing, with Östergård's algorithm being the close second.

For a broad overview of heuristics used for the maximum clique problem, see [63]. Greedy heuristics have been developed based on either a sequential addition of vertices until the clique is built or a removal of vertices from a larger graph. They typically use local information, such as vertex degree, to order the vertices in the greedy scheme [45, 48, 79]. Simulated annealing [1, 43] and tabu search [11, 40, 74, 75] have proven very successful, as have hybrid implementations combining simulated annealing and greedy heuristics [26], [46]. These hybrid methods outperform most greedy heuristics on random graphs with up to 1,000 vertices both in time and quality of solution.

Several algorithms for enumerating all maximal cliques in a graph are also available in the literature [22, 29, 47, 51, 55]. For a comprehensive survey of the maximum clique problem formulations, exact and heuristic algorithms as of 1999, the reader is referred to [18].

5.4.2 k-Club

The maximum k-club problem defined on an undirected graph has been proven NP-hard as a result of a polynomial reduction from the clique problem [20]. Moreover, it has been shown in [10] that the problem remains NP-hard even when restricted to graphs of diameter $k+1$. This has motivated several attempts at constructing efficient heuristics. For instance, Bourjolly et al. [19] have proposed three greedy heuristics, DROP, k-CLIQUE & DROP and CONSTELLATION. The main idea behind DROP consists of sequentially deleting nodes from the original graph until obtaining a k-club. At each iteration, the heuristic computes the shortest path lengths between all pairs of nodes. A single node is then ruled out of the graph which has the largest number of vertices with distance from it is larger than k. This process is repeated until all pairwise shortest path lengths in the remaining graph are k or less, hence, defining a k-club. With an overall time complexity of $O(|V|^3|E|)$, DROP yields near-optimal solutions for high-density graphs. A better performance for lower density graphs for $k = 3$ or 4 is obtained by applying DROP to the largest k-clique in the original graph. The obtained algorithm is referred to as k-CLIQUE & DROP and has exponential complexity due to the initial step of finding the largest k-clique in G. CONSTELLATION, on the other hand, stems from the main idea that a star graph forms a 2-club. The heuristic first finds the largest star subgraph in G centered around the vertex with maximum degree in the graph, resulting in a $t+1$-club, where $t = 1$. Then, at iteration t, it identifies an external vertex with the largest

number of neighbors not yet included in the previously obtained t-club. Appending these vertices to the subgraph results in a $t+1$-club. The algorithm is then repeated until a k-club is reached. CONSTELLATION performs well for low density graphs when $k=2$, with an overall complexity of $O(k(|V|+|E|))$.

While these heuristics find good solutions efficiently for relatively large graphs, exact algorithms have also been developed within the framework of Branch-and-Bound (B&B), attempting to solve the maximum k-club problem on rather smaller graphs. Bourjolly et al. [20] proposed using a single iteration of the DROP heuristic to guide the branching step with upper bounds relying on solutions to the maximum stable set problem for an auxiliary graph. The auxiliary graph in this procedure consists of all vertices in G with edges between vertices if their distance in G exceeds k. While DROP can be performed efficiently, the bounding step at each node of the B&B tree may be expensive, since the maximum stable set problem is known to be NP-complete [39, 44]. This algorithm solved instances with no more than 200 vertices to optimality and reported more computationally efficient solutions for denser graphs. A more recent approach has been developed by Mahdavi and Balasundaram [54] based on a dual coloring problem. Their algorithm employs DROP and CONSTELLATION to initialize the incumbent solution. Vertex dichotomy is proposed as a branching strategy whereby the vertex with minimum number of k-neighbors is branched upon at each node of the B&B tree. Upper bounding, on the other hand, employs a distance k-coloring approach. Proper coloring is achieved by two heuristics, the first of which is DSATUR (refer to [21]), applied at top levels of the B&B tree. For lower levels, a simple greedy heuristic is used, coloring the highest degree vertex in the power of the graph with the color not yet assigned to its neighbors. Computations were reported for the same test-bed of instances as [20] suggesting a better performance for $k=3$ than for $k=2$.

5.4.3 k-Core

The maximum k-core problem has been proven solvable in polynomial time. In fact, a simple greedy algorithm is capable of generating optimal solutions as follows: First, a vertex v of minimum degree $\delta(G)$ is picked, and if $\delta(G) \geq k$, then the whole graph is a k-core. If $\delta(G) < k$, then that vertex cannot be in a k-core. Hence, this vertex is deleted updating the graph $G := G - \{v\}$ and continuing recursively until a maximum k-core or the empty set is found.

Detecting the maximum k-core is often used as a pre-processing step for solving optimization problems seeking cliques or other clique relaxation structures. This is due to the fact that some of these structures are guaranteed to be a part of the largest k-core for a certain value of k. For example, since a k-plex of size s cannot contain a vertex of degree less than $s-k$, when searching for such a k-plex, we can recursively remove all vertices of degree less than $s-k$. Hence, any k-plex of size s is a part of the largest $(s-k)$-core. The process of computing the largest k-core

as a preprocessing or scale-reduction step for solving another problem is known as *peeling* in the literature and has been successfully applied for solving the maximum clique problem [2] and the maximum k-plex problem [9].

5.4.4 k-Plex

The maximum k-plex problem was proven NP-hard for any fixed positive integer k in [9]. Because of the intractability of the problem, heuristics have been developed both to find "good" solutions and to assist branch and bound for finding exact solutions. McClosky [57] employed a simple heuristic to find cliques, which were then extended to maximal k-plexes, for use as a lower bound for branch and bound. In [80], a heuristic was used to help prune a branch and bound tree for k-plex. When the size of the maximum k-plex exceeds $2k-2$, it will have diameter 2. Assuming this to be true greatly reduced the candidate set as they built a k-plex one vertex at a time in branch and bound.

Most currently available exact approaches for solving the maximum k-plex problem are adaptations of either branch and bound or branch and cut. Balasundaram et al. [9] formulate the maximum k-plex problem as an integer linear program and use valid inequalities based on independent sets of size at least k. These inequalities were generated by a simple greedy algorithm both for the whole problem and at local branches in the search tree. They successfully ran a branch and cut algorithm on large-scale instances of real life social networks known as the Erdös graphs and successfully solved the maximum 2-plex problem on graphs with 80% density and 350 vertices in less than 8 h.

The focus of [61] is on the minimum d-bounded-degree deletion problem, which yields a k-plex in the compliment graph. It generalizes the relationship between clique and vertex cover. They use an algorithm that finds a kernel of the minimum d-bounded degree deletion problem to guide their branching in branch and bound. A kernel is a subgraph of vertices that must contain the optimal solution and finding it can significantly reduce the size of the branch and bound tree. They report results superior to [9] with guided branching on the Erdös graphs and comparable results for DIMACS graphs as those reported in [9].

In [57], some of the most successful techniques for solving the maximum clique problem were adapted as part of a branch and bound algorithm for k-plex. They develop a co-k-plex coloring problem that bounds the size of the k-plex in the same way the coloring problem serves to bound the clique number of a graph. They use this to help provide upper bounds in adaptations of the basic clique algorithm [25] and Östergard's algorithm [62] for solving the maximum k-plex problem. They successfully solved instances of the maximum k-plex problem on DIMACS graphs for $k = 2, 3, 4$ but in general the results were not as successful as the branch and cut algorithm of [9] for larger graphs.

The Carraghan–Pardalos and Östergard algorithms for the maximum clique problem were also the inspiration of a more general algorithm for solving any

problem that displays heredity in [80]. The key difference from [57] is including a pruning technique based on bounds for the size of a k-plex within subgraphs induced by subsets of vertices of the form $\{v_i, \ldots, v_n\}$. These are developed naturally as the algorithm runs. If i is the lowest index on any member of the candidate set, we calculate the size of the largest k-plex in $\{v_i, \ldots, v_n\}$ as part of the algorithm, and while it may not be achieved for a given candidate set within $\{v_i, \ldots, v_n\}$, it is often significantly lower than the weight of the entire candidate set. In [57], they prune branches of the search tree where the weight of the current solution and the entire candidate set was less than the current optimal solution. Trukhanov et al. [80] use both this pruning technique and pruning where the weight of the candidate set is replaced with the known size of the optimal solution using the vertex with lowest index. The algorithm significantly outperformed the techniques of [9] and [57] in nearly every instance of DIMACS graphs on which they were run. Graphs of up to 500 vertices were solved optimally for $k = 2, 3, 4$ and even larger graphs were solved optimally for $k = 2$.

Recently, Wu and Pei [85] developed a parallel algorithm for enumeration of all the maximal k-plexes in the graph. Their work emphasizes the parallelization of the algorithm and comparison between serial and parallel algorithms performance. Unfortunately, they do not provide any numerical results for the well-known benchmark instances.

5.4.5 Quasi-clique

The maximum γ-quasi-clique problem was shown to be NP-hard for any $\gamma \in (0, 1)$ in [64]. Due to the lack of structural properties in quasi-cliques that are utilized in exact algorithms for other problems of interest, the maximum quasi-clique problem is extremely challenging to solve to optimality. In fact, to the best of our knowledge, the only published exact approaches for this problem are based on using the MIP formulations presented in Sect. 5.3.4 in conjunction with an off-the-shelf solver [64]. This allows to solve medium-scale problem instances with several hundreds of vertices and low edge density to optimality.

To solve large-scale instances, heuristics are commonly used in practice. Since quasi-clique possesses the weak heredity property, greedy randomized adaptive search procedures (GRASP) [36, 37] appear to be a natural method of choice. In short, GRASP is a heuristic framework based on the following idea: at each iteration it constructs a greedy randomized solution and searches the neighborhood of each constructed solution in order to find the corresponding local optimum. The effectiveness of this method for solving the maximum quasi-clique problem is confirmed by the results of experiments with very large networks reported in the literature. One remarkable example is due to Abello et al. [2,3], who analyzed the so-called *call network*. This network represents phone call data; it has phone numbers as its vertices, and two vertices are connected by an edge if there was a phone call between the corresponding numbers within a specified period of time. Abello et al.

analyzed a call network representing the data from AT&T telephone billing records. This one-day call network had 53,767,087 vertices and over 170 millions of edges. The network had 3,667,448 connected components, 302,468 of which had more than three vertices. The largest connected component had 44,989,297 vertices. First, Abello et al. attempted to solve the maximum clique problem in this giant connected component. For this purpose, they ran 100,000 GRASP iterations on ten parallel processors, which took about one and a half days. Only 14,141 of the 100,000 cliques they generated were distinct, with the largest clique detected being of size 32. They also used GRASP to solve the maximum γ-clique problem for $\gamma = 0.9, 0.8, 0.7$, and 0.5 in the giant connected component. The largest γ-cliques they detected had the sizes of 44, 57, 65, and 98, respectively.

5.5 Extensions

While this chapter is motivated by developments in social network analysis, clique-like structures naturally arise in many other important applications of complex networks, in which one is looking for large, "tightly knit" groups of elements. Depending on the particular application, such groups are often referred to as *clusters*, *modules*, *complexes*, *communities*, etc. If the elements in the considered complex system are represented as vertices (nodes) and the relationships between the elements are represented as edges (links, arcs), then, depending on the structure of interest for the particular application, clusters can be naturally described using some of the clique relaxation concepts described above.

Recent work exploiting some of the clique relaxation models in this emerging area of *network-based data mining* [14, 30] has been abundant. In particular, these concepts have been used in studying structural properties of stock markets [16, 17], unraveling molecular structures to facilitate drug discovery and compound synthesis [24, 38], and for identifying frequently occurring patterns in data sets (modeled as graphs) [15, 82]. In biology, quasi-cliques have been used to detect large clusters in protein interaction networks [13]. A survey of applications of clique detection models in biochemistry and genomics is given in [24]. In internet research, cohesive subgroups correspond to collections of densely connected web sites [77]. This helps to organize topically related web sites and thus facilitate faster search and retrieval of information from the web. In wireless communication, clustering the *connectivity graph* of a wireless network is used to introduce a hierarchy, which facilitates routing of information through the network [76]. Clique and other low diameter models have been used to define a cluster in a wireless network [28, 50]. These are just a few examples of recent publications in the rapidly growing body of literature dealing with applications of cliques and clique relaxations in a wide range of settings.

5.6 Conclusion

Social network analysis is emerging as an important tool in network-based data mining, which is applied to a wide variety of settings ranging from biological systems to finance, and studying clique relaxation models is one of the cornerstones of this methodology. The aim of this chapter is to open the door to this rich and, at the same time, largely unexplored research avenue for an interested reader. Even though most of the discussed clique relaxation models have been around for several decades in the social network literature, many of the theoretical, algorithmic, computational, and applied aspects of these models are only beginning to be addressed.

To illustrate the potential impact the research on clique relaxations may have on areas beyond social network analysis, we turn to history. While complete subgraphs had been studied by mathematicians earlier, introduction of the term "clique" to graph theory was triggered by the developments in social sciences mentioned in the beginning of this chapter. Since then, the concept of a clique has been widely used in a variety of applied settings and is central to some major theoretical and algorithmic developments in graph theory, computer science and operations research. We believe that clique relaxation models provide an excellent opportunity for even more important developments.

References

1. E. Aarts and J. Korst. *Simulated Annealing and Boltzmann Machines*. John Wiley & Sons Incorporated, Chichester, UK, 1989.
2. J. Abello, P.M. Pardalos, and M.G.C. Resende. On maximum clique problems in very large graphs. In J. Abello and J. Vitter, editors, *External memory algorithms and visualization*, volume 50 of *DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, pages 119–130. American Mathematical Society, 1999.
3. J. Abello, M.G.C. Resende, and S. Sudarsky. Massive quasi-clique detection. In S. Rajsbaum, editor, *LATIN 2002: Theoretical Informatics*, pages 598–612, London, 2002. Springer-Verlag.
4. R.D. Alba. A graph-theoretic definition of a sociometric clique. *Journal of Mathematical Sociology*, 3:113–126, 1973.
5. D. Applegate and D.S. Johnson. Maximum clique solver implementation, dfmax.c. ftp://dimacs.rutgers.edu/pub/challenge/graph/solvers/.
6. L. Babel. Finding maximum cliques in arbitrary and in special graphs. *Computing*, 46(4): 321–341, 1991.
7. E. Balas and J. Xue. Weighted and unweighted maximum clique algorithms with upper bounds from fractional coloring. *Algorithmica*, 15:397–412, 1996.
8. E. Balas and C. Yu. Finding a maximum clique in an arbitrary graph. *SIAM Journal of Computing*, 15:1054–1068, 1986.
9. B. Balasundram, S. Butenko, and I.V. Hicks. Clique relaxations in social network analysis: The maximum k-plex problem. *Operations Research*, 59:133–142, 2011.
10. B. Balasundram, S. Butenko, and S. Trukhanov. Novel approaches for analyzing biological networks. *Journal of Combinatorial Optimization*, 10:23–29, 2005.

11. R. Battiti and M. Protasi. Reactive local search for the maximum clique problem. Technical report, Algorithmica, 2001.
12. N. Berry, T. Ko, T. Moy, J. Smrcka, J. Turnley, and B. Wu. Emergent clique formation in terrorist recruitment. *The AAAI-04 Workshop on Agent Organizations: Theory and Practice, July 25, 2004, San Jose, California*, 2004. Online: http://www.cs.uu.nl/ virginia/aotp/papers.htm.
13. M. Bhattacharyya and S. Bandyopadhyay. Mining the largest quasi-clique in human protein interactome. In *IEEE International Conference on Artificial Intelligence Systems*, pages 194–199, Los Alamitos, CA, USA, 2009. IEEE Computer Society.
14. V. Boginski. Network-based data mining: Operations research techniques and applications. In J. Cochran, editor, *Encyclopedia of Operations Research and Management Science*. Wiley, 2011.
15. V. Boginski, S. Butenko, and P. Pardalos. Network models of massive datasets. *Computer Science and Information Systems*, 1:79–93, 2004.
16. V. Boginski, S. Butenko, and P. Pardalos. Statistical analysis of financial networks. *Computational Statistics & Data Analysis*, 48:431–443, 2005.
17. V. Boginski, S. Butenko, and P. Pardalos. Mining market data: a network approach. *Computers & Operations Research*, 33:3171–3184, 2006.
18. I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo. The maximum clique problem. In D.-Z. Du and P. M. Pardalos, editors, *Handbook of Combinatorial Optimization*, pages 1–74, Dordrecht, The Netherlands, 1999. Kluwer Academic Publishers.
19. J.M. Bourjolly, G. Laporte, and G. Pesant. Heuristics for "nding k-clubs in an undirected graph. *Computers and Operations Research*, 27:559–569, 2000.
20. J.M. Bourjolly, G. Laporte, and G. Pesant. An exact algorithm for the maximum k-club problem in an undirected graph. *European Journal of Operational Research*, 138:21–28, 2002.
21. D. Brélaz. New methods to color the vertices of a graph. *Communications of the ACM*, 22: 251–256, 1979.
22. C. Bron and J. Kerbosch. Algorithm 457: Finding all cliques on an undirected graph. *Communications of ACM*, 16:575–577, 1973.
23. M. Brunato, H. Hoos, and R. Battiti. On effectively finding maximal quasi-cliques in graphs. In V. Maniezzo, R. Battiti, and J.P. Watson, editors, *Proc. 2nd Learning and Intelligent Optimization Workshop, LION 2*, volume 5313 of *LNCS*. Springer Verlag, 2008.
24. S. Butenko and W. Wilhelm. Clique-detection models in computational biochemistry and genomics. *European Journal of Operational Research*, 173:1–17, 2006.
25. R. Carraghan and P. Pardalos. An exact algorithm for the maximum clique problem. *Operations Research Letters*, 9:375–382, 1990.
26. M. Chams, A. Hertz, and A. de Werra. Some experiments with simulated annealing for coloring graphs. *European Journal of Operational Research*, 32:260–266, 1987.
27. H. Chen, W. Chung, J. J. Xu, G. Wang, Y. Qin, and M. Chau. Crime data mining: A general framework and some examples. *Computer*, 37(4):50–56, 2004.
28. Y. P. Chen, A. L. Liestman, and J. Liu. Clustering algorithms for ad hoc wireless networks. In Y. Pan and Y. Xiao, editors, *Ad Hoc and Sensor Networks, Wireless Networks and Mobile Computing*, pages 145–164. Nova Science Publishers, New York, 2005.
29. N. Chiba and T. Nishizeki. Arboricity and subgraph listing algorithms. *SIAM J. Comput.*, 14:210–223, 1985.
30. D. J. Cook and L. B. Holder. Graph-based data mining. *IEEE Intelligent Systems*, 15(2):32–41, 2000.
31. R. H. Davis. Social network analysis: An aid in conspiracy investigations. *FBI Law Enforcement Bulletin*, 50(12):11–19, 1981.
32. A. H. Dekker. Social network analysis in military headquarters using CAVALIER. In *Proceedings of Fifth International Command and Control Research and Technology Symposium*, pages 24–26, Canberra, Australia, 2000.
33. R. Diestel. *Graph Theory*. Springer-Verlag, Berlin, 3 edition, 2006.
34. DIMACS. Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge. http://dimacs.rutgers.edu/Challenges/, 1995.

35. E. Durkheim. *The Division of Labor in Society*. Free Press, New York, [1893] 1984. Translated by W. D. Halls.
36. T. A. Feo and M. G. C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133, 1995.
37. T. A. Feo, M. G. C. Resende, and S. H. Smith. A greedy randomized adaptive search procedure for maximum independent set. *Operations Research*, 42:860–878, 1994.
38. I. Fischer and T. Meinl. Graph based molecular data mining - an overview. In W. Thissen, P. Wieringa, M. Pantic, and M. Ludema, editors, *Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics*, pages 4578–4582, Piscataway, NJ, 2004. IEEE.
39. M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.
40. M. Gendreau, P. Soriano, and L. Salvail. Solving the maximum clique problem using a tabu search approach. *Ann. Oper. Res.*, 41:385–403, May 1993.
41. R. V. Gould. Multiple networks and mobilization in the paris commune, 1871. *American Sociological Review*, 56:716–729, 1991.
42. R. Hodson. *Dignity at Work*. Cambridge University Press, Cambridge, England, 2001.
43. S. Homer and M. Pinedo. *Cliques, coloring, and satisfiability: Second DIMACS implementation challenge*, chapter Experiments with polynomial-time clique approximation algorithms on very large graphs, pages 147–167. American Mathematical Society, 1996.
44. D. J. Johnson and M. A. Trick, editors. *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge, Workshop, October 11-13, 1993*. American Mathematical Society, Boston, MA, USA, 1996.
45. D.S. Johnson and C.R. Aragon. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9:256–278, 1974.
46. D.S. Johnson, C.R. Aragon, L.A. McGoech, and C. Shevon. Optimization by simulated annealing: An experimental evaluation; part ii, graph coloring and number partitioning. *Operations Research*, 39:378–406, 1991.
47. H.C. Johnston. Cliques of a graph – variations of Bron-Kerbosch algorithm. *Int. J. Comput. Inf. Sci.*, 5:209–238, 1976.
48. R. Kopf and G. Ruhe. A computational study of the weighted independent set problem for general graphs. *Foundations of Control Engineering*, 12:167–180, 1987.
49. V. Krebs. Mapping networks of terrorist cells. *Connections*, 24:45–52, 2002.
50. P. Krishna, N. H. Vaidya, M. Chatterjee, and D. K. Pradhan. A cluster-based approach for routing in dynamic networks. *ACM SIGCOMM Computer Communication Review*, 27(2): 49–64, 1997.
51. E. Loukakis and C. Tsouros. A depth first search algorithm to generate the family of maximal independent sets of a graph lexicographically. *Computing*, 27:249–266, 1981.
52. R.D. Luce. Connectivity and generalized cliques in sociometric group structure. *Psychometrika*, 15:169–190, 1950.
53. R.D. Luce and A.D. Perry. A method of matrix analysis of group structure. *Psychometrika*, 14:95–116, 1949.
54. F. Mahdavi and B. Balasundaram. On inclusionwise maximal and maximum cardinality k-clubs in graphs. *Preprint*, 2010.
55. K. Makino and T. Uno. New algorithms for enumerating all maximal cliques. In *Proc. Ninth Scandinavian Workshop Algorithm Theory (SWAT '04)*, pages 260–272, 2004.
56. D. McAndrew. The structural analysis of criminal networks. In D. Canter and L. Alison, editors, *The Social Psychology of Crime: Groups, Teams, and Networks, Offender Profiling Series, III*, Aldershot, UK, 1999. Dartmouth.
57. B. McClosky and I. Hicks. Combinatorial algorithms for the maximum *k*-plex problem. *Journal of Combinatorial Optimization*, to appear.
58. G. A. Miller. WordNet, Princeton University. http://wordnet.princeton.edu, 2009.
59. R.J. Mokken. Cliques, clubs and clans. *Quality and Quantity*, 13:161–173, 1979.
60. J. Moody and D. R. White. Structural cohesion and embeddedness: A hierarchical concept of social groups. *American Sociological Review*, 68:103–127, 2003.

61. H. Moser, R. Niedermeier, and M. Sorge. Algorithms and experiments for clique relaxations - finding maximum *s*-plexes. In *Proceedings of the 8th International Symposium on Experimental Algorithms*, pages 233–244, 2009.
62. P. R. J. Östergård. A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics*, 120:197–207, 2002.
63. J. Pattillo and S. Butenko. Clique, independent set, and graph coloring. In J. Cochran, editor, *Encyclopedia of Operations Research and Management Science*, pages 3150–3163. Wiley, 2011.
64. J. Pattillo, A. Veremyev, S. Butenko, and V. Boginski. On the maximum quasi-clique problem. Working paper.
65. P. Paxton. Is social capital declining in the united states? a multiple indicator assessment. *American Journal of Sociology*, 105:88–127, 1999.
66. F. J. Roethlisberger and W. J. Dickson. *Management and the Worker*. Harvard University Press, Cambridge, MA, 1939.
67. R. B. Rothenberg, J. J. Potterat, and D. E. Woodhouse. Personal risk taking and the spread of disease: Beyond core groups. *Journal of Infectious Diseases*, 174 (Supp. 2):S144–S149, 1996.
68. M. Sageman. *Understanding Terrorist Networks*. University of Pennsylvania Press, Philadelphia, PA, 2004.
69. R. J. Sampson and B. W. Groves. Community structure and crime: Testing social-disorganization theory. *American Journal of Sociology*, 94:774–802, 1989.
70. J. Scott. *Social Network Analysis: A Handbook*. Sage Publications, London, 2 edition, 2000.
71. S. B. Seidman. Network structure and minimum degree. *Social Networks*, 5:269–287, 1983.
72. S. B. Seidman and B. L. Foster. A graph theoretic generalization of the clique concept. *Journal of Mathematical Sociology*, 6:139–154, 1978.
73. E. C. Sewell. A branch and bound algorithm for the stability number of a sparse graph. *INFORMS Journal on Computing*, 10(4):438–447, 1998.
74. P. Soriano and M. Gendreau. *Cliques, coloring, and satisfiability: Second DIMACS implementation challenge*, chapter Tabu search algorithms for the maximum clique problem, pages 221–242. American Mathematical Society, 1996.
75. P. Soriano and M. Gendreau. Diversification strategies in tabu search algorithms for the maximum clique problem. *Annals of Operations Research*, 63:189–207, 1996.
76. I. Stojmenovic, editor. *Handbook of Wireless Networks and Mobile Computing*. Wiley InterScience, 2002.
77. L. Terveen, W. Hill, and B. Amento. Constructing, organizing, and visualizing collections of topically related, web resources. *ACM Transactions on Computer-Human Interaction*, 6:67–94, 1999.
78. E. Tomita and T. Kameda. An efficient branch-and-bound algorithm for finding a maximum clique with computational experiments. *Journal of Global Optimization*, 37(1):95–111, 2007.
79. E. Tomita, S. Mitsuma, and H. Takahashi. Two algorithms for finding a near-maximum clique. Technical report, UEC-TR-C1, 1988.
80. S. Trukhanov, B. Balasundaram, and S. Butenko. Exact algorithms for hard node deletion problems in graphs. Working paper, 2011.
81. A. Veremyev and V. Boginski. Identifying large robust network clusters via new compact formulations of maximum *k*-club problems. *European Journal of Operational Research*, in revision, 2010.
82. T. Washio and H. Motoda. State of the art of graph-based data mining. *SIGKDD Explor. Newsl.*, 5(1):59–68, 2003.
83. S. Wasserman and K. Faust. *Social Network Analysis*. Cambridge University Press, New York, 1994.
84. D. R. Wood. An algorithm for finding a maximum clique in a graph. *Operations Research Letters*, 21(5):211–217, 1997.
85. B. Wu and X. Pei. A parallel algorithm for enumerating all the maximal *k*-plexes. In J. G. Carbonell and J. Siekmann, editors, *Emerging Technologies in Knowledge Discovery and Data Mining*, Lecture Notes in Computer Science, chapter 47, pages 476–483. Springer, Berlin, Germany, 2007.