B. Balasundaram · S. Butenko · I. V. Hicks[*] · S. Sachdeva

Clique Relaxations in Social Network Analysis: The Maximum k-plex Problem

Received: date / Accepted: date

Abstract This paper introduces and studies the *maximum k-plex problem*, which arises in social network analysis, but can also be used in several other important application areas, including wireless networks, telecommunications, and graph-based data mining. We establish NP-completeness of the decision version of the problem on arbitrary graphs. An integer programming formulation is presented and basic polyhedral study of the problem is carried out. A branch-and-cut implementation is discussed and computational test results on the proposed benchmark instances are also provided.

Keywords maximum k-plex · maximum clique · integer programming · branch-and-cut · social network analysis

1 Introduction

In the wake of the information revolution, the interest in studying the network structure of organizations, in particular criminal in nature, has increased manifold. Social network concepts, despite their versatility, have come to the forefront especially for these applications. A social network is usually represented by a graph, in which the set of vertices corresponds to the "actors" in a social network and the edges correspond to the "ties" between them [45]. Actors can be people, and examples of a tie between two actors include the acquaintance, friendship, or other type of association between them, such as visiting the same social event or place at the same time. Alternately, actors can be companies, with ties representing business transactions between them.

[*] This author's research was partially supported by NSF grant DMI-0521209

Department of Industrial and Systems Engineering, Texas A&M University, College Station, Texas 77843, USA.
{baski,butenko,ivhicks,sandeeps}@tamu.edu

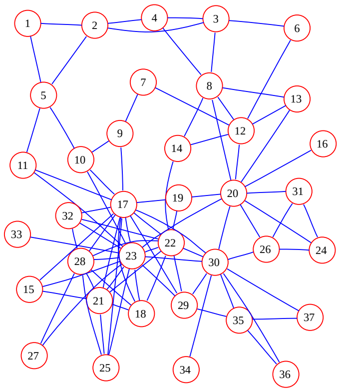
Thus, graphs can be used to conveniently model any such information and to make important deductions.

This paper introduces and studies the *maximum k-plex problem*, which arises in analysis of *cohesive subgroups* in social networks. *Social cohesion* is often used to explain and develop sociological theories. Members of a cohesive subgroup tend to share information, have homogeneity of thought, identity, beliefs, behavior, even food habits and illnesses [52]. Social cohesion is also believed to influence emergence of consensus among group members. Examples of cohesive subgroups include religious cults, terrorist cells, criminal gangs, military platoons, sports teams and conferences, work groups etc. Modeling a cohesive subgroup mathematically has long been a subject of interest in social network analysis. One of the earliest graph models used for studying cohesive subgroups was the *clique* model [35]. A clique is a subgraph in which there is an edge between any two vertices. However, the clique approach has been criticized for its overly restrictive nature [2, 52] and modeling disadvantages [47, 25].

Alternative approaches were suggested that essentially relaxed the definition of cliques. Clique models idealize three important structural properties that are expected of a cohesive subgroup, namely, *familiarity* (each vertex has many neighbors and only a few strangers in the group), *reachability* (a low diameter, facilitating fast communication between the group members) and *robustness* (high connectivity, making it difficult to destroy the group by removing members). Different models relax different aspects of a cohesive subgroup. [34] introduced a distance based model called *k-clique* and [2] introduced a diameter based model called *k-club*. These models were also studied along with a variant called *k-clan* by Mokken [38]. However, their originally proposed definitions required some modifications to be more meaningful mathematically. These drawbacks are pointed out and the models are appropriately redefined in [7], as described in Section 2. All these models emphasize the need for high reachability inside a cohesive subgroup and have their own merits and demerits as models of cohesiveness. The focus of this paper is on a degree based model introduced in [47] and called *k-plex*. This model relaxes familiarity within a cohesive subgroup and implicitly provides reachability and robustness.

Some direct application areas of social networks include studying terrorist networks [43, 9], which is essentially a special application of criminal network analysis that is intended to study organized crimes such as terrorism, drug trafficking and money laundering [36, 21]. Concepts of social network analysis provide suitable data mining tools for this purpose [17]. Figure 1 shows an example of a terrorist network, which maps the links between terrorists involved in the tragic events of September 11, 2001. This graph was constructed in [32] using the public data that were available before, but collected after the event. Even though the information mapped in this network is by no means complete, its analysis may still provide valuable insights into the structure of a terrorist organization. Another important application of these ideas is in internet research, where cohesive subgroups correspond to collections of densely connected web sites [49]. Topically related web sites are thus identified and organized to facilitate faster search and retrieval of

1 Wail Alshehri
2 Satam Suqami
3 Nabil al-Marabh
4 Raed Hijazi
5 Waleed Alshehri
6 Ahmed Alghamdi
7 Mohand Alshehri
8 Saeed Alghamdi
9 Fayez Ahmed
10 Mustafa Ahmed Al-Hisawi
11 Abdul Aziz Al-Omari
12 Hamza Alghamdi
13 Ahmed Alnami
14 Ahmed Al Haznawi
15 Mamoun Darkazanli
16 Mohamed Abdi
17 Marwan Al-Shehhi
18 Zakariya Essabar
19 Salem Alhazmi



20 Nawaf Alhazmi
21 Said Bahaji
22 Ziad Jarrah
23 Mohamed Atta
24 Abdussattar Shaikh
25 Mounir El Motassadeq
26 Khalid Al-Mihdhar
27 Zacarias Moussaoui
28 Ramzi Bin al-Shibh
29 Lofti Raissi
30 Hani Hanjour
31 Osama Awadallah
32 Agus Budiman
33 Ahmed Khalil Ibrahim Samir Al-Ani
34 Majed Moqed
35 Rayed Mohammed Abdullah
36 Faisal Al Salmi
37 Bandar Alhazmi

Fig. 1 The network surrounding the tragic events of September 11, 2001.

information from the web. Clique and other low diameter models have been popular in the area of wireless communication [18,33]. Clustering the *connectivity graph* of a wireless network introduces a hierarchy, otherwise absent in these dynamic networks. Existence of a hierarchy facilitates routing of information through the network. Efficient resource management, routing and better throughput performance can be achieved through adaptive clustering of these mobile nodes [48]. A similar principle is also used in *organizational management*, where social network analysis is also used to study organizational structure to suggest better work practices and improve communication and work flow [22]. Graph based data mining [19] has been used in studying structural properties of social networks [40] and stock markets [12,11], unraveling molecular structures to facilitate drug discovery and compound synthesis [24,16], and for identifying frequently occurring patterns in data sets (modeled as graphs) [51,10].

In spite of its potential applicability to a number of important practical situations, the optimization problems concerned with finding large k-plexes in a graph have not been studied from the mathematical programming perspective. It is surprising that since the introduction of the k-plex model and establishing its basic mathematical properties in the late 70's [47], it has been completely overlooked in mathematics, mathematical programming and computer science literature. This paper introduces the maximum k-plex problem to the mathematical programming community and analyzes its basic properties, including the computational complexity, mathematical programming formulations, and polyhedral structure. Results of computational experience with a branch-and-cut algorithm for finding a maximum k-plex are also provided. The remainder of this paper is organized as follows. Basic definitions, notations and background information of interest are presented in Section 2. Computational complexity analysis of the problem is carried out in Section 3. Section 4 introduces a binary integer programming formulation for the problem, presents some basic polyhedral study of the problem, and develops valid inequalities. A branch-and-cut algorithm is presented in Section 5, including implementation details and computational test results. Furthermore, the proposed formulation is studied in the context of the maximum clique problem. Finally the paper is concluded with a summary and directions for future work in Section 6.

2 Definitions, Notations, and Background

Let $G = (V, E)$ be a simple undirected graph representing a social network, $d_G(u, v)$ denote the length of a shortest path between vertices u and v in G and $diam(G) = \max_{u,v \in V} d_G(u, v)$ be the diameter of G. Denote by $G[S] = (S, E \cap (S \times S))$, the subgraph induced by S. A subset of vertices $S \subseteq V$ is a k-clique if $d_G(u, v) \leq k$ for all $u, v \in S$, and it is a k-club if $diam(G[S]) \leq k$. Note that a shortest path between two vertices in S may include vertices outside S. Hence, for a k-clique S there can exist two vertices $u, v \in S$ such that $d_G(u, v) \leq k$, but $d_{G[S]}(u, v) > k$ and thus u and v cannot exist in a k-club together. This is illustrated in Figure 2: The set $\{2, 3, 4, 5, 6\}$ is a 2-clique but not a 2-club. In a social network, it may be unreasonable to expect a cohesive subgroup to require outside members, and the concept of k-club overcomes this weakness common to k-cliques by bounding the diameter of the induced subgraph. From the definitions and above example it follows that any k-club in G is also a k-clique, but the converse is not true. However, k-clubs have certain drawbacks that we illustrate using a simple example involving 2-clubs. It is possible that there exists one vertex in a

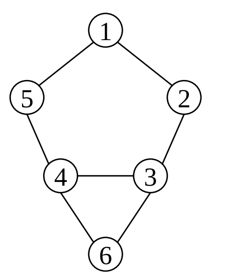


Fig. 2 2-clique vs. 2-club

2-club that is adjacent to all other vertices, making it a 2-club, but these neighbors are poorly connected among themselves. This is demonstrated by *star graphs* which have diameter two as the central vertex is adjacent to all other vertices, but the neighbors of the central vertex have no edges between them. Although these models ensure reachability, they may lack cohesiveness in terms of degree and connectivity. In particular, removal of just one central vertex in a star graph completely disconnects the graph.

The decision version of maximum k-clique and k-club problems are shown to be NP-complete on arbitrary graphs and on graphs with diameter more than k in [7]. That paper also studies the basic polyhedral properties and facets of the 2-club polytope. Heuristics for the maximum k-club problem are presented in [14] and an exact algorithm for the same is developed in [15]. Note that these models reduce to a clique for $k = 1$ and are relaxations of cliques for $k > 1$.

Degree based models of cohesion, which overcome the drawbacks inherent in the definitions of k-clique and k-club, were first introduced in [47] and [46]. Seidman [46] introduced the concept of a *k-core*, which is a subgraph with minimum degree at least k. In other words, $S \subseteq V$ is a k-core if $|N(v) \cap S| \geq k \;\; \forall\, v \in S$, where $N(v)$ denotes the set of neighbors of a vertex $v \in V$ in G. However, k-cores were noted to only indicate dense regions of the graph and not necessarily identify a cohesive subgroup [46, 52]. As suggested by Seidman, this approach was only to produce global measures that captured the cohesive subgroups as well as regions surrounding them. We will now describe a simple greedy algorithm that finds the largest k-core in a graph in polynomial time. Pick a vertex v of minimum degree $\delta(G)$, if $\delta(G) \geq k$ then we have a k-core. If $\delta(G) < k$, then that vertex cannot be in a k-core. Hence, delete the corresponding vertex, $G := G - v$ and continue recursively until a maximum k-core or the empty set is found. Note that even though these structures are easy to find, they only point out dense regions of the graph where interesting subgroups may be found.

The degree based model studied in this paper is called *k-plex* and was introduced by Seidman and Foster in [47]. A subset of vertices S is said to be a k-plex if the degree of every vertex in the induced subgraph $G[S]$ is at least $|S| - k$. That is, $S \subseteq V$ is a k-plex if the following condition holds:

$$deg_{G[S]}(v) = |N(v) \cap S| \geq |S| - k \quad \forall\, v \in S.$$

A k-plex is said to be *maximal* if it is not strictly contained in any other k-plex. We propose calling the cardinality of the largest k-plex in the graph as *the k-plex number* and denote it by $\rho_k(G)$. The *maximum k-plex problem* is to find the largest k-plex of the given graph. Note that, as with the maximum k-clique and maximum k-club problems, this reduces to the *maximum clique problem* [13] when $k = 1$ and is a relaxation of the clique requirement for all other $k > 1$, allowing for at most $k - 1$ non-neighbors inside the set. Figure 3 illustrates this concept: The set $\{1, 2, 3, 4\}$ is a 1-plex (clique), sets $\{1, 2, 3, 4, 5\}$ and $\{1, 2, 3, 4, 6\}$ are 2-plexes (maximal and maximum) and the entire graph is a 3-plex.

Apart from the basic definition that a graph G is k-plex if $\delta(G) \geq n - k$, an alternate characterization of k-plexes has been established in the following theorem from [47]. Let $N[v]$ denote the closed neighborhood of a vertex v, that is $N[v] = \{v\} \cup N(v)$.

Theorem 1 ([47]) *G is a k-plex if and only if for any k-element subset of vertices* $\{v_1, \ldots, v_k\} \subseteq V$, $V = \bigcup_{i=1}^{k} N[v_i]$.

In other words, the above theorem states that if G is a k-plex then any k vertices form a dominating set in the graph (A subset S of vertices of the graph $G = (V, E)$ is called a *dominating set* if $\cup_{i \in S} N[i] = V$). For instance, in the graph shown in Figure 3, $\{1, 5\}$ is not a dominating set, but with any other vertex, say 6, the resulting set $\{1, 5, 6\}$ is a dominating set, and

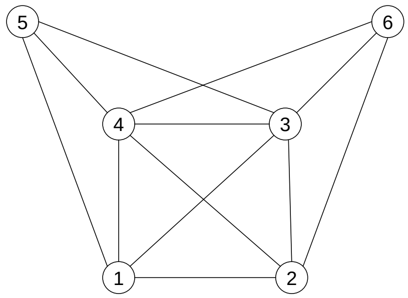


Fig. 3 Illustration of k-plexes for $k = 1, 2, 3$

so is any other triplet of vertices. The paper by Seidman and Foster also establishes some of the basic graph theoretic properties of a k-plex that are stated here. Let graph G be a k-plex. Then,

1. Any vertex-induced subgraph of a k-plex is a k-plex.
2. If $k < \frac{(n+2)}{2}$, then $diam(G) \leq 2$.
3. $\kappa(G) \geq n - 2k + 2$,

where the vertex connectivity $\kappa(G)$ is the minimum number of vertices whose removal results in a disconnected or trivial graph [29].

By definition, members of a k-plex S can have at most $k-1$ non-neighbors inside S. Thus, k-plexes with low k values ($k = 2, 3$) provide good relaxations of clique that closely resemble the cohesive subgroups that can be found in real life social networks. In addition, the above results indicate that a k-plex, besides being a natural generalization of a clique, also retains the properties of a clique such as low diameter and high connectivity, for low values of k. The k-plex model overcomes the disadvantages of k-cliques and k-clubs by directly limiting the number of *non-neighbors* inside the cohesive subgroup. This structure imposes a degree bound that varies with the size of the group and hence ensures a cohesive subgroup even as the size of the group varies. Implicitly, it also achieves reachability and robustness. By allowing some strangers in a social group, k-plex provides a more realistic alternative to model cohesive subgroups in a social network.

Maximum clique problem is closely related to another well known graph problem, which is the *maximum independent set problem*. An independent set (or stable set) is a subset of vertices such that there does not exist an edge between any two vertices in that set. In other words, the subgraph induced by an independent set is edgeless. A subset of vertices forms a clique in $G = (V, E)$ if and only if it forms an independent set in the complement graph $\bar{G} = (V, \bar{E})$. Naturally, the k-plex can also be closely related to a similar problem on the complement graphs which we formalize as follows. We call a subset of vertices S of a graph $G = (V, E)$ a *co-k-plex* if $|N(i) \cap S| \leq k-1$ for all $i \in S$. In other words, the induced subgraph $G[S]$ has a maximum degree of $k-1$ or less. It should be noted that S is a co-k-plex in G if and only if S is a k-plex in the complement graph $\bar{G}$. In particular, 1-plex is a clique and a co-1-plex is an independent set. Thus, k-plexes and co-k-plexes provide a systematic way to generalize two important graph models.

3 Computational Complexity

This section presents computational complexity results for the problem of interest. The decision version of the maximum k-plex problem can be stated as follows:

> k-Plex: Given a simple undirected graph $G = (V, E)$ and positive integer constants c, k, does there exist a k-plex of size c in G?

Theorem 2 *k-Plex is NP-complete for any constant positive integer k.*

Proof. We prove this by reducing Clique [26], a well-known NP-complete problem, to k-Plex. Given an instance $\langle G = (V, E), c\rangle$ of Clique, we construct an instance $\langle G' = (V', E'), c'\rangle$ in polynomial time such that G has a clique of size c *if and only if* G' has a k-plex of size c'. To construct G', we expand G by adding $k-1$ copies of the complete graph of order $n = |V|$. Denote the vertex set of the r^{th} such copy by $V_r,\ r = 1, \dots, k-1$, where $V_r = \{1_r, \dots, n_r\}$, and let $R = \bigcup_{r=1}^{k-1} V_r$. Put $V' = V \cup R$ and $E' = E \cup \hat{E} \cup \tilde{E}$, where

$$\hat{E} = \{(i, j_r) : i \in V, j_r \in V_r, i \neq j, r = 1, \dots, k-1\}$$

and

$$\tilde{E} = \{(i_p, j_r) : i_p \in V_p, j_r \in V_r, i \neq j, p, r = 1, \dots, k-1\}.$$

In other words, the set $\hat{E}$ represents the edges between V and R, where every vertex $u \in V$ is connected to every vertex in every complete graph except its copies, *i.e.* u is adjacent to every vertex in $R \setminus \{u_1, \dots, u_{k-1}\}$. The set $\tilde{E}$ includes the cross edges between distinct V_p and V_r, as well as all possible edges between vertices in V_p, $p = 1, \dots, k-1$. In other words, every vertex $u_p \in V_p, p = 1, \dots, k-1$ is adjacent to all the vertices in $V_r \setminus \{u_r\},\ r = 1, \dots, k-1$. Putting $c' = c + (k-1)n$ completes the reduction. Note that the instance $\langle G' = (V', E'), c'\rangle$ can be constructed in polynomial time. Figure 4 illustrates this transformation when G is a path on three vertices a, b and c.

We now show that if there exists a clique of size c in G then G' has a k-plex of size c'. Let $C \subseteq V$ induce a clique of size $c = |C|$ in G. We claim that the set $S = C \cup R$, where $|S| = c + n(k-1) = c'$, is a k-plex. For any $u \in C$, there exist $c-1$ neighbors inside C, and $(n-1)(k-1)$ neighbors in R. Thus, for $u \in C$, $deg_{G[S]}(u) = c - 1 + (n-1)(k-1) = c' - k$. For any $v_r \in R$, there exist $(n-1)(k-1)$ neighbors in R and c neighbors in C if $v \notin C$, and $c-1$ neighbors in C if $v \in C$. Again, for $v_r \in R$, $deg_{G[S]}(v_r) \geq c - 1 + (n-1)(k-1) = c' - k$. Hence, S induces a k-plex of size c'.

We now establish the other direction stating that if there exists a k-plex of size c' in G' then G has a clique of size c. Let S be a k-plex of size $c' = c + n(k-1)$. Let $P = R \setminus S$ denote the set of vertices from R not

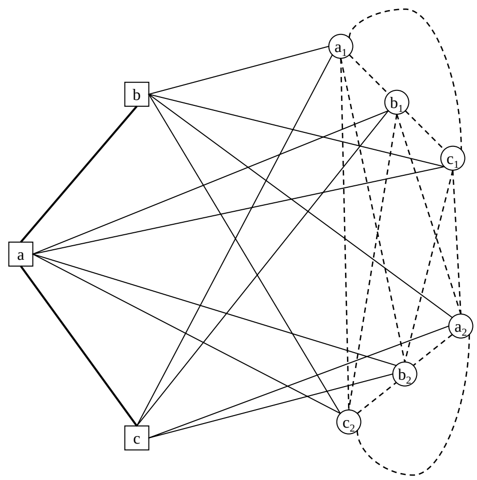


Fig. 4 Illustration of the 3-PLEX instance G'. Original graph G in box-vertices and heavy edges. $\hat{E} \cup \tilde{E}$ is denoted by dashed edges.

included in the k-plex and let $|P| = p$. Then, the c' vertices in S consist of $n(k-1) - p$ vertices in $S \cap R$ and $c + p$ vertices in $S \cap V$. Without loss of generality, suppose that $S \cap V = \{1, \ldots, c+p\}$ and further assume that for each $i \in S \cap V$ there exist q_i copies of i in P that are left out of the k-plex. Since every $i \in S \cap V$ has $p - q_i$ neighbors in P, we know that

$$|N(i) \cap (S \cap R)| = (n-1)(k-1) - (p - q_i).$$

Since S is a k-plex, $\forall\ i \in S \cap V$:

$$deg_{G[S]}(i) = |N(i) \cap (S \cap R)| + |N(i) \cap (S \cap V)| \geq c + n(k-1) - k,$$

$$\Rightarrow |N(i) \cap (S \cap V)| \geq c + p - 1 - q_i. \tag{1}$$

Recall that each q_i is a non-negative integer counting copies of vertex $i \in S \cap V$ in P and note that P can contain vertices that are not copies of any vertex in $S \cap V$. Thus, we have $\sum_{i=1}^{c+p} q_i \leq p$. Hence, there can exist at most p terms, q_i, in that sum that are strictly greater than 0, meaning that there exist at least c terms in that equation, that are equal to 0. Without loss of generality, suppose that $q_i = 0,\ i \in \{1, \ldots, c\}$. Now, let $C = \{1, \ldots, c\}$. We already know from (1) that for all $i \in C \subseteq S \cap V = \{1, \ldots, c+p\}$:

$$|N(i) \cap (S \cap V)| \geq c + p - 1 - q_i = c + p - 1.$$

But $|S \cap V| = c + p$, so for all $i \in C$,

$$|N(i) \cap (S \cap V)| = c + p - 1.$$

Thus, every vertex in $C \subseteq S \cap V$ is adjacent to every vertex in $S \cap V$. Hence, every vertex in C is adjacent to every other vertex in C. Therefore C induces a clique of size c in G. This completes the proof. □

This complexity result demonstrates that the maximum k-plex problem is hard not only because it is a generalization of the maximum clique problem, but it is a hard problem in its own respect, as Theorem 2 states that the decision version of the problem is NP-complete for every constant k.

4 Mathematical Programming Approaches

This section presents an integer programming formulation of the maximum k-plex problem followed by preliminary polyhedral study of the problem. Valid inequalities for the polytope under study are also presented.

4.1 Integer Programming Formulation

Given a graph $G = (V, E)$ with $|V| = n$, recall that $N[i]$ is the closed neighborhood of a vertex i and $\rho_k(G)$ is the k-plex number of G. Let $\bar{d}_i = |V \setminus N[i]|$ denote the degree of vertex i in the complement graph $\bar{G} = (V, \bar{E})$. Further assume that $k > 1$ since the $k = 1$ case yields the well known maximum clique problem. The following 0-1 program finds the largest k-plex in G.

$$\rho_k(G) = \max \sum_{i \in V} x_i \tag{2}$$

subject to:

$$\sum_{j \in V \setminus N[i]} x_j \leq (k-1)x_i + \bar{d}_i(1 - x_i) \qquad \forall\, i \in V \tag{3}$$

$$x_i \in \{0, 1\} \qquad \forall\, i \in V \tag{4}$$

In this formulation, $x_i = 1$ if and only if $i \in V$ is in the k-plex and $x_i = 0$ otherwise. Constraint (3) ensures that if a vertex i is in the k-plex then it has at most $k - 1$ non-neighbors inside the k-plex. The constraint is made redundant for vertices not in the k-plex.

4.2 Polyhedral Study

Let $Q(G) \subseteq \{0, 1\}^n$ denote the collection of feasible binary vectors of the aforementioned formulation (2) - (4). Then the *k-plex polytope* $P_k(G) = conv(Q(G))$, the convex hull of the feasible points. The following theorem establishes the trivial facets of the k-plex polytope.

Theorem 3 *Let $P_k(G)$ denote the k-plex polytope of a given graph $G = (V, E)$, where $k > 1$. Then,*

1. *$dim(P_k(G)) = n$.*

2. *$x_i \geq 0$ induces a facet of $P_k(G)$ for every $i \in V$.*
3. *$x_i \leq 1$ induces a facet of $P_k(G)$ for every $i \in V$.*

Proof: We will use following notations in the proof. Let e_i be the unit vector with i^{th} component 1 and the rest 0; $e_{ij} = e_i + e_j$

1. This is shown by demonstrating $n + 1$ affinely independent points in $P_k(G)$. The points $\mathbf{0}, e_1, e_2, \ldots, e_n$ are clearly $n + 1$ affinely independent points in $P_k(G) \subset \mathbb{R}^n$. Hence, $dim(P_k(G)) = n$.
2. Let $F = \{x \in P_k(G) : x_i = 0\}$. Since an empty set or any vertex by itself is a k-plex, we have $\mathbf{0}, e_j$ for all $j \in V \setminus \{i\}$ forming n affinely independent points in F. This shows that $dim(F) = n - 1$ and it is a facet.
3. Let $F' = \{x \in P_k(G) : x_i = 1\}$. We first observe that every vertex and any pair of vertices form a k-plex for any k such that $1 < k < n$. Then e_i and e_{ij} for all $j \in V \setminus \{i\}$ form n affinely independent points in F', indicating that $dim(F') = n - 1$ and it is a facet. □

4.3 Valid inequalities

The following valid inequalities are derived by identifying induced subgraphs that are not k-plexes and hence cannot be present in any k-plex. Although the result that every vertex-induced subgraph of a k-plex is a k-plex, is presented in [47], the following explanation is given here for the sake of clarity. Note that if G is a k-plex, the minimum degree $\delta(G)$ is at least $n-k$. In the graph G' obtained by deleting any vertex from G, the degree of all the vertices and hence the minimum degree, can drop by at most 1. Hence, the new graph continues to be a k-plex as $\delta(G') \geq (n-1)-k$. Also note that any k-element subset of vertices is a k-plex and any k-plex is also a $k+r$-plex ($1 \leq r \leq n-k$). Before we introduce the first family of inequalities, we present the following two lemmas.

Lemma 1 *Let k be even. Then, there does not exist a co-k-plex that contains a k-plex of size $2k - 1$.*

Proof. Let G be a co-k-plex on n vertices. Assume that $n \geq 2k - 1$ as the result is trivial otherwise. Now suppose that S is a k-plex of size $2k - 1$ in G. Then we have

$$|N(i) \cap S| \geq 2k - 1 - k = k - 1 \quad \forall i \in S.$$

Since G is co-k-plex we have,

$$|N(i) \cap S| \leq |N(i)| \leq k - 1 \quad \forall i \in S.$$

The two conditions then imply that the induced graph $G[S]$ is regular with all degrees equal to $k-1$ and is of order $2k-1$. But $k-1$ is odd and we cannot have an odd number of vertices of odd degree. This contradiction establishes that S does not exist. □

Note that this bound is sharp since the graph family $G_k = K_k \cup K_{k-1}$, the union of complete graphs, for each k forms a co-k-plex of size $2k - 1$, which contains $K_{k-1} \cup K_{k-1}$, a k-plex of size $2k-2$. Figure 5 illustrates this when $k = 4$.

Lemma 2 *Let k be odd. Then, there does not exist a co-k-plex that contains a k-plex of size $2k$.*

Proof. As before, let G be a co-k-plex on n vertices ($n \geq 2k$). Suppose that S is a k-plex of size $2k$ in G. Then we have

$$|N(i) \cap S| \geq 2k - k = k \quad \forall i \in S.$$

Since G is co-k-plex we have,

$$|N(i) \cap S| \leq |N(i)| \leq k - 1 \quad \forall i \in S.$$

This contradiction establishes that S does not exist. □

This bound is also sharp since the following family of graphs have $2k$ vertices forming a co-k-plex containing a k-plex of size $2k-1$. Construct the graph $G_k = (V, E)$, where

$$V = V' \cup \{2k\}, \quad V' = \{1, \ldots, 2k-1\},$$

and

$$E = \{(i,j) : i \in V' \text{ and } j = i+1, \ldots, \left(i + \frac{k-1}{2}\right) \mod (2k-1)\}.$$

Maximum degree in G_k is $k-1$ and hence it is a co-k-plex of order $2k$. The induced subgraph $G_k[V']$ is a $(k-1)$-regular k-plex of order $2k-1$ in which every vertex has exactly $k-1$ neighbors and non-neighbors each. It is also known as an *antiweb* and its complement is known as a *web*. Webs were introduced in [50] to generalize odd hole and antihole inequalities developed in [41] for the independent set polytope. Figure 5 illustrates this when $k = 5$.

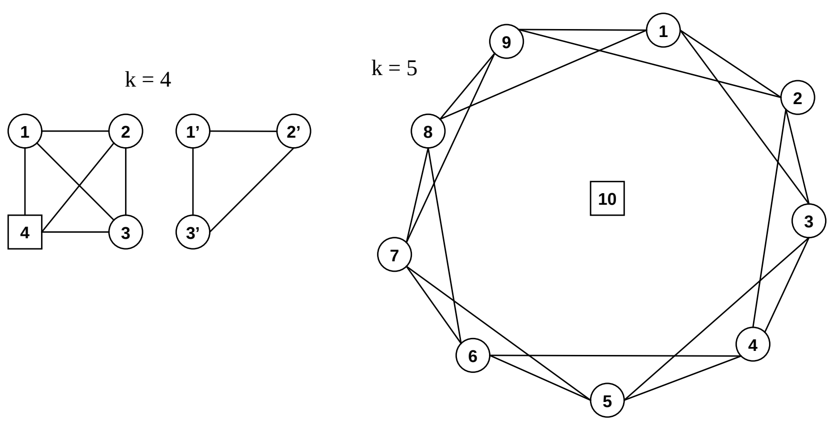


Fig. 5 Graphs demonstrating the sharpness of the bounds in Lemmas 1 and 2. The circled vertices form the said k-plexes

We next present three types of valid inequalities for the k-plex polytope: Independent set inequalities, hole inequalities, and co-k-plex inequalities.

Independent Set Inequalities. Let $I \subseteq V$ be an independent set. Note that no k-plex can contain an independent set of more than k vertices as $k+1$ or more independent vertices do not form a k-plex. Let $\mathcal{I}_{k+1}$ represent the collection of all maximal independent sets of size $k+1$ or more in G. Then we have the following family of valid inequalities.

$$\sum_{i \in I} x_i \leq k \qquad \forall\, I \in \mathcal{I}_{k+1} \tag{5}$$

Hole Inequalities. Let $H \subseteq V$ be a hole (induced chordless cycle). If $|H| \leq k+2$, then H is a k-plex. Now suppose $|H| > k+2$, then H is not a k-plex and for every proper subset $S \subset H$, we have $\delta(G[S]) \leq 1$. Hence, if $|S| - k \geq 2$, S is not a k-plex. Thus, any k-plex can contain at most $k+1$ vertices from the hole and this bound is sharp. Let $\mathcal{H}_{k+3}$ represent the collection of all holes of size $k+3$ or more in G. Then we have the following family of valid inequalities.

$$\sum_{i \in H} x_i \leq k+1 \qquad \forall\, H \in \mathcal{H}_{k+3} \tag{6}$$

Co-k-plex Inequalities. Lemmas 1 and 2, when combined, imply that the size of a maximum k-plex in any co-k-plex is less than or equal to $r_k = 2k-1-\frac{1+(-1)^k}{2}$. Let $\mathcal{J}_{r_k+1}$ represent the collection of all maximal co-k-plexes of size more than r_k in G. We have the following family of valid inequalities.

$$\sum_{i \in J} x_i \leq r_k \qquad \forall\, J \in \mathcal{J}_{r_k+1} \tag{7}$$

Observe that, if J is a *maximal* co-k-plex, there does not exist another co-k-plex of which J is a proper subset. Then, for every $v \in V \setminus J$, at least one of the following conditions must hold.

1. $\exists\, j \in J \cap N(v)$ such that $|N(j) \cap J| = k-1$ and including v would cause degree of j in the induced subgraph to be k.
2. $|N(v) \cap J| \geq k$ and hence upon inclusion v would have degree k or more in the induced subgraph.

The next theorem uses this observation to show that for $k = 2$ the co-2-plex inequalities actually form facets for the 2-plex polytope, $P_2(G)$.

Theorem 4 *Let $P_2(G)$ denote the 2-plex polytope described in Section 4.2. Then, the co-2-plex inequality given by,*

$$\sum_{i \in J} x_i \leq 2, \tag{8}$$

where J is a maximal co-2-plex with $|J| \geq 3$, induces a facet of $P_2(G)$.

Proof. First, recall that any 2 vertices from J form a 2-plex. Second, for every $v \in V \setminus J$, the above two conditions for a maximal co-2-plex imply the existence of two vertices $u, w \in J$ such that $\{v, u, w\}$ is a 2-plex. Indeed, if the first case holds, let $u \in J \cap N(v)$, then $N(u) \cap J = \{w\}$ and $\{v, u, w\}$ is a 2-plex. If the second case holds, $\{u, w\} \subseteq J \cap N(v)$ and again $\{v, u, w\}$ is a 2-plex. We use these observations to construct n affinely independent vectors that lie on the face defined by $F = \{x \in P_2(G) : \sum_{i \in J} x_i = 2\}$, so F is $(n-1)$-dimensional and hence it is a facet. Without loss of generality, assume that $J = \{1, \ldots, r\}$ and $V \setminus J = \{r+1, \ldots, n\}$, where $r \geq 3$. Let, as before, $e_i \in \mathbb{R}^n$ denote the unit vector with i^{th} component one and all others zero. The said affinely independent vectors $x^1, \ldots, x^n$ are constructed as follows.
$x^v = e_v + e_r, \quad \forall\, v = 1, \ldots, r-1;$
$x^r = e_1 + e_2$ (note that x^r is distinct from $x^1, \ldots, x^{r-1}$ as $r \geq 3$);
$x^v = e_v + e_u + e_w, \quad \forall\, v = r+1, \ldots, n$, where for each $v \in V \setminus J$, $u, w \in J$ are the particular vertices described before. Clearly, $x^v \in F$ and it can be easily verified that these vectors are affinely independent. Thus, the co-2-plex inequalities produce facets for the 2-plex polytope. □

Although co-k-plex inequalities form facets of $P_k(G)$ for $k = 1, 2$, they do not for $k \geq 3$. Consider a graph $G = (V, \emptyset)$ with at least k vertices. Note that G is a co-k-plex and the corresponding inequality $\sum_{i \in V} x_i \leq r_k$ is not supporting since $\rho_k(G) = k$ and there is no $x \in P_k(G)$ that satisfies it at equality. Hence, these inequalities do not form facets of $P_k(G)$ for all G. This is in contrast to the results known for $k = 1, 2$. The reason is $r_k = k$ for $k = 1, 2$ and every graph G with at least k vertices has a k-plex of size $r_k = k$. The next natural question, if they form facets when G is a co-k-plex with $\rho_k(G) = r_k, \quad k \geq 3$ is also settled in the negative by the following counterexamples.

Assume that k is even. Construct graphs G of arbitrary order $n \geq r_k$ as the union of $n - r_k$ clique components of size one and two clique components of size $k - 1 = r_k/2$. Then G is a co-k-plex with the two "large" clique components forming a k-plex of size r_k. Suppose $F = \{x \in P_k(G) : \sum_{i \in V} x_i = r_k\}$ is a facet of $P_k(G)$. Since $P_k(G)$ is an integral polytope, the extreme points of F are also integral and F is a convex hull of those integral vectors. Consider one such binary vector $x^o \in F$. If $x_i^o = 1$ for some i that is a one-vertex clique component of G, for x^o to be feasible we have $\sum_{j \in V \setminus N[i]} x_j^o \leq k - 1$. But since $V \setminus N[i] = V \setminus \{i\}$, we have $\sum_{i \in V} x_i^o \leq k$, which contradicts the fact that $x^o \in F$ as $r_k > k$. Hence, the components of extreme points of F corresponding to one vertex components of G are all zeros. Hence, there exists *exactly one extreme point* in $P_k(G)$ that satisfies $\sum_{i \in V} x_i \leq r_k$ at equality which is the characteristic vector of $K_{k-1} \cup K_{k-1}$. Thus, F is 0-dimensional and not a facet.

For odd k, we can have arbitrarily large graphs by adding single vertex components to the antiweb $G_k[V']$ constructed before. By using similar arguments, we can again show that there exists only one point in the k-plex polytope that satisfies the co-k-plex inequality at equality. From these observations we can conclude that $\rho_k(G) = r_k$ is only a necessary condition for co-k-plex inequality to induce a facet of $P_k(G)$. Identifying co-k-plex graphs for which the co-k-plex inequalities form facets of the k-plex polytope is important as these inequalities could be lifted to yield facets for a graph containing such structures. It is also a well-known fact that a graph is perfect if and only if its clique polytope is completely characterized by all the maximal independent set inequalities and non-negativity constraints [20]. Similarly, we could explore *k-plex perfectness* of graphs whose k-plex polytope can be completely described by the co-k-plex inequalities described here and non-negativity constraints. This is an interesting topic for future research.

5 Branch & Cut Framework

This section describes a simple branch-and-cut (BC) implementation incorporating the maximal independent set (MIS) cuts for the maximum k-plex problem. The aim of this part of the paper is to judge the effectiveness of MIS cuts in solving the problem of interest, the order and size of instances that can be solved under this simple framework, and to provide some benchmark instances for this problem. The experiments were conducted for $k = 1$ and 2. Even though the case of $k = 1$ corresponds to the well-researched maximum clique problem, these results illustrate the difference in the performance of our approach for two consecutive values of k.

5.1 Implementation Details

Branch-and-cut methods are popular and effective in optimally solving a wide variety of combinatorial optimization and general integer programming problems [42,3,6,5,4]. These methods incorporate cutting planes in solving the linear programming (LP) relaxation at the nodes of a branch-and-bound tree to get tighter bounds. Although BC methods have been successfully applied to solve several hard combinatorial optimization problems, tailoring a BC algorithm to effectively solve a specific problem is a delicate task that requires attention in itself in terms of extensive experimentation and tuning. For more information on BC methods and for other useful references, see [37].

In our experiments, the order and edge density of the graphs are varied and the maximum k-plex problem is solved on these graphs using our implementation of a MIS cuts based BC algorithm. The BC algorithm was implemented using ILOG CPLEX 9.0® [30] and the MIS cuts were generated using a greedy algorithm. The greedy algorithm starts by adding an arbitrary vertex and then removes its neighbors from the graph. Then it proceeds in a similar fashion by adding the vertex of minimum degree in the residual graph to the MIS and removing its neighbors until there are no more vertices to add. The following paragraphs describe the settings used and other relevant implementation details.

Two types of cuts, *local cuts* and *global cuts* were employed to solve the problem. Global cuts, which are valid at every node of the BC tree, are generated by finding maximal independent sets in the graph instance using a greedy algorithm. The greedy algorithm runs from every vertex, MIS are generated and distinct sets of size greater than k are stored, but never more than MaxGlobalCuts cuts are generated which is an externally set constant. The distinction between adding these as global cuts rather than constraints in the original system is that, CPLEX applies them only when they are violated, thereby keeping the size of the system small. Local cuts, which are valid at the node in which they're generated and for all its child nodes, are also generated in a similar fashion and implemented using the *goals* feature of CPLEX. They are generated using a constant *skip factor* of 32 which means local cuts were generated every 32 nodes in the BC tree. In the case of local cuts however, the vertices corresponding to variables fixed at zero are deleted before MIS are found, and only variables with high fractional value (≥ 0.5) are used as starting vertices for finding MIS. The *round of cuts* generated (distinct and always fewer than n) using only the violated MIS inequalities are added to the system and CPLEX re-solves the problem at that node and handles the cut management from that point onwards. It should be noted that CPLEX generates its own classes of cuts to solve any given MIP. These were turned off and only MIS local and global cuts were used in the BC algorithm. In addition, the number of rows in the problem with cuts added is limited to 3 times the original number of rows by setting the CPLEX parameter *CutsFactor* to 3. The biggest advantage of using the framework provided by CPLEX is the effective default settings that take care of the branching process, node selection, variable selection, primal heuristics, pre-solving among others, while the bounding is done by solving the LP relaxation with the user specified cuts.

The implementation described above is simple enough to help us judge the effectiveness of MIS cuts without intensifying the BC algorithm. Note however that, several critical issues in designing a sophisticated BC algorithm, such as using a dynamic global cut pool instead of the fixed size option offered by CPLEX and lifting local cuts so that they are globally valid can be considered. Even some of the basic settings can be improved such as using methods other than greedy algorithm to generate MIS cuts, distinguishing and selecting cuts from the violated cuts based on quality measures (such as maximum violation or maximum depth which is the Euclidean distance of the cut from the the point being separated), varying the frequency at which cuts are added dynamically (so that more cuts are generated and applied where the violation is "high") and the size of the cut pool that is applied and its management. These are directions that need to be carefully considered in the future in order to develop a more powerful BC algorithm for this problem.

5.2 Computational Test Results

The test bed of instances used in our experiments consists of a set of Erdös collaboration networks [28,8], clique instances from Second DIMACS challenge [23,31], and graphs of various order and size generated using *Sanchis generators* [44]. In scientific collaboration networks, the vertices represent scientists, and an edge connects two of them if they co-authored at least one paper. Erdös collaboration networks considered in [28,8] are centered around Paul Erdös and include authors who are connected to Erdös through a "path"

Table 1 Erdös networks: The number of vertices, edges, edge density, and the maximum k-plex size for $k = 1, \ldots, 4$.

Graph $G = (V, E)$	$\lvert V \rvert$	$\lvert E \rvert$	Edge Density	$\rho_k(G)$ for $k = \ldots$				
				1	2	3	4	5
ERDOS-97-1.NET	472	1314	0.0118212	7	8	9	11	12
ERDOS-98-1.NET	485	1381	0.0117662	7	8	9	11	12
ERDOS-99-1.NET	492	1417	0.0117315	7	8	9	11	12
ERDOS-97-2.NET	5488	8972	0.0005959	7	8	9	11	12
ERDOS-98-2.NET	5822	9505	0.0005609	7	8	9	11	12
ERDOS-99-2.NET	6100	9939	0.0005343	8	8	9	11	12

of collaborators. We used the following Erdös collaboration networks available from [8] in our experiments: ERDOS$-x-y$.NET, where x represents the last two digits of the year for which the network was constructed, and y represents the largest *Erdös number* of a scientist represented by a vertex in the graph, *i.e.*, the largest distance between Erdös vertex and any other vertex in the graph. For example, the vertices in graph ERDOS-99-2 correspond to 6100 authors who co-authored a paper either with Erdös or with at least one of his co-authors. We considered such networks for years 1997-1999 and $y = 1$ and 2. We will refer to ERDOS$-x-y$.NET graph as the y-neighborhood Erdös network for year x. Note that in the instances we used the vertex corresponding to Erdös himself is excluded.

It should be noted that in experiments with Erdös networks our goal was to solve the maximum k-plex problem to optimality on large-scale, real-life social network instances. On the other hand, in the set of experiments with DIMACS and Sanchis graphs our intention was to get a sense of the influence of order and density of graphs on the BC algorithm and the effectiveness of MIS cuts. The Sanchis generator available at [23] produces graphs with known maximum clique size with a specified number of vertices, edges and a *construction parameter*, r. In our experiments, the maximum clique size was fixed at $\lceil \frac{n}{3} \rceil$ and the construction parameter r which has to be an integer from interval $[0, \frac{n}{\omega(G)})$ was set at 1.

The basic settings for the experiments were as follows. The maximum k-plex problem was solved using the CPLEX BC implementation on a 3.06 Ghz PENTIUM®-4 computer for $k = 1, 2$. In experiments with Erdös networks, the following CPLEX settings were used: The MAXGLOBALCUTS parameter was set to 9; no more than 3 local cuts were added every 32 nodes of the branch-and-cut tree; only variables with fractional value ≥ 0.75 were used as starting vertices for finding MIS for local cuts.

Table 1 summarizes the input information and the maximum k-plex size for the Erdös networks analyzed, while Table 2 shows the run time of the BC algorithm on 1-neighborhood Erdös networks (ERDOS-x-1.NET, $x = 97, 98, 99$) for different values of k. Real-life social networks tend to be large but sparse, as illustrated by the analyzed instances. The large number of vertices of a small degree allows us to significantly reduce the graph size by recursively applying the following procedure similar to "peeling" proposed in [1] for the maximum clique problem. Suppose that we know that $\rho_k(G) \geq \rho$ for some lower bound ρ. Then any $i \in V$ such that $deg_G(i) < \rho - k$ cannot

Table 2 The run time (in CPU seconds) of the BC algorithm on 1-neighborhood Erdös networks.

	ERDOS-97-1	ERDOS-98-1	ERDOS-99-1
1	196.246	216.433	235.121
2	242.454	294.349	327.407
3	252.445	262.061	309.323
4	154.000	172.177	191.003
5	164.063	188.906	193.905

Table 3 The reduced graph sizes and run time (in seconds) for the BC algorithm on reduced graphs.

	ERDOS-97-2			ERDOS-98-2			ERDOS-99-2		
k	$\|V'\|$	$\|E'\|$	Time	$\|V'\|$	$\|E'\|$	Time	$\|V'\|$	$\|E'\|$	Time
1	174	1061	8.203	188	1160	11.766	194	1208	12.234
2	174	1061	25.561	188	1160	29.28	194	1208	40.468
3	174	1061	45.999	188	1160	38.781	194	1208	38.874
4	77	510	2.453	105	686	4.562	116	763	10.624
5	77	510	3.063	105	686	4.047	116	763	6.297

belong to a maximum k-plex, therefore i can be deleted from G without changing its k-plex number. The size of reduced graph $G' = (V', E')$ resulting from recursively applying the peeling procedure is shown in Table 3, which also contains the run time of BC algorithm applied to the reduced graph. Note that this procedure was only applied to 2-neighborhood Erdös networks, which were too large to be solved directly. The solutions obtained for the corresponding 1-neighborhood Erdös network was used as the lower bound in the peeling procedure for a 2-neighborhood Erdös network, *i.e.*, $\rho_k(G)$ of ERDOS-x-1.NET was used as the lower bound for $\rho_k(G)$ of ERDOS-x-2.NET.

The results of our experiments with Sanchis graphs are presented in Tables 4 and 5. The number of vertices in the generated Sanchis graphs was varied from 60 to 1000 and the edge density(d) was varied from 0.4 to 0.9. The number of edges was calculated as $\lfloor \frac{dn(n-1)}{2} \rfloor$, where $\lfloor a \rfloor$ is the largest integer less than or equal to a. The MaxGlobalCuts parameter in CPLEX was set at 200 for these experiments. In order to ensure a graceful termination of CPLEX, upper limits were set on runtime (8hrs) and working memory (2GB) when the instance cannot be solved optimally within these limits. It should also be noted that the memory limit was never reached as CPLEX *node file* option was used whereby the tree is compressed and written to disk without significant increase in runtime [30]. † and ‡ indicate non-optimal termination with gap greater than 5% and less than 5%, respectively. We were able to solve maximum 1-plex to optimality on graphs of order up to 1000 and density up to 0.7 except for the instance with 900 vertices and density 0.7 which terminated with a gap of 2.33%. On graphs of density 0.8, we were able to optimally solve up to 600 vertices and the 800 vertex instance. While the 700 vertex instance terminated with a gap of 0.855%, the 900 and 1000 vertex instances terminated with a gap 8.33% and 10.78% respectively. On instances with density 0.9, optimal solution was obtained for graphs with up to 500 vertices. The instances with order 600, 700, 800, 900 and 1000 had termination gaps of 8%, 11.97%, 13.11%, 17% and 17.96%, respectively. Recall that $\rho_1(G) = \omega(G) = \lceil \frac{n}{3} \rceil$ for these instances. In Table 4, the best clique size found is indicated in parentheses whenever we failed to solve the instance to optimality. Observe that whenever the upper limit on time was reached with a termination gap of more than 5% on a particular order and density graph, it also happened with graphs of higher order of that density. Based on this, whenever we observed CPLEX reach the upper time limit for a particular order and density with a termination gap of more than 5%, the runs for that density in higher order graphs were not conducted for the maximum 2-plex problem (marked by "-" in Table 5).

We were able to solve the maximum 2-plex problem to optimality on graphs of order up to 1000 on density 0.4 and 0.5, and up to order 600 on density 0.6. Instances of density 0.6 and order 700, 800 and 900 terminated with a gap of 5.13%, 8.61% and 13% respectively. Graphs of density 0.7 were solved to optimality up to order 500, while 600 and 700 vertex instances terminated with a gap of 1.5% and 7.27% respectively. Graphs of density 0.8 were solved up to order 250 optimally with 300 and 350 vertex instances terminating at 3% and 5.13% gaps respectively. With density at 0.9, we were able to solve only two small instances (60 and 80 vertices) optimally. Instances with 100, 120 and 140 vertices terminated with gaps of 18.42%, 23.36% and 29.17% respectively. Table 5 presents the runtime and Table 6 presents the 2-plex number (or best integer solution) found for each instance.

Ability to solve maximum 2-plex drastically reduced and the instances solved optimally required much higher running times as compared to maximum 1-plex. In a head-to-head comparison, 24 instances on which maximum 1-plex was optimally solved for, maximum 2-plex was not. Besides the obvious reason that the set of feasible solutions is larger for the maximum 2-plex problem, there is presently no other explanation for this behavior.

Given the intractability of the problems solved, these results are quite encouraging and indicate that the MIS cuts can be used effectively to solve this problem. Development of a more poverful MIS-integrated BC algorithm is a direction that needs further exploration in order to be able solve large-scale instances. Even for the instances presently solved optimally, the running times can certainly be improved by fine tuning, or even integrating other cuts in the BC algorithm and by using more powerful computational resources. But the improvement may be limited as the problem is basically intractable. However, these numerical results should indicate where we can expect computational difficulties while solving these problems.

Using the same experimental set up as on the Sanchis test-bed, we solved the maximum 2-plex problem on selected DIMACS clique instances [23]. However, the upper limit on runtime was increased to 24hrs and MaxGlobalCuts parameter was reduced to 50 for these runs. Table 7 presents the outcome of this experiment. Columns G, n, m, d and $\omega(G)$ provide the DIMACS graph name, number of vertices, number of edges, edge density and clique number, respectively. Columns BIS, UB and Time provide the best integer solution found by the BC algorithm, the best computed upper bound

Table 4 Runtime in seconds for solving maximum 1-plex problem on Sanchis graphs of different order (n) and density (d). Best solution found for non-optimally terminated instances are shown in parentheses, all of them actually correspond to the optimal solution.

n	$d = 0.4$	$d = 0.5$	$d = 0.6$	$d = 0.7$	$d = 0.8$	$d = 0.9$
60	0.515	0.14	0.109	0.094	0.171	0.125
80	0.953	0.281	0.188	0.328	0.344	0.625
100	1.656	1.015	0.374	0.718	0.313	0.61
120	4.11	1.781	1.515	1.5	1.046	1.875
140	6.782	2.734	1.891	1.75	1.469	2.032
160	9.391	4.094	2.843	2.375	1.954	5.266
180	14.625	6.562	4.843	3.765	2.735	3.766
200	26.312	11.234	6.703	4.89	4.938	6.907
250	53.609	27.672	15.375	12.563	8.89	29.328
300	130.311	49.828	36.844	49.218	31.578	58.094
350	167.64	69.874	80.093	95.046	91.89	311.734
400	305.576	146.405	123.953	145.296	298.891	3164.52
500	587.809	332.216	827.541	2640.28	2302.89	18802.8
600	1440.16	666.262	1265.4	2445.75	6324.92	28801† (200)
700	2084.06	1313.13	3128.04	1163.8	28801.1‡ (234)	28801.1† (234)
800	4188.82	2407.17	5537.67	9497.54	28202.3	28801.1† (267)
900	7484.71	7293.83	3138.48	28803‡ (300)	28802.5† (300)	28801† (300)
1000	10609.8	12619.5	5454.63	14436.1	28803.1† (334)	28800.9† (334)

Table 5 Runtime in seconds for solving maximum 2-plex problem on Sanchis graphs of different order (n) and density (d).

n	$d = 0.4$	$d = 0.5$	$d = 0.6$	$d = 0.7$	$d = 0.8$	$d = 0.9$
60	0.781	0.438	1.031	4.592	125.295	141.635
80	1.422	1.14	1.515	5.592	182.749	28298
100	3.108	2.172	2.329	6.89	249.187	28801†
120	6.375	3.687	4.422	10.687	815.7	28801.1†
140	9.812	6	10	37.938	1775.66	28801†
160	16.265	9.828	21.156	45.843	1623.88	-
180	26.093	22.406	29.14	50.234	8458.37	-
200	28.453	29.953	57.281	152.749	1687.75	-
250	66.984	59.219	131.311	709.823	14062.5	-
300	131.39	138.921	314.373	1204.26	28802.5‡	-
350	255.012	174.875	638.072	2400.58	28803†	-
400	321.754	483.749	1085.99	3957.04	-	-
500	1493.74	1130.23	3728.15	15078.9	-	-
600	2594.98	2802.97	7990.93	28820.9‡	-	-
700	2451.03	4929.64	28865.5†	28801.9†	-	-
800	4571.52	6961.75	28901†	-	-	-
900	5643.9	17661.6	28941.7†	-	-	-
1000	15307.2	22162.1	-	-	-	-

Table 6 2-plex numbers of Sanchis graphs of different order (n) and density (d).

n	$d = 0.4$	$d = 0.5$	$d = 0.6$	$d = 0.7$	$d = 0.8$	$d = 0.9$
60	20	20	20	20	21	28
80	27	27	27	27	27	33
100	34	34	34	34	34	≥ 38
120	40	40	40	40	40	≥ 43
140	47	47	47	47	47	≥ 48
160	54	54	54	54	54	-
180	60	60	60	60	60	-
200	67	67	67	67	67	-
250	84	84	84	84	84	-
300	100	100	100	100	≥ 100	-
350	117	117	117	117	≥ 117	-
400	134	134	134	134	-	-
500	167	167	167	167	-	-
600	200	200	200	≥ 200	-	-
700	234	234	≥ 234	≥ 234	-	-
800	267	267	≥ 267	-	-	-
900	300	300	≥ 300	-	-	-
1000	334	334	-	-	-	-

on the optimum and running time in seconds respectively. The experiments were also repeated for $k = 1$, results of which are presented in the next section, where we specifically consider the maximum clique problem.

Table 7 Maximum-2-plex problem on DIMACS graphs.

Graph details							
G	n	m	d	$\omega(G)$	BIS	UB	Time
c-fat200-1	200	1534	0.077	12	12	12	57.285
c-fat200-2	200	3235	0.163	24	24	24	46.861
c-fat200-5	200	8473	0.426	58	58	58	40.235
hamming6-2	64	1824	0.905	32	32	32	0.469
hamming6-4	64	704	0.349	4	6	6	4.391
hamming8-2	256	31616	0.969	128	128	130.395	86402.2
hamming8-4	256	20864	0.639	16	16	46.694	86401.6
johnson8-2-4	28	210	0.556	4	5	5	3.611
johnson8-4-4	70	1855	0.768	14	14	14	7423.78
johnson16-2-4	120	5460	0.765	8	10	14.963	86400.8
keller4	171	9435	0.649	11	15	26.8387	86401.7
MANN_a9	45	918	0.927	16	26	26	2.344
san200_0.7_2	200	13930	0.7	18	26	32.1343	86400.8
san200_0.9_3	200	17910	0.9	44	50	79.9747	86405.5

5.3 On Maximum Clique Problem

Consider the following 0,1-formulation of the maximum clique problem on $G = (V, E)$. Let $\bar{d}_i = |V \setminus N[i]|$ as before.

$$\omega(G) = \max \sum_{i \in V} x_i \tag{9}$$

subject to:

$$\sum_{j \in V \setminus N[i]} x_j \le \bar{d}_i (1 - x_i) \qquad \forall\ i \in V \tag{10}$$

$$x_i \in \{0, 1\} \qquad \forall\ i \in V \tag{11}$$

This formulation is a special case of the maximum-k-plex formulation (2) - (4) presented before when $k = 1$ as $\rho_1(G) = \omega(G)$. Constraint (10) ensures that no non-neighbor of a vertex is included in the clique for every vertex in the clique and becomes redundant for others. We will refer to the formulation (9) - (11) of the maximum clique problem as the *1-plex formulation*.

This case was excluded in the previous polyhedral study for the following reason. Unlike the maximum-k-plex problem for $1 < k < n$, $x_i \le 1$ is not always a facet for the maximum clique problem. It induces a facet *if and only if* $V = N[i]$. Since when the condition is satisfied, there exist n affinely independent vectors e_i and $e_i + e_j$ for all $j \in N(i) = V \setminus \{i\}$, that lie on the face defined by $x_i \le 1$, and when the condition is not satisfied, there exists a $j \in V \setminus N[i]$ such that $x_i + x_j \le 1$ is valid for the clique polytope. In other words $x_i \le 1$ is a facet if and only if $\{i\}$ is a maximal independent set. This is just a special case of a classical result presented in [41].

Another interesting observation is that, to the best of our knowledge, this is the most compact integer programming formulation of the maximum clique problem with exactly n variables and n constraints. The classical (complement) *edge formulation* (12) - (14), has n variables and $|\bar{E}|$ constraints which could be $O(n^2)$ in the worst case.

$$\omega(G) = \max \sum_{i \in V} x_i \tag{12}$$

subject to:

$$x_i + x_j \le 1 \qquad \forall\ (i, j) \in \bar{E} \tag{13}$$

$$x_i \in \{0, 1\} \qquad \forall\ i \in V \tag{14}$$

The other well known *MIS formulation* using maximal independent sets, where constraint (13) is replaced by (15) has n variables and $O(3^n)$ constraints in the worst case [39].

$$\sum_{i \in I} x_i \le 1 \qquad \text{for each MIS, } I \text{ in } G \tag{15}$$

However the compactness of the 1-plex formulation comes at a price which we will make clear now. The edge formulation is closely related to the 1-plex formulation in the following sense. Note that constraint (13) can be rewritten as

$$x_i + x_j \leq 1 \qquad \forall\, i \in V,\ j \in V \setminus N[i], \tag{16}$$

which amounts to repeating constraints in the edge formulation. Now it is easy to see that constraint (10) in the 1-plex formulation can be obtained by summing constraint (16) over all $j \in V \setminus N[i]$, for each $i \in V$, which results in n constraints. Let $C(G)$ denote the feasible binary vectors of either formulation (characteristic vectors of cliques in G), $P(G)$ denote the LP relaxation polytope of the edge formulation and let $R(G)$ denote the LP relaxation polytope of the 1-plex formulation. Then we have, $conv(C(G)) \subseteq P(G) \subseteq R(G)$ since $R(G)$ is defined by a surrogate system [27] of $P(G)$. Thus, we solve a poorer relaxation in the nodes of the BC tree when we use the 1-plex formulation to solve the maximum clique problem. However, the run times for 1-plex formulation were on many instances better than for the edge formulation, as demonstrated by the results in Table 8. This suggests that further investigation into the 1-plex formulation for the maximum clique problem could provide interesting new insights into this well researched problem.

Table 8 Maximum clique problem on DIMACS graphs.

Graph details					Time for	Time for
G	n	m	d	$\omega(G)$	1-plex form	edge form.
c-fat200-1	200	1534	0.077	12	17.969	50.562
c-fat200-2	200	3235	0.163	24	30.157	67.344
c-fat200-5	200	8473	0.426	58	15.797	31.687
hamming6-2	64	1824	0.905	32	0.015	0.016
hamming6-4	64	704	0.349	4	0.265	0.485
hamming8-2	256	31616	0.969	128	0.031	0.031
hamming8-4	256	20864	0.639	16	2089.69	1046.06
johnson8-2-4	28	210	0.556	4	0.031	0.016
johnson8-4-4	70	1855	0.768	14	0.781	0.031
johnson16-2-4	120	5460	0.765	8	0.297	0.047
keller4	171	9435	0.649	11	315.574	80.969
MANN_a9	45	918	0.927	16	0.203	0.032
san200_0.7_2	200	13930	0.7	18	23.531	25.641
san200_0.9_3	200	17910	0.9	44	4725.49	23.922

6 Conclusion and Future work

This paper studies the maximum k-plex problem which is a graph-theoretic relaxation of the maximum clique problem introduced by [47] in the context of social networks. Some important properties of a k-plex are reviewed and a detailed introduction to clique relaxation models in social network analysis is presented. We prove complexity results establishing the intractability of this problem for every constant k. The problem is formulated as a binary integer program and basic polyhedral results are presented. In addition, classes of valid inequalities are developed for the problem, one of which is implemented in a basic branch-and-cut framework. Results of our computational experiments are presented which indicate the effectiveness of the cuts used. Further, the formulation developed is studied in the context of the maximum clique problem for which this work has led to a new compact formulation.

Several research problems and directions have been identified through the course of the paper that need attention. To summarize, facets of the k-plex polytope need to be discovered, the branch-and-cut algorithm needs to be modified and tuned to be able to solve larger instances to optimality and finally the new formulation for the maximum clique problem needs to be explored further. Development of meta-heuristics capable of solving massive instances in a reasonable amount of time would be of great practical value.

References

1. Abello, J., Pardalos, P., Resende, M.: On maximum clique problems in very large graphs. In: J. Abello, J. Vitter (eds.) External memory algorithms and visualization, *DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, vol. 50, pp. 119–130. American Mathematical Society (1999)
2. Alba, R.: A graph-theoretic definition of a sociometric clique. Journal of Mathematical Sociology **3**, 113–126 (1973)
3. Applegate, D., Bixby, R., Chvátal, V., Cook, W.: On the solution of traveling salesman problems. Documenta Mathematica Journal der Deutschen Mathematiker-Vereinigung, International Congress of Mathematicians pp. 645–656 (1998)
4. Balas, E., Ceria, S., Cornuéjols, G.: Mixed 0-1 programming by lift-and-project in a branch-and-cut framework. Management Science **42**, 1229–1246 (1996)
5. Balas, E., Ceria, S., Cornuéjols, G., Natraj, N.: Gomory cuts revisited. Operations Research Letters **19**, 1–9 (1996)
6. Balas, E., Ceria, S., Cornuejols, G., Pataki, G.: Polyhedral methods for the maximum clique problem. In: D.S. Johnson, M.A. Trick (eds.) Cliques, Coloring, and Satisfiability: Second DIMACS Challenge, DIMACS Series in Discrete Mathematics and Theoretical Computer Science 26, pp. 11–28. American Mathematical Society, Providence, RI (1996)
7. Balasundaram, B., Butenko, S., Trukhanov, S.: Novel approaches for analyzing biological networks. Journal of Combinatorial Optimization **10**, 23–39 (2005)
8. Batagelj, V.: Networks/Pajek Graph Files. http://vlado.fmf.uni-lj.si/pub/networks/pajek/data/gphs.htm. Accessed January 2005
9. Berry, N., Ko, T., Moy, T., Smrcka, J., Turnley, J., Wu, B.: Emergent clique formation in terrorist recruitment. The AAAI-04 Workshop on Agent Organizations: Theory and Practice, July 25, 2004, San Jose, California (2004). Http://www.cs.uu.nl/ virginia/aotp/papers.htm
10. Boginski, V., Butenko, S., Pardalos, P.: Network models of massive datasets. Computer Science and Information Systems **1**, 79–93 (2004)
11. Boginski, V., Butenko, S., Pardalos, P.: Mining market data: a network approach. Computers & Operations Research (2005). In press, available online at http://www.sciencedirect.com/
12. Boginski, V., Butenko, S., Pardalos, P.: Statistical analysis of financial networks. Computational Statistics & Data Analysis **48**, 431–443 (2005)
13. Bomze, I.M., Budinich, M., Pardalos, P.M., Pelillo, M.: The maximum clique problem. In: D.Z. Du, P.M. Pardalos (eds.) Handbook of Combinatorial Optimization, pp. 1–74. Kluwer Academic Publishers, Dordrecht, The Netherlands (1999)
14. Bourjolly, J.M., Laporte, G., Pesant, G.: Heuristics for finding k-clubs in an undirected graph. Computers & Operations Research **27**, 559–569 (2000)

15. Bourjolly, J.M., Laporte, G., Pesant, G.: An exact algorithm for the maximum k-club problem in an undirected graph. European Journal Of Operational Research **138**, 21–28 (2002)
16. Butenko, S., Wilhelm, W.: Clique-detection models in computational biochemistry and genomics. European Journal of Operational Research (2006). To appear. Available online at http://www.sciencedirect.com/
17. Chen, H., Chung, W., Xu, J.J., Wang, G., Qin, Y., Chau, M.: Crime data mining: A general framework and some examples. Computer **37**(4), 50–56 (2004)
18. Chen, Y.P., Liestman, A.L., Liu, J.: Clustering algorithms for ad hoc wireless networks. In: Y. Pan, Y. Xiao (eds.) Ad Hoc and Sensor Networks. Nova Science Publishers (2004). To be published
19. Cook, D.J., Holder, L.B.: Graph-based data mining. IEEE Intelligent Systems **15**(2), 32–41 (2000)
20. Cornuejols, G.: Combinatorial Optimization: Packing and Covering. CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM, Philadelphia (2001)
21. Davis, R.H.: Social network analysis: An aid in conspiracy investigations. FBI Law Enforcement Bulletin pp. 11–19 (1981)
22. Dekker, A.H.: Social network analysis in military headquarters using CAVALIER. Proceedings of Fifth International Command and Control Research and Technology Symposium pp. 24–26 (2000)
23. DIMACS: Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge. http://dimacs.rutgers.edu/Challenges/ (1995). Accessed September 2005
24. Fischer, I., Meinl, T.: Graph based molecular data mining - an overview. In: W. Thissen, P. Wieringa, M. Pantic, M. Ludema (eds.) IEEE SMC 2004 Conference Proceedings, pp. 4578–4582 (2004)
25. Freeman, L.C.: The sociological concept of "group": An empirical test of two models. American Journal of Sociology **98**, 152–166 (1992)
26. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-completeness. W.H. Freeman and Company, New York (1979)
27. Glover, F.: Tutorial on surrogate constraint approaches for optimization in graphs. Journal of Heuristics **9**, 175–227 (2003)
28. Grossman, J., Ion, P., Castro, R.D.: The Erdös Number Project. http://www.oakland.edu/enp/. Accessed January 2005
29. Harary, F.: Graph Theory. Narosa Publishing House, New Delhi (1988)
30. ILOG CPLEX. http://www.ilog.com/products/cplex/. Accessed September 2005
31. Johnson, D., Trick, M. (eds.): Cliques, Coloring, and Satisfiablility: Second DIMACS Implementation Challenge, October 11-13, 1993, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 26. AMS, Providence, RI (1996)
32. Krebs, V.: Mapping networks of terrorist cells. Connections **24**, 45–52 (2002)
33. Krishna, P., Vaidya, N., Chatterjee, M., Pradhan, D.: A cluster-based approach for routing in dynamic networks. In: ACM SIGCOMM Computer Communication Review, pp. 49–65 (1997)
34. Luce, R.: Connectivity and generalized cliques in sociometric group structure. Psychometrika **15**, 169–190 (1950)
35. Luce, R., Perry, A.: A method of matrix analysis of group structure. Psychometrika **14**, 95–116 (1949)
36. McAndrew, D.: The structural analysis of criminal networks. In: D. Canter, L. Alison (eds.) The Social Psychology of Crime: Groups, Teams, and Networks, Offender Profiling Series, III. Aldershot, Dartmouth (1999)
37. Mitchell, J.: Branch-and-cut algorithms for combinatorial optimization problems. In: P.M. Pardalos, M.G.C. Resende (eds.) Handbook of Applied Optimization, pp. 65–77. Oxford University Press, New York (2002)
38. Mokken, R.: Cliques, clubs and clans. Quality and Quantity **13**, 161–173 (1979)
39. Moon, J.W., Moser, L.: On cliques in graphs. Israel Journal of Mathematics **3**, 23–28 (1965)

40. Mukherjee, M., Holder, L.B.: Graph-based data mining on social networks. In: Workshop on Link Analysis and Group Detection (LinkKDD2004) (2004)
41. Padberg, M.W.: On the facial structure of set packing polyhedra. Mathematical Programming **5**, 199–215 (1973)
42. Padberg, M.W., Rinaldi, G.: A branch-and-cut algorithm for the resolution of large-scale symmetric travelling salesman problems. SIAM Review **33**, 60–100 (1991)
43. Sageman, M.: Understanding Terrorist Networks. University of Pennsylvania Press (2004)
44. Sanchis, L.A., Jagota, A.: Some experimental and theoretical results on test case generators for the maximum clique problem. INFORMS Journal on Computing **8**(2), 103–117 (1996)
45. Scott, J.: Social Network Analysis: A Handbook, 2 edn. Sage Publications, London (2000)
46. Seidman, S.B.: Network structure and minimum degree. Social Networks **5**, 269–287 (1983)
47. Seidman, S.B., Foster, B.L.: A graph theoretic generalization of the clique concept. Journal of Mathematical Sociology **6**, 139–154 (1978)
48. Stojmenovic, I. (ed.): Handbook of Wireless Networks and Mobile Computing. Wiley InterScience (2002)
49. Terveen, L., Hill, W., Amento, B.: Constructing, organizing, and visualizing collections of topically related web resources. ACM Transactions on Computer-Human Interaction **6**, 67–94 (1999)
50. Trotter, L.E.: A class of facet producing graphs for vertex packing polyhedra. Discrete Mathematics **12**, 373–388 (1975)
51. Washio, T., Motoda, H.: State of the art of graph-based data mining. SIGKDD Explor. Newsl. **5**(1), 59–68 (2003)
52. Wasserman, S., Faust, K.: Social Network Analysis. Cambridge University Press (1994)