European Journal of Operational Research 210 (2011) 489–494

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

Upper bounds and heuristics for the 2-club problem

Filipa D. Carvalho [a,b], M. Teresa Almeida [a,b,*]

[a] ISEG, Universidade Técnica de Lisboa, Rua do Quelhas 6, 1200-781 Lisboa, Portugal
[b] CIO, FC, Universidade de Lisboa, Bloco C6, Piso 4, 1749-016 Lisboa, Portugal

ARTICLE INFO

Article history:
Received 5 January 2010
Accepted 18 November 2010
Available online 25 November 2010

Keywords:
Combinatorial optimization
Integer programming
k-club problem
Heuristics

ABSTRACT

Given an undirected graph $G = (V,E)$, a k-club is a subset of V that induces a subgraph of diameter at most k. The k-club problem is that of finding the maximum cardinality k-club in G. In this paper we present valid inequalities for the 2-club polytope and derive conditions for them to define facets. These inequalities are the basis of a strengthened formulation for the 2-club problem and a cutting plane algorithm. The LP relaxation of the strengthened formulation is used to compute upper bounds on the problem's optimum and to guide the generation of near-optimal solutions. Numerical experiments indicate that this approach is quite effective in terms of solution quality and speed, especially for low density graphs.

1. Introduction

Let $G = (V,E)$ be an undirected graph. For each pair of nodes i and j, the distance $dist_G(i,j)$ is the minimum number of edges needed to link i and j in G. The maximum distance between any pair of nodes in G is called the diameter of G. The subgraph induced in G by a subset of nodes S is $G_{[S]} = (S,E(S))$, where $E(S)$ is the set of edges of E with both end nodes in S. If $G_{[S]}$ has diameter at most k, S is a k-club in $G(k \in \mathbb{N})$. For $k = 1$, a k-club is simply a clique, Bomze et al. (1999), Alidaee et al. (2007), Martins (2010). For $k > 1$, a k-club and a k-clique have different structures: in a k-clique every pair of nodes must be linked by a chain in G with at most k edges, but the chain may include nodes not in the k-clique whereas in a k-club all nodes in the chain must be in the k-club. The k-club problem consists of finding a maximum cardinality k-club in a given graph G.

For small values of k, large cardinality k-clubs represent dense structures in networks. They have been used by social scientists to identify cohesive groups of actors in social networks that represent individuals, companies, market areas or other entities. A discussion of k-clubs and k-cliques in the context of social networks can be found in Alba (1973), Mokken (1979). These structures have also been used by biologists to study protein interactions, Balasundaram et al. (2005). For more on applications of k-clubs and other clique-related problems see Wasserman and Faust (1994), Balasundaram et al. (2009), Boginski et al. (2006), Butenko and Wilhelm (2006). For a review of combinatorial optimization contributions in modelling and solving biology problems see Blażewicz et al. (2005).

The k-club problem has been addressed by several authors. Bourjolly et al. (2000) first derived some properties of k-clubs and used them to develop heuristic algorithms. These algorithms will be reviewed in Section 4. In a second paper, Bourjolly et al. (2002), they showed by reduction from the clique problem that the k-club is NP-hard for any $k > 1$, and they developed a branch-and-bound algorithm for its solution. Balasundaram et al. (2005) derived results for the 2-club polytope P_{2c}, which will be reviewed in Section 2.

In this paper we derive new inequalities needed for the description of P_{2c} and show that they can contribute to decreasing the LP upper bound on the optimum of the 2-club problem as well as guide the generation of approximate solutions.

2. Integer programming formulations for the 2-club problem

A pair of nodes i and j may belong to a 2-club in $G = (V,E)$ if and only if they are linked by an edge or they are both linked to another node also in the 2-club.

Let us associate with each node $i \in V$ a binary variable x_i such that $x_i = 1$ if and only if node i is included in the 2-club and let us denote by N_i the set of nodes linked to node i in G, *i.e.* the set of its neighbours.

For each pair of nodes i and j, the following constraints must hold in any 2-club:

* Corresponding author at: Instituto Superior de Economia e Gestão, Departamento de Matemática, Rua do Quelhas 6, 1200-781 Lisboa, Portugal. Tel.: +351 213 925 800; fax: +351 213 922 781.
E-mail addresses: filipadc@iseg.utl.pt (F.D. Carvalho), talmeida@iseg.utl.pt (M.T. Almeida).

0377-2217/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2010.11.023

$$x_i + x_j \leqslant 1 \quad \text{if} \quad dist_G(i,j) > 2, \tag{1}$$

$$x_i + x_j - \sum_{r \in (N_i \cap N_j)} x_r \leqslant 1 \quad \text{if} \quad (i,j) \notin E, \quad dist_G(i,j) \leqslant 2. \tag{2}$$

A maximum cardinality 2-club in G may be found by solving the integer programming problem, Bourjolly et al. (2002):

$$(P)\ max\left\{\sum_{i \in V} x_i : (1)\&(2) \text{ hold and } x_i \in \{0,1\}, \forall i \in V\right\}.$$

Constraints (1) will be called *packing constraints*, and constraints (2) will be called *neighbourhood constraints*. The optimum of (P) will be represented by $Z(P)$, its linear programming relaxation will be called (P_{LP}), and the linear optimum denoted by $Z(P_{LP})$.

Setting all variables $x_i = 0.5$ yields a feasible solution of (P_{LP}). Therefore $Z(P_{LP}) \geqslant 0.5|V|$. This means that the gap $Z(P_{LP}) - Z(P)$ may be quite large and that tighter formulations are needed to solve the 2-club problem using LP methods, especially in low density graphs. Balasundaram et al. (2005) made a first contribution in this direction, showing that:

(a) The dimension of the 2-club polytope P_{2c} is $|V|$;
(b) $x_i \geqslant 0$ defines a facet of P_{2c} for every $i \in V$;
(c) For an arbitrary $i \in V$, $x_i \leqslant 1$ defines a facet of P_{2c} if and only if $dist_G(i,j) \leqslant 2$ for all $j \in V$;
(d) If $I \subseteq V$ is a 2-independent set in G (*i.e.* $dist_G(i,j) > 2$, $\forall i,j \in I$), then the inequality $\sum_{i \in I} x_i \leqslant 1$ is valid for P_{2c} and is facet defining if I is a maximal set.

Inequalities in (d) will be called *BBT inequalities* in the remainder of the paper.

For more on polyhedral theory, readers are referred to Nemhauser and Wolsey (1988). Polyhedral studies of clique problems are presented in Sorensen (2004), Macambira and Souza (2000), Park et al. (1996).

2.1. Strong formulation (SP)

In general, *packing* and *neighbourhood constraints* are redundant in the description of the 2-club polytope P_{2c}. Replacing redundant constraints with non-redundant constraints results in a better formulation from the LP point of view. We shall first characterize *packing* and *neighbourhood constraints* that are necessary for the description of P_{2c} and show how to lift those that are redundant in order to obtain facet defining inequalities.

Each *packing constraint* is associated with two nodes a and b such that $dist_G(a,b) > 2$.

Any 2-independent set $I \subseteq V$ such that $\{a,b\} \subset I$ can be used to generate a *BBT inequality* $\sum_{i \in I} x_i \leqslant 1$ that lifts the *packing constraint* $x_a + x_b \leqslant 1$. As already mentioned, Balasundaram et al. (2005) showed that if I is a maximal set, $\sum_{i \in I} x_i \leqslant 1$ defines a facet of P_{2c}. Therefore, a *packing constraint* $x_a + x_b \leqslant 1$ defines a facet of P_{2c} if and only if $\min\{dist_G(i,a), dist_G(i,b)\} \leqslant 2$ for all $i \in V \backslash \{a,b\}$. Otherwise, it is dominated by constraints of the form $\sum_{i \in I} x_i \leqslant 1$, with $I \supset \{a,b\}$, which define facets of P_{2c} if I is a maximal 2-independent set.

Each *neighbourhood constraint* is associated with two nodes a and b such that $dist_G(a,b) = 2$. For any node $i \in V \backslash \{a,b\}$ such that $\min\{dist_G(i,a), dist_G(i,b)\} > 2$, inequality $x_a + x_b + x_i - \sum_{r \in (N_a \cap N_b)} x_r \leqslant 1$ is valid for P_{2c} because neither a nor b can be included in a 2-club that includes node i. This inequality is a lifting of the *neighbourhood constraint* $x_a + x_b - \sum_{r \in (N_a \cap N_b)} x_r \leqslant 1$.

Proposition 1 generalizes this idea and gives a necessary and sufficient condition for the lifted *neighbourhood constraints* to define facets of P_{2c}.

Proposition 1. *Let a, $b \in V$ be such that $dist_G(a,b) = 2$ and let $I \subseteq V \backslash \{a,b\}$ be such that $I \cup \{a\}$ and $I \cup \{b\}$ are 2-independent sets in G. The strengthened neighbourhood inequality*

$$\sum_{i \in I_{ab}} x_i - \sum_{r \in (N_a \cap N_b)} x_r \leqslant 1, \tag{3}$$

where $I_{ab} = I \cup \{a,b\}$ is valid for P_{2c}. It defines a facet of P_{2c} if and only if I is a maximal set.

Proof. A 2-club may have at most one node of I. If it has one node of I, neither a nor b can be included in it. To include node a and node b, a 2-club must include at least one of their common neighbours.

If set I is not maximal, this inequality is dominated by the corresponding inequalities associated with maximal sets containing I. For the proof of the sufficiency condition see Carvalho and Almeida (2008). □

Therefore, a *neighbourhood constraint* $x_a + x_b - \sum_{r \in (N_a \cap N_b)} x_r \leqslant 1$ defines a facet of P_{2c} if and only if $\min\{dist_G(i,a), dist_G(i,b)\} \leqslant 2$ for all $i \in V \backslash \{a,b\}$. Otherwise, it is dominated by constraints of the form $\sum_{i \in I_{ab}} x_i - \sum_{r \in (N_a \cap N_b)} x_r \leqslant 1$, with $I_{ab} = I \cup \{a,b\}$, which define facets of P_{2c} if I is a maximal set.

Strong formulations may be built by lifting each redundant *packing constraint* to a maximal *BBT inequality* and each redundant *neighbourhood constraint* to a maximal *strengthened neighbourhood inequality*.

To obtain one of these formulations we followed an algorithmic procedure initialized with (P). To lift *packing constraints*, we considered in lexicographic order the pairs (a,b) such that $dist_G(a,b) > 2$ and $\{a,b\}$ is not a maximal 2-independent set. Following that order, at each step we generated a *BBT inequality* with a maximal 2-independent set containing $\{a,b\}$, added the inequality to the formulation, removed all dominated *packing constraints*, and deleted the corresponding pairs from the ordered list. The maximal 2-independent sets were generated with a greedy algorithm. For small density graphs, the number of *BBT inequalities* generated is much smaller than the number of *packing constraints* removed. A similar procedure was used to replace each *neighbourhood constraint* that does not define a facet of P_{2c} with a *strengthened neighbourhood inequality*.

This strong LP model will be called formulation (SP).

3. Upper bounds

By construction, $Z(P) \leqslant Z(SP_{LP}) \leqslant Z(P_{LP})$. These upper bounds on $Z(P)$ may be improved by adding new valid inequalities to (P_{LP}) and to (SP_{LP}).

3.1. New valid inequalities

Given two nodes a, $b \in V$, if $dist_G(a,b) > 2$, then at most one of them may be in a 2-club; if $dist_G(a,b) = 2$, they may both be in a 2-club provided that at least one of their common neighbours is also in the 2-club. Proposition 2 generalizes this idea for sets with three nodes and gives a necessary and sufficient condition for the new inequalities to define facets of P_{2c}.

Proposition 2. *Let $R = \{a,b,c\}$ be a set of three nodes in G such that $E(R) = \emptyset$ and let $I \subseteq V \backslash \{a,b,c\}$ be such that $I \cup \{a\}$, $I \cup \{b\}$ and $I \cup \{c\}$ are 2-independent sets in G.*

The inequality

$$\sum_{i \in I_{abc}} x_i - \sum_{t \in V} \alpha_t x_t \leqslant 1 \tag{4}$$

with $I_{abc} = I \cup \{a,b,c\}$ and

$$\alpha_t = \begin{cases} 2 & \text{if} \quad t \in (N_a \cap N_b \cap N_c), \\ 1 & \text{if} \quad t \in (N_a \cap N_b) \cup (N_a \cap N_c) \cup (N_b \cap N_c) \setminus (N_a \cap N_b \cap N_c), \\ 0 & \text{otherwise} \end{cases}$$

is valid for P_{2c}. It defines a facet of P_{2c} if and only if I is a maximal set.

Proof. The proof is similar to that of proposition 1. For the proof of the sufficiency condition see Carvalho and Almeida (2008). □

Note that, for each triple (a,b,c) such that $min\ \{dist_G(a,b), dist_G(a,c), dist_G(b,c)\} > 2$, inequality (4) can be interpreted as a *BBT inequality*. If the distance is equal to 2 for only one of the pairs (a,b), (a,c) or (b,c), inequality (4) can be interpreted as a *strengthened neighbourhood inequality* for that pair. In other cases, inequalities (4) will be called *roof inequalities*. When convenient, to distinguish *roof inequalities* associated with triples for which $dist_G(a,b) = dist_G(a,c) = dist_G(b,c) = 2$ from *roof inequalities* associated with triples for which $max\ \{dist_G(a,b), dist_G(a,c), dist_G(b,c)\} > 2$, we will call the former *roof* $^{=}$*inequalities* and the latter *roof* $^{>}$*inequalities*. Note that in *roof* $^{>}$*inequalities* $\alpha_t \in \{0,1\}$ for all t. □

3.2. Cutting plane algorithm

When the solution of (P_{LP}) or the solution of (SP_{LP}) is not integer, we try to identify facet defining violated *roof inequalities*, *strengthened neighbourhood inequalities*, and *BBT inequalities* to cut off the current fractional solution. If the search is successful, new inequalities are added and the enlarged problem is solved. The procedure is repeated until no more violated constraints are found or a stopping criterion is met. In each iteration, the search for violated constraints is performed in the following order: *roof* $^{=}$ *inequalities*, *roof* $^{>}$ *inequalities*, *strengthened neighbourhood inequalities*, then *BBT inequalities*. Within each group, the search is carried out until no violated constraint is found. The number of constraints added to the model before a new call to the LP solver is made is limited by a MAX_CUT parameter.

The separation problems can be solved by determining maximum weight cliques in an auxiliary graph (for more details, see Carvalho and Almeida (2008). As the maximum weight clique problem is NP-hard, we use a greedy algorithm.

4. Heuristics

Heuristics *Constellation* and *Drop* were proposed in Bourjolly et al. (2000). *Constellation* is a constructive heuristic that starts a k-club with a maximum degree node and its neighbours (which is a k-club for any $k > 1$) and then enlarges it iteratively, using a greedy criterion. The algorithm stops when either all nodes in V have been included in the k-club or the inclusion of a new node would result in a subgraph with diameter greater than k. For $k = 2$, *Constellation* stops at the end of the first iteration with a *star solution*. *Drop* is a destructive heuristic that starts with the whole set V and iteratively drops one node at a time until the resulting subgraph is a k-club. The nodes dropped are those that seem less likely to be part of a maximum cardinality k-club, due to their distances to the nodes in the current subgraph and to their degrees.

4.1. LP-based heuristics

The strategy adopted by LP & *Drop* is similar to that of *Drop*. The main difference is that the selection of the nodes to be dropped is guided by the LP optima of problems that result from adding to (SP) new constraints which account for the removal of nodes already dropped.

These LP problems will be generally called (SP_{drop}). In each iteration of *LP* & *Drop* a new problem (SP_{drop}) is solved. At the first iteration, (SP_{drop}) is simply the LP relaxation of (SP) with no additional constraints. If its optimal solution y^* is integer, then the set of nodes i for which $y_i^* = 1$ is a 2-club and the search for a feasible solution is completed. Otherwise, each node j in the current subgraph is assigned a penalty $q_j = \sum_{k:dist(j,k)>2} y_k^* - y_j^*$; one node with maximum penalty is dropped from the subgraph and the current (SP_{drop}) problem is enlarged with the corresponding $x_j = 0$ constraint. Then a new iteration is performed.

LP-edge & *Drop* is a heuristic that combines *LP* & *Drop* with a limited local search. At each iteration one pair of adjacent nodes in G-say k and l-is forced into the 2-club, by adding constraints $x_k = 1$ and $x_l = 1$ to (SP). If the optimal LP solution of the enlarged problem is integer, the iteration is completed; otherwise, a call is made to the routine that implements *LP* & *Drop*. The pair forced into the 2-club shall be a pair that seems likely to belong to a maximum cardinality 2-club. The selection is made based on the optimal solution of (SP_{LP}), x^*, and it follows the list of the edges of G in lexicographic order picking up the pair associated with the first edge (k,l) such that $x_k^* + x_l^* > \max\{x_i^* : i \in V\}$. The algorithm stops either after reaching the end of the edge list or upon hitting a time limit. The output of *LP-edge* & *Drop* is the largest 2-club generated during the procedure.

5. Computational experiments

All algorithms were coded in *C* and run on a 1.86 GHz PC Pentium III processor with 0.98 GB of RAM. The linear programming problems were solved with CPLEX 11.1.

The tests were performed on graphs randomly generated as described in Bourjolly et al. (2000). The generation of the graphs is controlled by two density parameters a and b $(0 \leqslant a \leqslant b \leqslant 1)$. The expected edge density is equal to $(a+b)/2$ and the node degree variance increases with $b - a$. The number of nodes ranges from $|V| = 50$ up to $|V| = 200$ and the expected density ranges from $D = 0.05$ up to $D = 0.25$. Within each dimension and expected density, 10 instances were generated with different values for a and b. The figures in Tables 1–3 are average values for 10 instances. In Table 4 the results are aggregated by density. We do not report the results for instances with $D > 0.25$ because the LP relaxation of (P) had integer optimal solution for almost all of the instances we tried.

Table 1 shows the results of the LP relaxations of (P), (SP), and the cutting plane algorithm. In the cutting plane algorithm, the stopping criterion is met whenever an integer solution is found, a proof of the solution's optimality is obtained, or an iteration yields a reduction of less than 1 in the upper bound value, with MAX_CUT = $2.5 \times |V|$. A solution is proven optimal if the upper bound value rounded down to the nearest integer is equal to the cardinality of the *star solution*.

Columns 3–6 in Table 1 contain information on the LP relaxations of models (P) and (SP). With formulation (SP), the average reductions on the number of constraints (columns 3 and 4) were approximately 43% and 12% for the instances with $D = 0.05$ and $D = 0.10$, respectively. The average improvement in the upper bound value was approximately 70% for the smallest density instances and 44% for instances with $D = 0.10$. For higher density instances the reductions were very small. When the node set V is itself a 2-club, there is no *packing constraint* in (P) and all *neighbourhood constraints* define facets of the 2-club polytope. In such cases formulations (P) and (SP) are the same. When the cardinality of a maximum 2-club is close to $|V|$, there are only a few *packing constraints* in (P) and the percentage of *neighbourhood constraints* that are facet defining for P_{2c} is very high. In such cases, the sets of constraints in (P) and in (SP) are very similar. This explains

Table 1
Upper bounds (across 10 instances).

D	$\|V\|$	# Constraints		Upper bounds				# Integer sol			
		(P)	(SP)	$Z(P_{LP})$	$Z(SP_{LP})$	$Z(P_{CUT})$	$Z(SP_{CUT})$	(P_{LP})	(SP_{LP})	(P_{CUT})	(SP_{CUT})
0.05	50	1163.90	488.00	25.00	8.74	8.69	**8.21**	0	2	0	3
	100	4699.20	2383.90	50.00	14.44	13.36	**13.21**	0	1	0	1
	150	10614.30	5922.90	75.00	21.43	**18.35**	18.44	0	0	0	0
	200	18938.10	11441.00	100.00	28.34	22.62	**22.42**	0	0	0	0
Average		8853.88	5058.95	62.50	18.24	15.76	**15.57**				
0.10	50	1105.60	685.60	25.00	11.35	11.58	**11.34**	0	3	0	3
	100	4461.50	3502.00	50.00	24.46	**23.19**	23.33	0	0	0	0
	150	10068.40	8838.00	75.00	42.56	**41.46**	41.60	0	0	0	0
	200	17970.40	16600.10	100.00	62.40	**61.46**	61.66	0	0	0	0
Average		8401.48	7406.43	62.50	35.19	**34.42**	34.48				
0.15	50	1043.80	856.70	25.00	16.80	16.67	**16.61**	0	1	0	1
	100	4216.30	4007.80	51.10	43.79	**43.63**	43.67	0	0	0	0
	150	9506.90	9417.80	84.09	81.40	**81.36**	81.37	0	0	0	0
	200	16920.70	16898.20	130.90	130.54	130.55	**130.53**	0	0	0	0
Average		7921.93	7795.13	72.77	68.13	**68.05**	**68.05**				
0.20	50	985.10	932.40	27.05	25.08	25.07	**25.05**	0	0	0	0
	100	3986.20	3969.20	68.31	68.15	**67.85**	68.15	2	2	2	2
	150	8981.10	8977.90	**128.14**	**128.14**	**128.14**	**128.14**	7	7	8	7
	200	15987.00	15987.00	**189.10**	**189.10**	**189.10**	**189.10**	10	10	10	10
Average		7484.85	7466.63	103.15	102.62	**102.54**	102.61				
0.25	50	927.00	921.60	34.68	34.61	**34.56**	**34.56**	5	5	5	5
	100	3731.00	3730.80	**91.30**	**91.30**	**91.30**	**91.30**	10	10	10	10
	150	8406.50	8406.50	**147.60**	**147.60**	**147.60**	**147.60**	10	10	10	10
	200	15023.30	15023.30	**198.70**	**198.70**	**198.70**	**198.70**	10	10	10	10
Average		7021.95	7020.55	118.07	118.05	**118.04**	**118.04**				

Table 2
2-Clubs generated by the heuristics (across 10 instances).

D	$\|V\|$	Constellation		Drop		LP & drop		LP-edge & drop	
		2-Club size	# Best	2-Club size	# Best	2-Club size	# Best	2-Club size	# Best
0.05	50	**8.00**	10	7.50	8	7.90	9	7.90	9
	100	**12.70**	10	6.90	0	11.50	6	**12.70**	10
	150	**17.60**	10	8.30	0	15.40	5	**17.60**	10
	200	**21.10**	10	8.70	0	11.10	0	**21.10**	10
Average		**14.85**		7.85		11.48		14.83	
0.10	50	**10.90**	10	7.90	0	10.30	7	**10.90**	10
	100	19.20	9	11.70	0	13.50	0	**19.30**	10
	150	**26.90**	10	15.30	0	17.00	0	23.40	2
	200	**35.70**	10	17.40	0	20.40	0	21.80	0
Average		**23.18**		13.08		15.30		18.85	
0.15	50	14.40	6	11.70	1	12.60	3	**14.70**	9
	100	27.30	1	28.30	1	32.10	2	**34.20**	10
	150	36.20	0	41.20	0	50.70	2	**56.90**	10
	200	48.20	0	74.20	0	94.30	2	**100.60**	9
Average		31.53		38.85		47.43		**51.60**	
0.20	50	17.40	0	19.50	2	21.60	2	**22.70**	10
	100	32.30	0	60.50	0	63.70	4	**65.40**	10
	150	48.90	0	124.60	0	**128.00**	10	**128.00**	10
	200	62.40	0	188.50	5	**189.10**	10	**189.10**	10
Average		40.25		98.28		100.60		**101.30**	
0.25	50	21.40	0	33.20	5	33.90	9	**34.30**	10
	100	40.20	0	90.40	5	**91.30**	10	**91.30**	10
	150	58.30	0	**147.60**	10	**147.60**	10	**147.60**	10
	200	75.60	0	**198.70**	10	**198.70**	10	**198.70**	10
Average		48.88		117.48		117.88		**117.98**	

the results obtained for $D = 0.20$ and $|V| = 200$ and for $D = 0.25$ and $|V| \in \{100, 150, 200\}$.

Columns 7 and 8 in Table 1 show the average upper bound values found by running the cutting plane procedure after solving the linear relaxation of models (P) and (SP), respectively. The cutting plane algorithm decreased the bounds obtained by (P) for the $D = 0.05$ and the $D = 0.10$ instances to approximately 25% and 50% of the initial value, respectively. It also decreased the bounds obtained by (SP) for the $D = 0.05$ instances to approximately 85% of the initial value. For the other groups of instances the cutting plane algorithm had little effect.

The values obtained by (P_{cut}) and (SP_{cut}) were similar but the number of integer solutions was higher with the strong formulation (columns 9–12).

Table 3
CPU time in seconds (across 10 instances).

D	$\|V\|$	(P_{CUT})	(SP_{CUT})	LP & drop	LP-edge & drop
0.05	50	0.70	0.50	0.20	0.10
	100	3.40	0.70	1.00	8.30
	150	31.30	7.50	3.90	136.40
	200	184.70	66.10	18.40	420.00
Average		55.03	18.70	5.88	141.20
0.10	50	0.80	0.00	0.20	0.30
	100	6.00	1.70	1.60	77.80
	150	22.60	5.10	7.50	420.00
	200	59.50	11.10	27.70	420.00
Average		22.23	4.48	9.25	229.53
0.15	50	0.60	0.40	0.30	3.00
	100	1.70	1.00	1.20	180.00
	150	3.50	1.80	4.50	420.00
	200	4.70	4.10	10.80	385.20
Average		2.63	1.83	4.20	247.05
0.20	50	0.50	0.30	0.20	5.60
	100	0.70	0.50	0.50	103.10
	150	0.60	1.00	0.60	0.40
	200	0.30	0.50	0.50	0.60
Average		0.53	0.58	0.45	27.43
0.25	50	0.20	0.30	0.30	0.70
	100	0.20	0.30	0.20	0.50
	150	0.50	0.30	0.10	0.20
	200	0.70	0.70	0.40	0.60
Average		0.40	0.40	0.25	0.50

Table 4
Average percentage deviations (across 40 instances).

D	GapB1	GapB2	GapNew	# Opt	GapNew/GapB1_B2
0.05	306.03	719.06	1.34	34	0.004
0.10	163.87	365.10	70.60	10	0.431
0.15	116.60	113.94	30.66	3	0.269
0.20	133.25	14.29	3.07	26	0.215
0.25	127.95	1.27	0.08	39	0.063

Table 2 compares the results obtained by *LP* & *Drop* and *LP-edge* & *Drop* with those obtained by *Constellation* and *Drop* for the same instances. For *LP-edge* & *Drop*, the time limit was set to 3 min for instances with $|V| \leqslant 100$ and to 7 min for the remaining instances.

Considering the whole set of instances, on average the largest 2-clubs were generated by *LP-edge* & *Drop*; the second best was *LP* & *Drop*, followed by *Drop*. *LP-edge* & *Drop* and *LP* & *Drop* generated the best solution in over 89% and 50% of the instances, respectively. The corresponding figures for *Constellation* and *Drop* were 43% and 24%, respectively. *LP* & *Drop* outperformed *Drop* in all density groups of instances.

Bourjolly et al. (2000) noted that the *star solution* (generated by *Constellation* if $k = 2$) is likely to be a maximum cardinality 2-club when the graph density is low. The results obtained in this study corroborate their statement. *Constellation* was the best heuristic for all instances with $D = 0.05$. For $D = 0.10$, it was the best for all but one instance whereas *LP-edge* & *Drop* found the best solution only for 55% of the instances. *Star solutions* had an average cardinality of around 23% above the average cardinality of the 2-clubs generated by *LP-edge* & *Drop*. For these two groups of instances, *LP-edge* & *Drop* was a close second best: on average the 2-clubs it generated fell short of the *star solution* by less than 3 nodes whereas the difference for *Drop* was almost 9 nodes. However, *star solutions* seem to be far from optimal for higher density graphs. For $D > 0.10$, *LP-edge* & *Drop* was on average the best of the four heuristics, followed by *LP* & *Drop*.

For $D = 0.15$, *LP-edge* & *Drop* generated the largest 2-club in all but two instances. While for the $|V| = 50$ group of instances *LP-edge* & *Drop* was closely followed by *Constellation*, for the other groups neither *Constellation* nor *Drop* were close to *LP-edge* & *Drop*: the *2*-clubs generated by *LP-edge* & *Drop* were on average 34%, 30%, 44%, and 45% larger than those generated by *Drop* for the $|V| = 50$, $|V| = 100$, $|V| = 150$, and $|V| = 200$ groups of instances, respectively.

For the $D = 0.20$ density group of instances, *LP-edge* & *Drop* generated the largest 2-clubs for all forty instances. *Drop* matched this result for only seven instances.

For the $D = 0.25$ density group of instances, the performances of *Drop*, *LP* & *Drop*, and *LP-edge* & *Drop* were comparable. However, as (P_{LP}) had an integer optimal solution for 35 instances in the group, their good performances have very limited practical application.

Table 3 contains CPU times. For instances with integer LP solutions, the CPU times reported in columns 3 and 4 refer to building the LP model – (P_{LP}) or (SP_{LP}) – and solving it with the LP routine of CPLEX. For the other instances, they include the time spent building and solving the LP model together with the time taken by the cutting plane algorithm. The CPU times spent solving (P_{LP}) and (SP_{LP}) are very small and are included in the figures shown in columns 3 and 4. The cutting plane algorithm is much faster when it is called after solving (SP_{LP}).

Computing times for *Constellation* and *Drop* were negligible. The CPU times for *LP* & *Drop* and for *LP-edge* & *Drop* are shown in columns 5 and 6 of Table 3. *LP* & *Drop* was extremely fast: no instance required more than 35 s. The time limits imposed on *LP-edge* & *Drop* were only hit on less than 25% of the instances. For these instances, in 80% of the cases *LP-edge* & *Drop* generated a higher cardinality solution than the other heuristics. By adopting a wiser edge list ordering, its performance could possibly be enhanced while keeping these time limits.

To obtain a global assessment of the methods proposed in this paper, we computed the gaps shown in Table 4 using the traditional formula $Gap = 100 \times (upperbound–lowerbound)/lowerbound$.

GapB1 and GapB2 refer to the methods proposed in Bourjolly et al. (2000, 2002): the upper bound is $Z(P_{LP})$ and the lower bound is the cardinality of the 2-club built by *Constellation* and by *Drop*, respectively. To compute GapNew, we considered $Z(SP_{cut})$ and the cardinality of the 2-club built by *LP-edge* & *Drop*. The best gap given by the methods proposed by Bourjolly et al. (2000, 2002) is represented by GapB1_B2 = min {GapB1, GapB2}.

Figures in column 5 show that 56% of the 2-clubs generated by *LP-edge* & *Drop* were proven optimal with the upper bound Z (SP_{cut}).

As shown in Table 2, for $D = 0.10$ *Constellation* generated the largest average cardinality 2-clubs among all four heuristic methods. The gap calculated with the results of *Constellation* and $Z(SP_{cut})$ for that group is 35.83%, and the ratio to GapB1_B2 is 0.219. Taking this ratio into account the conclusion is that the methods proposed in this paper bridged over 73% of the average gap obtained using the methods known from the literature.

6. Conclusions

This work corroborates the statement in Bourjolly et al. (2002) that graph density has a strong impact on the difficulty of the *2*-club problem. Although the results were quite different from one density group to another, our approach managed to bridge over 73% of the average gap between the linear upper bound $Z(P)$ and the cardinality of the 2-clubs generated by the heuristics *Constellation* and *Drop* in every density group.

For $D = 0.05$ and $D = 0.25$, on average, the difference between the upper bound value and the cardinality of the best 2-club found was less than 1, but for $D = 0.15$ this difference was approximately 16. We suspect that this large value is mostly due to the value of the upper bound. Due to this, we compared $Z(SP_{cut})$ with the

2-clique number. However this was of no use for our purpose because the 2-clique number was never smaller than $Z(SP_{cut})$. The comparison merely added to the evidence that solving the 2-club problem is in practice harder than solving the 2-clique problem.

Acknowledgements

The authors thank the anonymous referees for their suggestions for improving the presentation of the paper. Thanks are also due to Ann Henshall for her assistance in editing the final version of the paper.

References

Alba, R.D., 1973. A graph-theoretic definition of a sociometric clique. Journal of Mathematical Sociology 3, 113–126.

Alidaee, B., Glover, F., Kochenberger, G., Wang, H., 2007. Solving the maximum weight clique problem via unconstrained quadratic programming. European Journal of Operational Research 181, 592–597.

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

Balasundaram, B., Butenko, S., Hicks, I.V., 2009. Clique Relaxations in Social Network Analysis: The Maximum k-plex Problem. Available at <http://iem.okstate.edu/baski/files/kplex4web.pdf> (accessed December 2009).

Blażewicz, J., Formanowicz, P., Kasprak, M., 2005. Selected combinatorial problems of computational biology. European Journal of Operational Research 161, 585–597.

Boginski, V., Butenko, S., Pardalos, P.M., 2006. Mining market data: a network approach. Computers & Operations Research 33, 3171–3184.

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

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

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

Butenko, S., Wilhelm, W.E., 2006. Clique detection models in computational biochemistry and genomics. European Journal of Operational Research 173, 1–7.

Carvalho, F.D., Almeida, M.T., 2008. Strong valid inequalities for the 2-club problem. Centro de Investigação Operacional, Working Paper 2/2008. Available at <http://cio.fc.ul.pt> (accessed December 2009).

Macambira, E.M., Souza, C.C., 2000. The edge-weighted clique problem: valid inequalities facets, and polyhedral computations. European Journal of Operational Research 123, 346–371.

Martins, P., 2010. Extended and discretized formulations for the maximum clique problem. Computers & Operations Research 37, 1348–1358.

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

Nemhauser, G.L., Wolsey, L.A., 1988. Integer and Combinatorial Optimization. John Wiley, New York.

Park, K., Lee, K., Park, S., 1996. An extended formulation approach to the edge-weighted maximal clique problem. European Journal of Operational Research 95, 671–682.

Sorensen, M., 2004. New facets and branch and cut algorithm for the weighted clique problem. European Journal of Operational Research 154, 57–70.

Wasserman, S., Faust, K., 1994. Social Network Analysis. Cambridge University Press, New York.