



UNIVERSITÀ POLITECNICA DELLE MARCHE
Repository ISTITUZIONALE

LP-based dual bounds for the maximum quasi-clique problem

This is the peer reviewed version of the following article:

Original

LP-based dual bounds for the maximum quasi-clique problem / Marinelli, F.; Pizzuti, A.; Rossi, F.. - In: DISCRETE APPLIED MATHEMATICS. - ISSN 0166-218X. - STAMPA. - 296:(2021), pp. 118-140. [10.1016/j.dam.2020.02.003]

Availability:

This version is available at: 11566/293642 since: 2024-05-09T07:15:18Z

Publisher:

Published

DOI:10.1016/j.dam.2020.02.003

Terms of use:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. The use of copyrighted works requires the consent of the rights' holder (author or publisher). Works made available under a Creative Commons license or a Publisher's custom-made license can be used according to the terms and conditions contained therein. See editor's website for further information and terms and conditions.

This item was downloaded from IRIS Università Politecnica delle Marche (<https://iris.univpm.it>). When citing, please refer to the published version.

note finali coverage

(Article begins on next page)

LP-based dual bounds for the maximum quasi-clique problem

Fabrizio Marinelli^{a,*}, Andrea Pizzuti^a, Fabrizio Rossi^b

^a *Università Politecnica delle Marche, Ancona, Italy*

^b *Università degli Studi dell'Aquila, L'Aquila, Italy*

Abstract

A γ -quasi-clique is a simple and undirected graph with edge density of at least γ . Given a graph G , the maximum γ -quasi-clique problem (γ -QCP) consists of finding an induced γ -quasi-clique with the maximum number of vertices. γ -QCP generalizes the well-known maximum clique problem and its solution is useful for detecting dense subgraphs. After reviewing known integer linear programming formulations and dual bounds for γ -QCP, a new formulation obtained by decomposing star inequalities and combining edge inequalities is proposed. The model has an exponential number of variables but a linear number of constraints and its linear relaxation allows the computation by column generation of dual bounds for large and dense graphs. The connectivity of γ -quasi-cliques is also discussed and a new sufficient connectivity condition presented. An extensive computational experience shows the quality of the computed dual bounds and their performance in a branch-and-price framework, as well as the practical effectiveness of the connectivity condition.

Keywords: quasi-clique, mixed integer programming, integer reformulation

1. Introduction

A clique is a complete graph, i.e., a graph with an edge for any pair of vertices, and it is one of the basic combinatorial structures in graph theory.

*Corresponding author

Email addresses: fabrizio.marinelli@staff.univpm.it (Fabrizio Marinelli),
a.pizzuti@pm.univpm.it (Andrea Pizzuti), fabrizio.rossi@univaq.it (Fabrizio Rossi)

The *maximum clique problem* (MCP) consists of finding an induced *clique* of maximum order in a simple and undirected graph G [23]. Solutions of the MCP are meaningful, at least in principle, for a wide range of applications, e.g., social network analysis, coding theory, telecommunication and genetics. In fact, cliques express an ideal aggregation measure and are representative when it is interesting to evaluate the degree of interaction between entities. However, the search for a complete structure like a clique often prevents the discovery of similarly interesting dense subgraphs. Furthermore, graphs derived from real-world applications are generated from incomplete data and are often acquired through error-prone processes. For these reasons, several clique relaxations have been defined and the corresponding *maximum relaxed-clique problems* (MRCP-s) have been investigated [25]. Clique relaxations can be classified according to the number of relaxed properties: first-order relaxations are defined by slackening a single clique-typifying property related, for instance, to the degree (*k-core*, *k-plex*), the distance (*k-clique*, *k-club*), the density (γ -*quasi-clique*, *k-defective clique*) and the connectivity (*k-block*, *k-bundle*) [17]. Higher-order relaxations can also be considered by relaxing more than one properties at the same time, see [25].

Among discrete optimization problems deriving from clique relaxations, there is a pair of \mathcal{NP} -hard reciprocal problems: the *maximum quasi-clique problem* (γ -QCP) [26] and the *k-densest subgraph problem* (KDSP). The former asks for the maximum-order induced subgraph with edge density of at least γ of a simple undirected graph G , whereas the latter calls for the densest subgraph of G of order k . Another problem closely related to the γ -QCP is the *maximum degree-based γ -QCP* [24]. A *degree-based γ -quasi-clique* is a subgraph $H = (Q, E_Q)$ of a graph G induced by the set of vertices Q such that the degree of any vertex of H is at least $\gamma(|Q| - 1)$. It is easy to see that any degree-based γ -quasi-clique is a γ -quasi-clique but not vice-versa.

The γ -QCP is both theoretically and computationally difficult. Critical aspects (among others) lie in the lack of the *hereditary property* and in the existence of disconnected optimal quasi-cliques. Recall that a property \mathcal{P} of graphs

is an infinite class of graphs which is closed under isomorphism and it is *hereditary* if every induced subgraph of every member of \mathcal{P} is also in \mathcal{P} [5]. This implies that the inclusionwise maximality of the vertex-set of a graph satisfying a hereditary property can be tested in polynomial time. However, it is well known that γ -quasi-cliques are only *quasi-hereditary*. In fact, to obtain an induced γ -quasi-clique K from a γ -quasi-clique $H = (Q, E)$ it is sufficient to remove a vertex v from H with $\deg(v) < \frac{2 \cdot |E|}{|Q|}$ but this makes not straightforward the maximality check of non-trivial γ -quasi-cliques.

The connectivity role in γ -QCP is also not completely settled. Allowing disconnected γ -quasi-cliques clearly extends the solution space but the computational consequences have not been investigated. Note that, differently from other clique relaxations (k -plex, k -defective clique, and k -bundle) for which disconnected subgraphs can be optimal solutions only if sufficiently small, the size of a disconnected γ -QCP optimal solution is generally not limited from above [17]. However, many real applications implicitly ask to find dense connected subgraphs in order to properly capture the relations between elements of clusters, and a solution comprised of more than one connected component misses this aspect and may become less meaningful for the application (the presence of disconnected optimal solutions could be even more frequent in both the vertex-weighted and edge-weighted version of γ -QCP). Therefore, the quasi-clique definition (and similar) should include the connectivity property to some extent. Section 4 is devoted to discuss such issues.

Several heuristic approaches were proposed to solve the γ -QCP, or problems with a slightly different definition of γ -quasi-clique [4][8][16]. In [1] the authors describe a greedy randomized adaptive search procedure (GRASP) to detect maximal quasi-cliques in massive sparse graphs, where the local search phase exploits the concept of vertex potential to move up to local optima; Tsourakakis et al. [30] present two heuristics, one based on iterated elimination of the vertex with the smallest degree, and the other performing a local search looking for a sequence of induced subgraphs with non-decreasing value of density.

To the best of our knowledge, the reference exact methods for γ -QCP are

the MIP-based approach presented in [31] and the combinatorial branch-and-bound algorithm described in [22]; for KDSP instead, the state-of-the-art exact method is the enumeration scheme with dual bounds computed via semidefinite programming proposed in [18].

In this paper we develop earlier ideas originally formulated in [19]. We firstly review the main combinatorial and LP-based dual bounds for γ -QCP available in the literature. Then, we propose an integer reformulation $[D_\gamma]$ of γ -QCP and a surrogate relaxation $[D_\gamma^S]$ of $[D_\gamma]$ that provides dual bounds as good as those computed by the linear relaxation of $[D_\gamma]$.

The surrogate relaxation uses a number of constraints linear in the number of vertices of the graph and therefore it can be exploited for computing dual bounds on large and dense graphs, or even directly solved by branch-and-price. Then, we present a new sufficient condition for obtaining connected γ -quasi-cliques that dominates the previous result reported in the literature.

The outline of the paper is as follows: in Section 2 the γ -QCP is formalized, then mixed integer linear programming formulations (MILPs) and combinatorial dual bounds from the literature are reviewed; a Dantzig-Wolfe [11] reformulation and a branch-and-price algorithm are presented in Section 3; in Section 4 a new sufficient condition for solutions connectivity is given; finally, computational results are reported in Section 5 while conclusions are captured in Section 6.

2. Problem definition, MILP formulations and bounds

The γ -QCP can be formalized as follow. Let $H = (Q, E_Q) = G[Q]$ be the subgraph of $G = (V, E)$ induced by the set of vertices $Q \subseteq V$. Given $\gamma \in (0, 1]$, an optimal solution of γ -QCP is an induced subgraph H of maximum order $|Q^*|$ and with a number of edges $|E_{Q^*}| \geq \gamma \cdot \frac{|Q^*|(|Q^*|-1)}{2}$.

2.1. MILP formulations

Veremyev et al. [31] propose four MILP formulations for γ -QCP, the tightest of which, reported in the following, consists of $O(|V| + |E|)$ variables and $O(|E|)$ constraints. Let $x_i, i \in V$, and $z_e, e \in E$, be binary variables with $x_i = 1$ iff

$i \in Q$, and $z_e = 1$ iff $e \in E_Q$. Moreover, let y_k , $k \in K = \{k_L, \dots, k_U\}$, be the binary variable with $y_k = 1$ if H is of order k . The formulation reads as:

$$[C_\gamma]: \quad |Q^*| = \max \sum_{i \in V} x_i \quad (1)$$

$$z_e \leq x_i, \quad z_e \leq x_j \quad \forall e = \{i, j\} \in E \quad (2)$$

$$\sum_{i \in V} x_i \leq \sum_{k \in K} ky_k \quad (3)$$

$$\sum_{k \in K} y_k = 1 \quad (4)$$

$$\gamma \sum_{k \in K} \frac{k(k-1)}{2} y_k \leq \sum_{e \in E} z_e \quad (5)$$

$$x_i, z_e, y_k \in \{0, 1\} \quad \forall i \in V, \forall e \in E, \forall k \in K \quad (6)$$

Edge $e = \{i, j\}$ belongs to the γ -quasi-clique H if (and only if) vertex i and j are both in Q , see constraints (2). The order k of H is defined by constraints (3) and (4), and constraint (5) bounds from below the density of H by γ .

$[C_\gamma]$ can be easily modified to model other density-based clique relaxations: a formulation for the maximum s -defective clique problem can be obtained by replacing constraint (5) with

$$\sum_{k \in K} \frac{k(k-1)}{2} y_k \leq \sum_{e \in E} z_e + s$$

whereas a formulation for the maximum degree-based γ -QCP results by replacing (5) with the set of constraints

$$\gamma \left(\sum_{k \in K} ky_k - 1 \right) \leq \sum_{e=\{i,j\} \in E} z_e + \gamma(k_U - 1)(1 - x_i) \quad \forall i \in V.$$

The size of $[C_\gamma]$ grows with the density of G whereas an alternative MILP, originally presented in [26], does not. Such a formulation is obtained by linearizing the quadratic constraint that models the density threshold condition

$|E_{Q^*}| \geq \gamma \cdot \frac{|Q^*|(|Q^*|-1)}{2}$. Namely, by introducing an additional variable w_i , for each $i \in V$, such that:

$$w_i = \gamma x_i x_i + \sum_{j \in V} (a_{ij} - \gamma) x_i x_j$$

where a_{ij} is equal to one if $\{i, j\} \in E$ and zero otherwise, the γ -QCP can be formulated as follows:

$$[P_\gamma] : \quad |Q^*| = \max \sum_{i \in V} x_i \quad (7)$$

$$\sum_{i \in V} w_i \geq 0 \quad (8)$$

$$w_i \leq u_i x_i, \quad w_i \geq l_i x_i \quad \forall i \in V \quad (9)$$

$$w_i \leq \gamma x_i + \sum_{j \in V} (a_{ij} - \gamma) x_j - l_i (1 - x_i) \quad \forall i \in V \quad (10)$$

$$w_i \geq \gamma x_i + \sum_{j \in V} (a_{ij} - \gamma) x_j - u_i (1 - x_i) \quad \forall i \in V \quad (11)$$

$$x_i \in \{0, 1\}, w_i \in \mathbb{R} \quad \forall i \in V \quad (12)$$

where u_i and l_i respectively are upper and lower bounds on the value of w_i obtained by setting:

$$u_i = (1 - \gamma) \sum_{j \in V} a_{ij}, \quad l_i = -(n - 1 - \sum_{j \in V} a_{ij}) \gamma \quad \forall i \in V.$$

Formulation $[C_\gamma]$ rapidly grows due to its $O(|E|)$ constraints and therefore it is suitable for computing dual bounds only on graphs sparse enough. On the contrary, the number of variables and constraints of $[P_\gamma]$ grow linearly with $|V|$ and does not depend on graph density, but the bound provided by the linear relaxation of $[P_\gamma]$ is weaker than the one obtained by $[C_\gamma]$ (see Section 5).

A smaller formulation on dense graphs can be derived by looking at *complementary* γ -quasi-cliques, i.e., induced subgraphs $G[Q]$ with a density that does not exceed $\gamma \in [0, 1)$ [6]. Any γ -quasi-clique on G then corresponds to a complementary $(1 - \gamma)$ -quasi-clique on the complement graph $\bar{G} = (V, \bar{E})$, both

induced by the subset of vertices Q . Hence, an optimal solution of γ -QCP can be achieved by solving to optimality the maximum complementary $(1-\gamma)$ -quasi-clique problem by means of the integer program $[\bar{C}_\gamma]$ consisting of (1), (3), (4), (6) plus the following constraints:

$$x_i + x_j - 1 \leq z_e \quad \forall e = \{i, j\} \notin E \quad (13)$$

$$\sum_{e \notin E} z_e \leq (1-\gamma) \sum_{k \in K} \frac{k(k-1)}{2} y_k \quad (14)$$

The formulation is a straightforward adaptation of $[C_\gamma]$; constraints (13) is expressed for each edge in \bar{E} and ensures the presence of an edge $e = \{i, j\}$ within the complementary $(1-\gamma)$ -quasi-clique if (and only if) both endpoints i and j are selected. Finally, inequality (14) models the density restriction of a complementary $(1-\gamma)$ -quasi-clique. Formulation $[\bar{C}_\gamma]$ uses $O(|V|+|\bar{E}|)$ variables and $O(|\bar{E}|)$ constraints.

2.2. Primal and dual bounds

A lower bound k_L to $|Q^*|$ is given by the order of any clique of G . On the other hand, a basic upper bound to $|Q^*|$ is in [26]:

$$k_U = \left\lfloor \frac{1}{2} + \frac{1}{2} \sqrt{1 + \frac{8|E|}{\gamma}} \right\rfloor.$$

A better upper bound \bar{k}_U can be obtained in $O(|V| \log |V|)$ as follows, see also [22]. Any γ -quasi-clique $H = (Q, E_Q)$ fulfils by definition $|Q|(|Q| - 1) \leq 2 \frac{|E_Q|}{\gamma}$. Moreover, $|E_Q| = \sum_{i \in Q} \frac{d_i^H}{2} \leq \sum_{i \in Q} \frac{\min\{|Q|-1, d_i\}}{2}$, where d_i is the degree of i in G and d_i^H is the degree of i in H . Therefore:

$$|Q|(|Q| - 1) \leq \frac{1}{\gamma} \sum_{i \in Q} \min\{|Q| - 1, d_i\} \leq \frac{1}{\gamma} \sum_{i=1}^{|Q|} \min\{|Q| - 1, d_i\}$$

where the last inequality holds if vertices of G are sorted by non-increasing degrees, i.e., $d_i \geq d_j$ for $i < j$. It easy to see that the largest integer Q

that satisfies the above inequality is a valid upper bound \bar{k}_U for $|Q^*|$, and that $\bar{k}_U < k_U$ for the $|Q^*| < |V|$.

Let $\omega(G)$ be the clique number of graph G , i.e. the order of the maximum clique of G . If $1 - \frac{1}{\omega(G)} < \gamma$, an alternative upper bound k_U^ω on $|Q^*|$ has been defined in [26] as:

$$k_U^\omega = \left\lfloor \frac{\omega(G)\gamma}{1 - \omega(G) + \omega(G)\gamma} \right\rfloor \geq |Q^*| \quad (15)$$

Unlike the aforementioned bounds, k_U^ω does not depend directly on the order and the size of G . Figure 1 shows the value of k_U^ω with respect to $\omega(G)$ and γ . Although particularly useful when applied to sparse graphs, in which $\omega(G)$ value is limited, it rapidly becomes poor on dense graphs. For instance let us consider a graph G with $|E| = 20,000$, more than 230 vertices, and $\omega(G) = 31$. Given $\gamma = 0.97$, it results $k_U^\omega = 429$ and $k_U = 203$. Moreover, the efficacy of k_U^ω diminishes as γ gets smaller; for instance, k_U^ω with $\gamma = 0.9$ is only defined up to $\omega(G) \leq 9$. Computing $\omega(G)$ can be extremely time consuming on sufficiently large graphs, so that a reasonable choice can be to substitute $\omega(G)$ with a suitable upper bound $\omega_U(G)$ within (15). However, condition $1 - \frac{1}{\omega_U(G)} < \gamma$ must hold, limiting the applicability of the substitution, and the resulting bound degrades as $\omega_U(G)$ loses tightness.

3. Star-based reformulation

Model $[C_\gamma]$ can be reformulated by integer decomposition [11]. Let d_i be the degree of vertex $i \in V$, $N(i)$ the set of neighbours of i , and $S(i)$ the set of incidence edges of i that, for the sake of conciseness, we call the *star* of i . The *star constraint*

$$\sum_{e \in S(i)} z_e \leq (k_U - 1)x_i \quad (16)$$

is a valid inequality for the convex hull of the integer solutions to $[C_\gamma]$. Namely, the set of integer points that non-trivially satisfy (16) corresponds to the collection $\mathcal{S}_i = \{S_{i1}, S_{i2}, \dots\}$ of all the nonempty partial stars of i containing less

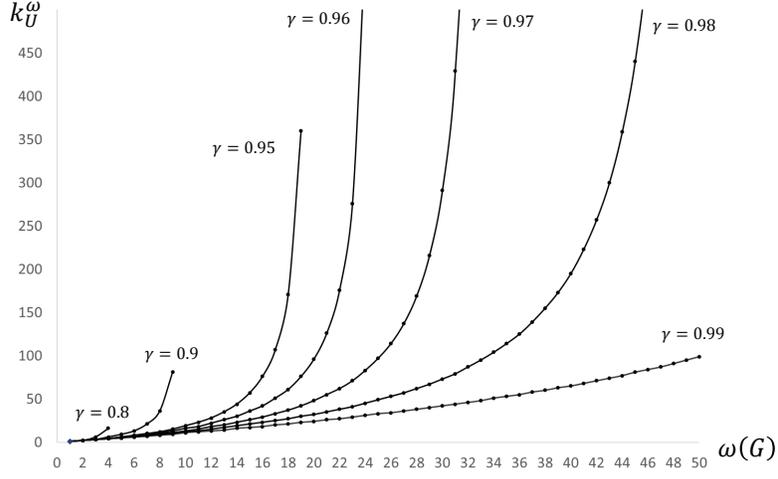


Figure 1: k_U^ω for several γ values and $\omega(G) \leq 50$

than k_U edges. Let $\mathcal{S}_i^j \subset \mathcal{S}_i$ be the collection of the partial stars including the edge $\{i, j\} \in E$. Then for any $e = \{i, j\} \in E$, the variables x_i and z_e of $[C_\gamma]$ can be rewritten as

$$x_i = \sum_{h=1}^{|\mathcal{S}_i|} \lambda_{ih} \quad z_e = \sum_{h=1}^{|\mathcal{S}_i^j|} \lambda_{ih} \quad \sum_{h=1}^{|\mathcal{S}_i|} \lambda_{ih} \leq 1 \quad \lambda_{ih} \in \{0, 1\},$$

where variable λ_{ih} is equal to 1 if the partial star \mathcal{S}_{ih} is selected, and 0 otherwise.

The resulting MIP formulation reads as follows:

$$[D_\gamma]: \quad |Q^*| = \max \sum_{i \in V} \sum_{h=1}^{|\mathcal{S}_i|} \lambda_{ih} \quad (17)$$

$$\sum_{h=1}^{|\mathcal{S}_i|} \lambda_{ih} \leq 1 \quad \forall i \in V \quad (18)$$

$$\sum_{h=1}^{|\mathcal{S}_i^j|} \lambda_{ih} - \sum_{h=1}^{|\mathcal{S}_j^i|} \lambda_{jh} = 0 \quad \forall e = \{i, j\} \in E \quad (19)$$

$$\sum_{i \in V} \sum_{h=1}^{|\mathcal{S}_i|} \lambda_{ih} \leq \sum_{k \in K} k y_k \quad (20)$$

$$\sum_{k \in K} y_k = 1 \quad (21)$$

$$\sum_{k \in K} \left\lceil \frac{\gamma k(k-1)}{2} \right\rceil y_k - \frac{1}{2} \sum_{i \in V} \sum_{h=1}^{|\mathcal{S}_i|} |\mathcal{S}_{ih}| \lambda_{ih} \leq 0 \quad (22)$$

$$\lambda_{ih} \in \{0, 1\} \quad \forall i \in V, h \in \{1, \dots, |\mathcal{S}_i|\} \quad (23)$$

$$y_k \in \{0, 1\} \quad \forall k \in K \quad (24)$$

Constraint (18) requires that at most one partial star can be selected for each vertex. Constraints (19) impose a consistent selection of partial stars, that is, if a partial star S_{ih} is chosen and S_{ih} contains the edge $\{i, j\}$, then a partial star S_{jp} including the edge $\{j, i\}$ must be selected too. To this purpose, the coefficient of the variable λ_{ih} is $+1$ for edges $\{i, j\} \in S_{ih}$ with $i < j$ and -1 otherwise. Constraints (20), (21) and (22) directly derive from $[C_\gamma]$

Figure 2 shows the coefficients matrix of constraints (18) and (19) for the *kite graph* (the kite has $|V| = 5$ and $|E| = 6$, see [7]). Gray columns describe the optimal solution for $\gamma = 0.8$, corresponding to the clique induced by vertices $\{1, 2, 3\}$. The clique is composed by overlapping partial stars S_{11} , S_{21} and S_{31} , where edges are weighted $\frac{1}{2}$ in density constraint (22).

Since each star inequality (16), together with the variable bound constraints, defines an integral polyhedron, the continuous relaxations of $[C_\gamma]$ plus star inequalities and the integer reformulation $[D_\gamma]$ have the same optimal value. Therefore, the dual bound provided by $[D_\gamma]$ is at least as tight the one pro-

Figure 2: Star reformulation on a *kite graph*, $|V| = 5$, $|E| = 6$

	S_1	S_2	S_3	S_4	S_5	
V 1	1 1 1 1 1 1					≤ 1
2		1 1 1 1 1 1				≤ 1
3			1 1 1 1 1 1			≤ 1
4				1		≤ 1
5					1 1 1	≤ 1
E {1, 2}	1 1 1	-1 -1 -1				$= 0$
{1, 3}	1 1 1		-1 -1 -1			$= 0$
{1, 4}	1 1 1			-1		$= 0$
{2, 3}		1 1 1	-1 -1 -1			$= 0$
{2, 5}		1 1 1			-1 -1	$= 0$
{3, 5}			1 1 1		-1 -1	$= 0$

vided by $[C_\gamma]$, where possible improvements arise from star constraints (16) or the rounding up of the y_k 's coefficients in (22). For instance, on a set of sparse graphs taken from [31] the effect of star constraints appears extremely limited, with a mean percentage improvement of 0.4% for $\gamma = 0.9$. When edge-weighted graphs are taken into account [28], the problem can be generalized to the weighted γ -QCP. In this case, the polyhedron described by each corresponding weighted star constraint plus variable bound constraints becomes fractional and $[D_\gamma]$ is an integer reformulation providing tighter dual bounds.

Model $[D_\gamma]$ consists of an exponential number of variables in $|E|$ and therefore the solution of its continuous relaxation requires a column generation approach. The search for useful columns (i.e., partial stars) requires to solve for each vertex i the following pricing problem. Let $\sigma_i, \delta, \psi \in \mathbb{R}^+$ and $\pi_e \in \mathbb{R}$ be the values of dual variables associated to constraints (18),(20),(22) and (19), respectively, and let w_e be a binary variable equal 1 if edge e belongs to the partial star (0 otherwise). The reduced cost of the most profitable partial star

of vertex i is:

$$\bar{c}(i) = -\sigma_i - \delta + \max \sum_{e \in S(i)} \left(\frac{1}{2}\psi + \Gamma(e)\pi_e \right) w_e \quad (25)$$

with $\Gamma(e = \{i, j\}) = -1$ if $i < j$ and $\Gamma(e) = 1$ otherwise. The pricing problem search for the set of at most $(k_U - 1)$ edges that maximize $\bar{c}(i)$ (25). Such a set can be obtained simply by ranking the edges of $S(i)$ by non decreasing values of $(\frac{1}{2}\psi + \Gamma(e)\pi_e)$.

Formulations $[C_\gamma]$ and $[D_\gamma]$ are not suitable for solving the γ -QCP on dense graphs due their $O(|E|)$ constraints, and even the continuous relaxation can be difficult to solve for moderate size instances. However, at the cost of a small loss in the dual bounds quality, a surrogate relaxation $[D_\gamma^S]$ of $[D_\gamma]$ can be considered by replacing constraints (19) with the following ones:

$$\sum_{h=1}^{|S_i|} |S_{ih}| \lambda_{ih} - \sum_{j \in N(i)} \sum_{h=1}^{|S_j^i|} \lambda_{jh} = 0 \quad \forall i \in V \quad (26)$$

Constraint (26) are obtained, for each vertex i , by summing up all constraints in (19) for $e = \{i, N(i)\}$. Hence, $[D_\gamma^S]$ has $O(|V|)$ constraints. Now, let $\theta_i \in \mathbb{R}$ be the values of the dual variable related to (26). The pricing problem for $[D_\gamma^S]$ associated to vertex i can be easily derived from (25) by properly adapting it into

$$\bar{c}(i) = 1 - \sigma_i - \delta + \max \sum_{e \in S(i)} \left(\frac{1}{2}\psi - \theta_i + \theta_j \right) w_e \quad (27)$$

3.1. A branch-and-price algorithm

Formulation $[D_\gamma^S]$ is in principle not tighter than both $[\bar{C}_\gamma]$ and $[C_\gamma]$. Nevertheless, it has the same dual bound provided by $[C_\gamma]$ (which we recall being much better than both \bar{k}_U and the linear relaxation of $[P_\gamma]$) but it is considerably smaller, having only $O(|V|)$ constraints. These features make the linear relaxation of $[D_\gamma^S]$ an interesting bound for large and/or dense instances, either embedding the column generation procedure into a combinatorial branch-and-bound scheme (like the one proposed in [22]), or considering to solve the integer program $[D_\gamma^S]$ directly by branch-and-price.

In the following, we describe a straightforward implementation of a branch-and-price algorithm that, although lacking of several performance boosting elements (acceleration and stabilization techniques, early termination and effective primal heuristics, see [12]), shows the practical viability of $[D_\gamma^S]$. The pricing strategy and the branching rule of this prototype version are the following.

As reported in Section 3, each iteration of the column generation scheme requires, for each $i \in V$, the solution of the pricing problem (27). The pricing phase can follow various strategies: one can sequentially generate all the promising columns associated with a vertex i before moving to another vertex; alternatively, one can generate for each vertex a limited subset of columns (even just one column). Preliminary tests showed that the latter strategy is more effective, though it requires a large number of iterations producing a large number of columns. To overcome this drawback, we keep the size of the master problem under a given threshold (8000 columns in our setting) by periodically deleting variables with the most negative reduced cost.

The branching is performed on the binary variables x_i ($i \in V$) of the compact formulation $[C_\gamma]$, i.e., the vertex i is either included in the γ -quasi-clique or deleted from the graph associated to the subproblem. The branching on x_i is implemented on $[D_\gamma^S]$ by modifying the i -th constraint (18) in

$$\sum_{h=1}^{|\mathcal{S}_i|} \lambda_{ih} = 1 \quad (x_i = 1)$$

and

$$\sum_{h=1}^{|\mathcal{S}_i|} \lambda_{ih} = 0 \quad (x_i = 0).$$

Consequently, the structure of the pricing problem (27) remains untouched since the changes only involve the star $S(i)$. In particular, given a subproblem P of the search tree, let F_0 be the set of vertices deleted from G in P , and F_1 the set of vertices included in the partial γ -quasi-clique associated to P . The solution of the i -th pricing problem on vertex i , must be a partial star $S^* \in \mathcal{S}_i$ such that:

- S^* does not contain edges in $S(j)$ for any $j \in F_0$;

- if $i \in F_1$, then S^* contains the edge $\{i, j\}$ for any $j \in F_1 \cap N(i)$;
- $|S^*| \leq k_U - 1 - (|F_1| - |F_1 \cap N(i)| - |F_1 \cap \{i\}|)$.

Finally, branching is performed on the vertex i with the larger number of fractional star variables λ_{ih} . The tree search is performed selecting the subproblem with the best bound, because depth-first search, although accelerates updating the current LP and its solution, has been proven ineffective.

4. Quasi-clique connectivity

In several real-world applications, finding cohesive clusters is naturally referred to the identification of single connected components on graphs [17]. In [15] a thorough discussion of approaches for the community detection problem is presented and the connectivity is assumed as a required property. Such assumption is often reasonable as disjointed clusters actually represent clusters whose mutual interaction can be assumed irrelevant with respect to the aggregation properties of interest. By contrast, solutions composed of multiple connected components are suitable or even characterizing for alternative problems, such as *clique relaxation packing problems* or *maximal clique relaxation enumeration problems*, which arise by generalizing the corresponding optimization problems originally defined on cliques [9][2].

Optimal k -core, k -defective and γ -quasi cliques can be disconnected and therefore the corresponding MRCPs should explicitly require the connectivity condition. One can argue that connectivity can be ensured by considering alternative clique relaxations, such as k -club (an induced subgraph of diameter at most k , see [3][20]) or k -block (an induced subgraph whose minimum *vertex cut* is at least k , see [10][17]), for which optimal solutions are always connected. Generally speaking however, the purpose of k -clubs and k -blocks is to guarantee respectively a given degree of reachability and robustness, whereas density based relaxations, such as k -defective and γ -quasi-cliques, are useful to deal with noisy and missing data. Moreover, although a γ -quasi-clique can be composed

by several very dense disconnected subgraphs, typically the significant γ threshold ($\gamma > 0.5$) results in solutions made of a single large connected component along with other small ones. On the other hand, it is easy to see that a γ -quasi-clique can be composed by k connected components of the same order only if $\gamma < 1/k$. This also implies that cut-edges of a connected solution generally link a large component to small ones.

We suppose that the lack of an explicit request of connectivity within formulations of MRCP can be attributed to the easiness of modeling it in mathematical programming terms. Indeed, several solution approaches can be conceived that take advantage of connectivity. For instance, in a branch-and-bound the selection of the branching variable can be done assuming the connectivity of the chased solution.

Ensuring the connectivity within MILP formulations can be done in several ways, typically by recalling variables and constraints used to model connectivity in Hamiltonian or shortest path problems [29]. Indeed, a large body of research exists on this topic, see for example [27] and the references therein, as well as the recent polyhedral study of the *connected subgraph polytope* [32].

As example, connectivity can be guaranteed for γ -QCP (and similarly for other MRCPs) by introducing variables $c_i \in \{0, 1\}$ for each vertex $i \in V$, where c_i is set to 1 iff vertex i is selected as source, and flow variables $f_{ij} \in \mathbb{R}$ for each edge $\{i, j\} \in E$. Optimal connected γ -quasi-cliques can be found by solving a MILP formulation that adds to the formulation $[C_\gamma]$ in Section 2 the following constraints:

$$\sum_{i \in V} c_i = 1 \quad (28)$$

$$c_i \leq x_i \quad \forall i \in V \quad (29)$$

$$\sum_{h \in V} x_h - 1 - k_u(1 - c_i) \leq \sum_{j \in N(i): i < j} f_{ij} - \sum_{j \in N(i): j < i} f_{ji} \quad \forall i \in V \quad (30)$$

$$\sum_{h \in V} x_h - 1 + k_u(1 - c_i) \geq \sum_{j \in N(i): i < j} f_{ij} - \sum_{j \in N(i): j < i} f_{ji} \quad \forall i \in V \quad (31)$$

$$-k_u(1 + c_i - x_i) - 1 \leq \sum_{j \in N(i): i < j} f_{ij} - \sum_{j \in N(i): j < i} f_{ji} \quad \forall i \in V \quad (32)$$

$$k_u(1 + c_i - x_i) - 1 \geq \sum_{j \in N(i): i < j} f_{ij} - \sum_{j \in N(i): j < i} f_{ji} \quad \forall i \in V \quad (33)$$

$$-(k_u - 1)z_e \leq f_{ij} \leq (k_u - 1)z_e \quad \forall e = \{i, j\} \in E \quad (34)$$

$$c_i \in \{0, 1\} \quad \forall i \in V \quad (35)$$

By means of (28) and (29), one vertex is selected as source node among the ones belonging to the γ -quasi-clique. Constraints (30)-(33) resemble single-flow constraints for TSP adapted on undirected graphs: (30) and (31) enforces that exactly $|Q^*| - 1$ units of flow leave the source, whereas (32) and (33) ensures that a single unit of flow is absorbed by any other vertices of the γ -quasi-clique. Finally, (34) and (35) set the bounds for f_{ij} variables and c_i respectively, where f_{ij} results zero if $e = \{i, j\} \notin E_Q$. On the whole, this method requires the addition of $O(|V| + |E|)$ variables and constraints to $[C_\gamma]$.

In [25] the sufficient conditions that ensures the connectivity of optimal solutions are depicted for the main first-order clique relaxations. In particular, a solution of the γ -QCP with $|Q|$ vertices is connected if

$$\left[\gamma \binom{|Q|}{2} - \binom{|Q| - 1}{2} \right] \geq 1 \quad (36)$$

holds. Figure (3) reports the upper bound on $|Q|$ set by inequality (36) with respect to γ .

Firstly, condition is ineffective for $\gamma < 0.64$. Then, the values of $|Q|$ required are small even for reasonable values of γ . For instance, if $\gamma = 0.9$, the maximum

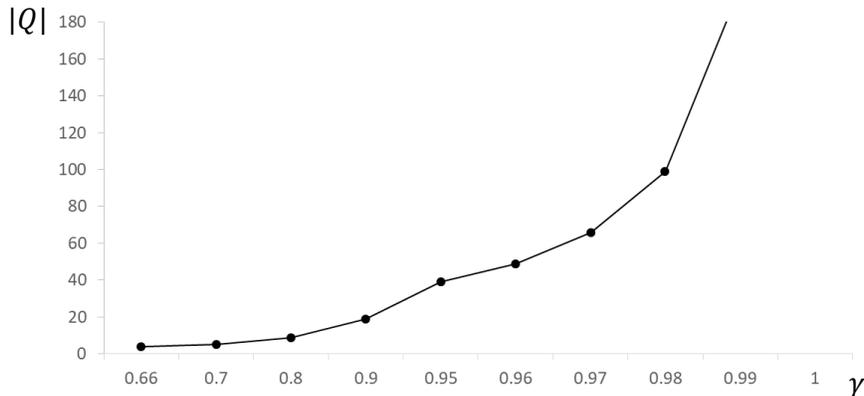


Figure 3: Upper bound on $|Q|$ values set by (36)

γ -quasi clique must contain at most 19 vertices to verify the condition and this value is not meaningful for graphs of practical interest.

In the following we present results related to the structure of γ -quasi-cliques based on the existence of small connected components. Recall that $G[Q]$ refers to the subgraph (Q, E_Q) of G induced by the subset of vertices $Q \subseteq V$.

Lemma 1. *Let G be connected and $G[Q]$ be a γ -quasi-clique formed by μ connected components with at least one component having at most three vertices. Then, there exists a γ -quasi-clique of the same order and no smaller size with at most $\mu - 1$ connected components.*

PROOF. Let X be a connected component of $G[Q]$ with at most 3 vertices, and $P = p_1 - \dots - p_q$ be the longest simple path connecting X to some other connected component of $G[Q]$, say Y , with p_1 vertex of X and p_q vertex of Y (P always exists since G is connected). If $q \leq 4$, (part of) X can be replaced by (part of) P thus reducing the number of connected components of $G[Q]$ by one, while the number of edges (vertices) of $G[Q]$ does not decrease (increase). If $q > 4$, there are at least 3 vertices $\{u, v, w\}$ in $G \setminus G[Q]$ that can be connected to some other connected component of $G[Q]$ (since G is connected). Hence

the vertices of X can be replaced by u, v, w without losing any edge and thus obtaining a γ -quasi-clique with at most $\mu - 1$ connected components. \square

The above lemma can be used to get the following result related to the maximum number of connected components of a γ -quasi-clique:

Lemma 2. *Let $G[Q]$ be a γ -quasi-clique with μ connected components and such that*

$$|E_Q| > \binom{|Q| - 4(\mu - 1)}{2} + 6(\mu - 1) \quad (37)$$

holds. If G is connected, then a γ -quasi-clique with $|Q|$ vertices and at most $\mu - 1$ connected components can be easily obtained by $G[Q]$.

PROOF. Inequality (37) forces $G[Q]$ to have a number of edges that strictly exceeds the total number of edges incident on $|Q|$ vertices disjointed into a clique Y of order $|Q| - 4(\mu - 1)$ and $\mu - 1$ cliques of 4 vertices each one (connected components with less than 4 vertices can be excluded by Lemma 1). For the properties of the binomial coefficient, removing vertices of Y (up to a minimum order of 4) to enlarge the other $\mu - 1$ cliques cannot increase the total number of edges of the μ cliques. Hence, the right-hand side of (37) defines an upper bound on the number of edges for any induced subgraph of G with $|Q|$ vertices divided into μ (or more) connected components. Therefore, if $|E_Q|$ is greater than such bound there exists at least a connected component in $G[Q]$ with at most three vertices. The thesis follows by Lemma 1. \square

A sufficient condition for the connectivity of γ -QCP solutions can be obtained by setting $\mu = 2$ in Lemma 2.

Proposition 1. *If G is connected and $|E_Q| \geq 17 + \frac{|Q|}{2}(|Q| - 9)$ then either $G[Q]$ is a connected γ -quasi-clique, or a connected γ -quasi-clique with $|Q|$ vertices can be easily obtained from $G[Q]$.*

Furthermore, the following dominance result holds:

Proposition 2. *The sufficient condition reported in Proposition 1 dominates condition (36) for any $|Q| \geq 5$.*

PROOF. Let $G[Q]$ be a γ -quasi-clique such that (36) is valid. By definition $|E_Q| \geq \gamma \binom{|Q|}{2}$ holds. Then, by inequality (36) it follows

$$\left[|E_Q| - \binom{|Q|-1}{2} \right] = |E_Q| - \binom{|Q|-1}{2} \geq 1$$

where the second member is given by the difference of two integer terms. The statement then derives by algebraic manipulations. \square

To show that the two conditions are not equivalent, we provide a simple example. Let $|Q| = 10$, $|E_Q| = 36$ and $\gamma = 0.8$. Clearly, $G[Q]$ is a γ -quasi-clique. It is easy to see that condition (36) is not satisfied, whereas Proposition 1 proves that $G[Q]$ is connected.

Note that Proposition 1 cannot explicitly detect the connectivity of γ -QCP solutions made of at most 4 vertices. Nevertheless, Lemma 1 implicitly ensures the connectivity of all γ -quasi-cliques with at most 7 vertices on connected graphs. Indeed, any partition of these vertices into two (or more) disjointed subsets would have (at least) a connected component made by at most 3 vertices. Hence, Proposition 1 integrated with Lemma 1 dominates (36) for any value of $|Q|$.

Finally, even if we contextualized the discussion to the γ -QCP, any result independent by γ can be straightforwardly inherited for analyzing the connectivity of solutions related to other optimization problems defined on graph, such as problems within the family of MRCP for which solutions can be disconnected (e.g., k -core, k -bundle).

5. Computational results

We carried out an extensive experimental campaign that mainly aim to study the quality of combinatorial and LP based dual bounds as well as the effectiveness of the sufficient conditions to recognize connected γ -quasi-cliques.

The column generation and branch-and-price algorithms have been coded in C++ and all the linear programs solved by IBM[®] CPLEX[®] 12.5.0.0 on a Intel[®] Core i7-7500U 2.70 GHz machine with 16Gb RAM. A CPU time limit of 7200 seconds has been used for all the experiments, and all the integer programs have been set with $K = \{k_L = 2, \dots, \bar{k}_U\}$ (see (24)). Solutions obtained by the greedy vertex elimination procedure of [30] have been used to measure the optimality gap (when optimal solutions were not available), and to evaluate the effectiveness of connectivity conditions.

Experiments have multiple purposes: in §5.1 the quality of the formulations $[C_\gamma]$, $[\bar{C}_\gamma]$, $[D_\gamma^S]$ and $[P_\gamma]$ has been evaluated with respect to the graph density d and parameter γ . In §5.2 the combinatorial dual bounds k_U and \bar{k}_U have been compared to each other, whereas CPU times and tightness of the dual bounds provided by $[D_\gamma^S]$ have been reported in §5.3. Finally, the branch-and-price results have been analyzed in §5.4 and the effectiveness of the sufficient connectivity conditions discussed in §5.5.

Detailed numerical results are listed in the Appendix. In the following we illustrate the experiments by means of *performance profiles* [13]: given a performance indicator β of two algorithms and/or programs a and b , e.g. the optimality gap or the CPU running time, the performance profile of a plots the fraction of the number of instances (the ordinate) for which the ratio $\beta_a / \min\{\beta_a, \beta_b\}$ is less than or equal to a given threshold (the abscissa). For the sake of readability, the abscissa axis is in logarithmic scale in all the following charts.

5.1. LP-based dual bounds: sensitivity analysis

We perform a sensitivity analysis of models in sections 2 and 3 by using six values of $\gamma = \{0.5, 0.6, 0.7, 0.8, 0.9, 0.95\}$ and 80 Erdős-Rényi uniform random

graphs [14] with $|V| = 50$. The graphs are grouped into 4 classes R_p of 20 instances each, where $p \in \{0.2, 0.4, 0.6, 0.8\}$ indicates the mean density d , e.g., $R_{0.2}$ is the set of 20 random graphs whose mean density is $d = 0.2$. Let $|Q^*|$ be the optimal (or the best) integer solution provided by CPLEX and Q_β^U the dual bound computed by means of the formulation β , with $\beta \in \{[C_\gamma], [D_\gamma^S], [\bar{C}_\gamma], [P_\gamma]\}$. The performance is evaluated in terms of the percentage optimality gap

$$OG_\beta = 100 \cdot \frac{Q_\beta^U - |Q^*|}{Q_\beta^U}. \quad (38)$$

Generally speaking, the optimality gaps often reach high values and the dual bounds are quite weak (see Table 2 in the Appendix for details). However, the best one always dominates \bar{k}_U . The gaps decrease as the density of the graphs gets larger or the γ diminishes. In any case, the bound provided by $[P_\gamma]$ is always dominated by either $Q_{[C_\gamma]}^U$ or $Q_{[\bar{C}_\gamma]}^U$.

For each class of graphs and value of γ , Figure 4 shows the best percentage gap between $OG_{[C_\gamma]}$ and $OG_{[\bar{C}_\gamma]}$: white bars indicate that $OG_{[C_\gamma]}$ is better than $OG_{[\bar{C}_\gamma]}$, black bars otherwise.

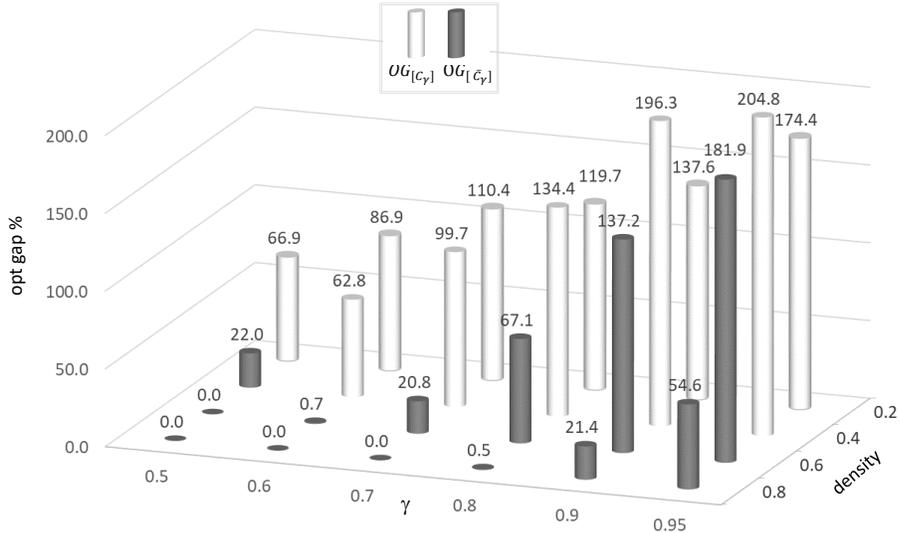


Figure 4: $\min\{OG_{[C_\gamma]}, OG_{[\bar{C}_\gamma]}\}$ for different γ and graph densities

$[C_\gamma]$ achieves the best results for $d \leq 0.4$ but one case ($\gamma = 0.5$ on class $R_{0.4}$). For larger densities, $[\bar{C}_\gamma]$ performs better in all the non-trivial cases, i.e., when the optimal solution corresponds to the trivial upper bound $|V|$. Formulations $[C_\gamma]$ and $[\bar{C}_\gamma]$ therefore looks to be complementary with respect to the graph density: $[C_\gamma]$ is suitable for computing dual bounds on sparse graphs, whereas $[\bar{C}_\gamma]$ is convenient on dense graphs. As a consequence of this combined behaviour random graphs with $d \leq 0.5$ appear the hardest to solve.

Looking at the formulation $[D_\gamma^S]$, the dual bound $Q_{[D_\gamma^S]}^U$ still continues to be slightly tighter than $Q_{[C_\gamma]}^U$ (see Table 2) though $[D_\gamma^S]$ is a surrogate relaxation of $[D_\gamma]$. On the other hand, the computational burden to solve the linear relaxation of $[D_\gamma]$ becomes much higher as the graph density increases, making $[D_\gamma]$ poorly competitive. As a final remark, the computation of all the dual bounds always required a negligible CPU time, given the small order of the considered random graphs.

5.2. Combinatorial dual bound comparison

The dual bounds k_U and \bar{k}_U has been compared to each other on two groups of instances: the 16 benchmark sparse graphs used in [31], and the 64 benchmark DIMACS instances [21] for $\gamma = \{0.5, 0.7, 0.8, 0.9\}$ (see Table 3 and Table 4 in the Appendix). The graphs in the former set are very sparse (average density $d = 0.008$) and represent real-world networks in the fields of social networks, biology, telecommunications and transportation. Those in the latter set are denser graphs (average density $d = 0.621$) often used as benchmarks for clique problems.

The percentage optimality gap (38) has been computed by means of the best known lower bound $|Q^l|$, i.e., the maximum between an optimal solution value (if available) and the heuristic solution value.

Numerical results are reported in the Appendix, Tables 5 and 6. The performance profile on the quality of the two bounds is depicted in Figure 5. The cumulative distributions show that (i) \bar{k}_U always dominates k_U (as expected), (ii) the weakness of k_U is more pronounced on sparse graphs for which the ratio

$OG_{k_U}/OG_{\bar{k}_U}$ is always ≥ 2 and reaches peaks of $2^{5.9}$, and (iii) $OG_{\bar{k}_U}$ improves OG_{k_U} of at least 50% in roughly the 80% of DIMACS graphs.

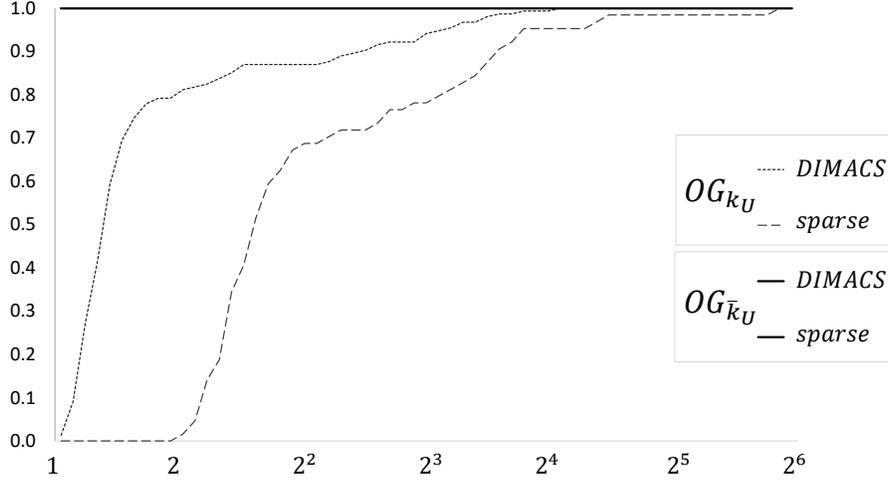


Figure 5: Performance profile of gaps OG_{k_U} and $OG_{\bar{k}_U}$

Besides trivial-instances and three additional cases with $\gamma = 0.5$, in which both OG_{k_U} and $OG_{\bar{k}_U}$ are $\leq 1.02\%$, the average optimality gaps always result considerably large, in particular for sparse graphs, and grow for increasing values of γ . The mean ratio $OG_{k_U}/OG_{\bar{k}_U}$ decreases as γ increases on DIMACS instances (from 3.97 for $\gamma = 0.5$ to 1.83 for $\gamma = 0.9$) and exhibits an opposite behaviour on sparse graphs (from 4.43 for $\gamma = 0.5$ to 8.96 for $\gamma = 0.9$).

As remarked in §2.2, the clique-based bound k_U^ω defined by (15) rapidly becomes weak on dense graphs. Indeed, it can be exploited only in two DIMACS instances, namely D_{35} and D_{38} with $\gamma = 0.9$, provided that we assume the best known lower bound $\omega_L(G)$ on the clique number $\omega(G)$ as the optimal clique number [21]. Under this hypothesis, the quality of k_U^ω appears largely better than \bar{k}_U , with an absolute reduction of 72 vertices in D_{35} and 104 vertices in D_{38} , respectively. Nevertheless, the optimality gap still remains large, i.e. 350% for the former instance and 710% for the latter. We point out that the bound k_U^ω is defined up to $\omega(G) = 9$ for $\gamma = 0.9$, and we used $\omega_L(G) = 8$ for D_{35} and

$\omega_L(G) = 9$ for D_{35} that are very close to the upper limit. In all other DIMACS instances, the value of $\omega_L(G)$ is already sufficiently large to make k_U^ω unusable.

5.3. LP-based dual bound comparison

On the same set of instances (Table 3 and Table 4) we compared the linear relaxation of integer programs $[D_\gamma^S]$, $[C_\gamma]$ and $[\bar{C}_\gamma]$ for $\gamma = \{0.5, 0.7, 0.8, 0.9\}$. Numerical results on gaps and CPU times are reported in Tables 7 and 8 (sparse graphs) and Tables 9 and 12 (DIMACS graphs).

Sparse graphs. Figures 6 and 7 depict the performance profiles of percentage optimality gaps and running times, respectively. The dual bound provided by $[C_\gamma]$ is not strictly dominated for all the values of γ . However, the quality of $[D_\gamma^S]$ is comparable to that of $[C_\gamma]$ since the mean ratio between gaps is 1.08, and in the 73.44% of the cases the bounds coincide. Moreover, $Q_{[D_\gamma^S]}^U$ improves on average the combinatorial bound \bar{k}_U by 134.15% ($\gamma = 0.5$), 140.34% ($\gamma = 0.7$), 139.43% ($\gamma = 0.8$) and 138.63% ($\gamma = 0.9$).

We do not report the performance of $[P_\gamma]$ and $[\bar{C}_\gamma]$ because the former provides a dual bound always dominated by the other formulations, whereas the latter is either not able to provide a bound within the time limit (given its $O(|\bar{E}|)$ constraints) or $Q_{[\bar{C}_\gamma]}^U$ is dominated by both $Q_{[C_\gamma]}^U$ and $Q_{[D_\gamma^S]}^U$.

The computation of $Q_{[D_\gamma^S]}^U$ is faster than that of $Q_{[C_\gamma]}^U$ (up to 20%) in about the 60% of the 64 cases, whereas $Q_{[C_\gamma]}^U$ is obtained roughly 3.5 time faster in about the 6% of the cases: the whole CPU times to get the bounds for all the 64 cases are 1919.27 seconds for the former and 2062.90 seconds for the latter, with an overall gap of 7.48%. Varying the value of γ seems to not affect the computational time significantly and there is no evidence of correlation between the two measures.

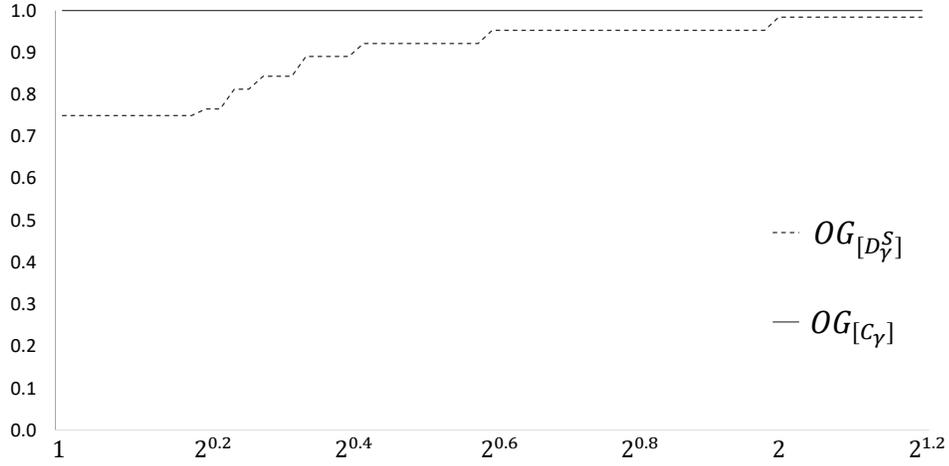


Figure 6: Performance profile of gaps $OG_{[D_\gamma^S]}$ and $OG_{[C_\gamma]}$ on sparse graphs.

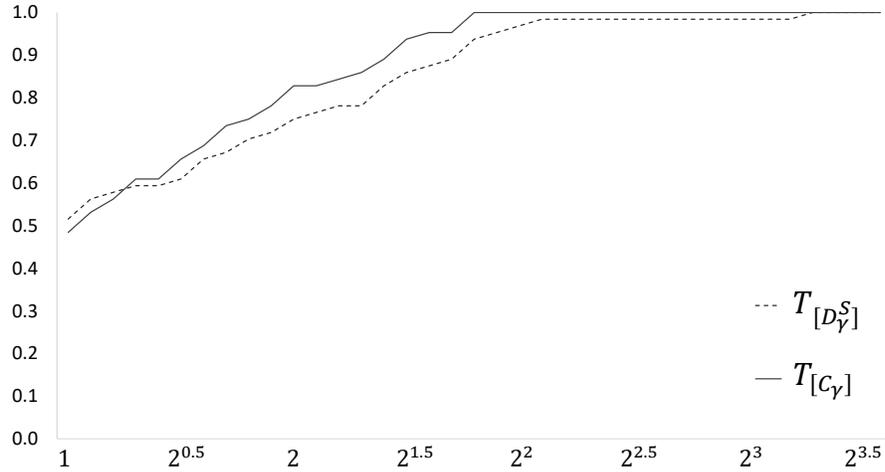


Figure 7: Performance profile of CPU running times for computing $Q_{[D_\gamma^S]}^U$ and $Q_{[C_\gamma]}^U$ on sparse graphs.

Dense graphs. Figures 8 and 9 show the performance profiles (percentage optimality gaps and running times, respectively) of the formulations $[D_\gamma^S]$ and $[\tilde{C}_\gamma]$.

$Q_{[\bar{C}_\gamma]}^U$ is better than $Q_{[D_\gamma^S]}^U$ in 90.25% of the cases providing a smaller gap up to 86%. However, $[\bar{C}_\gamma]$ is not able to produce a valid bound within the time limit in 9.10% of the cases. In particular, on D_1 - D_{12} instances $Q_{[\bar{C}_\gamma]}^U$ is, on average, 61.76% tighter than $Q_{[D_\gamma^S]}^U$ for $\gamma = 0.8$ and 79.79% for $\gamma = 0.9$, whereas it is 48.86% tighter for $\gamma = 0.9$ on the nontrivial instances of the subclass D_{51} - D_{60} . Instead, $Q_{[D_\gamma^S]}^U$ is generally better in groups D_{13} - D_{19} and D_{35} - D_{49} : in the former, $[D_\gamma^S]$ dominates $[\bar{C}_\gamma]$ for larger values of γ , whereas in the latter $[D_\gamma^S]$ is worse only for instances with high density value (d above 0.74) where however $[\bar{C}_\gamma]$ is not able to give a valid bound in 12 cases.

We do not report the performance profile of formulation $[C_\gamma]$ because when the solution of the continuous relaxation of $[C_\gamma]$ does not reach the time limit, the dual bounds $Q_{[C_\gamma]}^U$ and $Q_{[D_\gamma^S]}^U$ are always very close to each other. Indeed, $[C_\gamma]$ is not able to provide a valid bound in 31 of the 256 cases and, for the remaining instances, $[D_\gamma^S]$ is only slightly better with a 0.12% of mean gap. As a final remark on the quality of bounds, $Q_{[D_\gamma^S]}^U$ improves \bar{k}_U by 7.85%, 7.74%, 5.79% and 6.37% for $\gamma = 0.5, 0.7, 0.8$ and 0.9 , respectively.

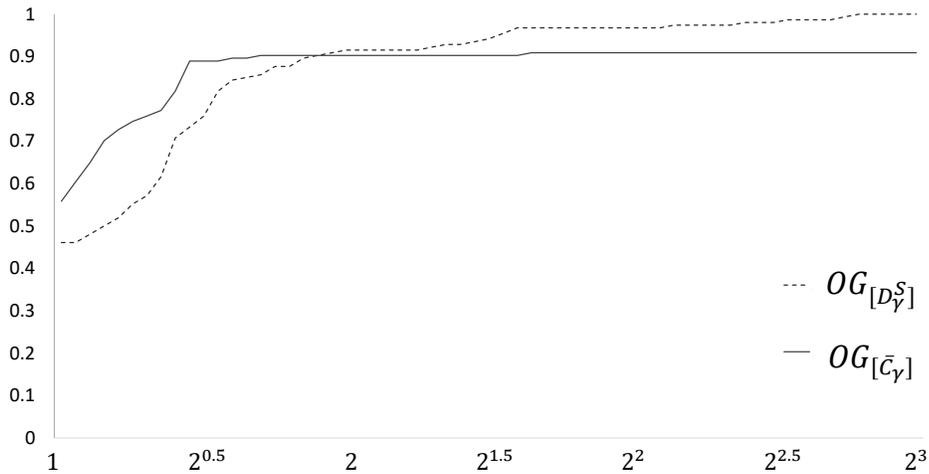


Figure 8: Performance profile of gaps $OG_{[D_\gamma^S]}$ and $OG_{[\bar{C}_\gamma]}$ on DIMACS graphs.

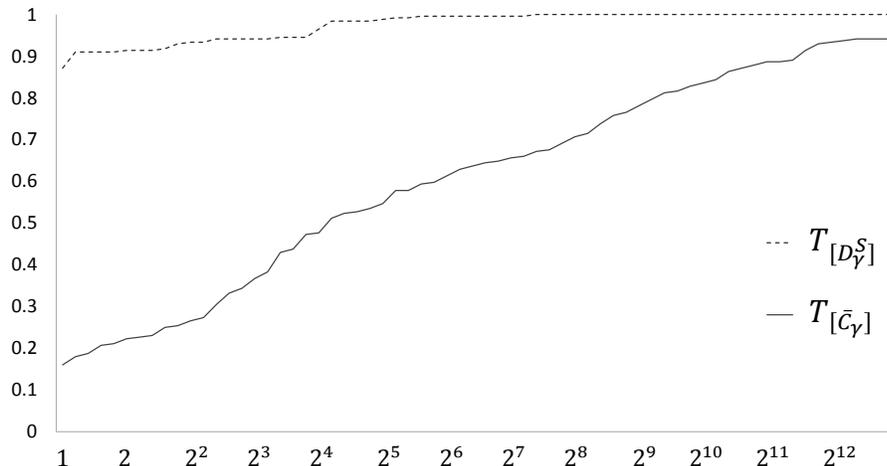


Figure 9: Performance profile of CPU running times for computing $Q_{[D_\gamma^S]}^U$ and $Q_{[\bar{C}_\gamma]}^U$ on DIMACS graphs.

The CPU time required for computing $Q_{[D_\gamma^S]}^U$ is always much smaller than that needed for obtaining $Q_{[\bar{C}_\gamma]}^U$: in 50% of the cases up to 16 times, in 8% of the cases up to 3 orders of magnitude. In particular, the column generation procedure ran for 208.42 seconds, 0.81 seconds on average, to compute the $Q_{[D_\gamma^S]}^U$ bound of all the instances; CPLEX reached the CPU time limit 31 and 15 times to compute respectively $Q_{[C_\gamma]}^U$ and $Q_{[\bar{C}_\gamma]}^U$, and required 424.66 and 290.69 seconds on average to get the bound for the remaining instances. On the base of the cases in which time limit was reached, it appears that DIMACS instances become harder as γ increases. Looking more in detail, $D_{26}-D_{29}$ is the only subgroups in which the time spent by $[\bar{C}_\gamma]$ has been roughly the same as $[D_\gamma^S]$ and the bound is better. For several cases of D_1-D_{12} and $D_{51}-D_{60}$, as example, the better dual bound provided by $[\bar{C}_\gamma]$ has been obtained by spending much more time.

Generally speaking, the above results show that $[\bar{C}_\gamma]$ is advantageous for computing bounds on quite dense graphs ($d > 0.4$), with a density threshold which slightly increases as γ grows, e.g., on instances D_{14} and D_{17} , $Q_{[\bar{C}_\gamma]}^U$ is the

best bound for $\gamma = 0.5$ and then becomes dominated for greater values of γ . In any case, the employment of $[\bar{C}_\gamma]$ requires a significant amount of CPU time that rapidly increases with the size of the graph, appearing much less scalable if compared to $[D_\gamma^S]$.

5.4. Branch-and-price results

The surrogate relaxation $[D_\gamma^S]$ is a valid alternative to rapidly compute dual bounds on large and dense graphs. Therefore, the branch-and-price algorithm described in Section 3.1 has been tested on DIMACS instances (with $\gamma = 0.9$). Table 1 reports the cases solved to optimality; all other results are listed in the Appendix, Table 13. The first three columns of Table 1 report the optimality gap, the running time $T_{[D_\gamma^S]}$ and the number of columns generated by the branch-and-price algorithm. The following columns list the optimality gaps and the CPU times $T_{[C_\gamma]}$ and $T_{[\bar{C}_\gamma]}$ used by CPLEX for solving the MIPs $[C_\gamma]$ and $[\bar{C}_\gamma]$, respectively. The optimality gap (38) is computed by using the current primal solution value and the largest dual bound among the active nodes of the search tree. The term “limit” indicates that the algorithm did not close the optimality gap after two hours of CPU time.

22 out of 64 instances have been solved to optimality within the time limit. In particular:

- the continuous relaxations of $[D_\gamma^S]$, $[C_\gamma]$ and $[\bar{C}_\gamma]$ provide the optimal value in ten cases, thus ending the search at the root node of the enumeration tree. In eight of such cases, the column generation runs approximately one order of magnitude faster.
- The instances D_{13} and D_{28} have been solved by all the integer programs $[D_\gamma^S]$, $[C_\gamma]$ and $[\bar{C}_\gamma]$, yet $T_{[D_\gamma^S]}$ is larger than the best between $T_{[C_\gamma]}$ and $T_{[\bar{C}_\gamma]}$. However, the branch-and-price is faster than CPLEX on D_{16} where CPLEX took 6323.84 seconds on $[C_\gamma]$ and runs out on $[\bar{C}_\gamma]$.
- In the two non-trivial cases D_{30} and D_{35} the branch-and-price was the

only one able to find an optimal solution within the time limit, whereas CPLEX ended with huge optimality gaps ranging from 280% to 2500%.

- Finally, CPLEX found an optimal solution in the remaining seven cases (six due to $[\bar{C}_\gamma]$ and one to $[C_\gamma]$), whereas the branch-and-price reached the time limit with a mean optimality gap of 54%.

The solution of the integer programs $[D_\gamma^S]$, $[C_\gamma]$ and $[\bar{C}_\gamma]$ reached the time limit in all the remaining 42 instances, see Table 13. Except two cases (D_1 and D_{26}), $G_{[D_\gamma^S]}$ is always much better than $\min\{OG_{[C_\gamma]}, OG_{[\bar{C}_\gamma]}\}$, though it still is quite large (544% on average), and CPLEX even was not able to compute the continuous relaxation of $[C_\gamma]$ and $[\bar{C}_\gamma]$ in 14 and 4 cases, respectively.

As a final remark, the number of columns computed by the column generation is always quite large, suggesting that the solution policy of the $|V|$ pricing problems per iteration is one of the most critical issue of the branch-and-price algorithm which deserve further investigation.

5.5. Effectiveness of connectivity conditions

The γ -quasi-cliques found by the primal heuristic on the set of uniform random graphs are all connected. Inequality (36) certifies the connectivity of the solutions in 184 of 480 cases. Proposition 1 verifies additional 107 solutions, for a total of 291 positive occurrences. The numerical results show that both the sufficient conditions are weaker for small values of γ and large density of graphs. This seems reasonable because the order $|Q|$ of γ -quasi-cliques usually get larger on instances with such features, thus making sufficient conditions poorly effective. Indeed, Proposition 1 recognizes that the computed solution is connected only on the 28.13% of DIMACS instances, though the solutions obtained for this set of graphs are all connected for any value of γ . In particular, inequality (36) is fulfilled by 17 solutions, whereas Proposition 1 is verified in other 55 additional cases. On the other hand, 15 of the 64 solutions computed for

Table 1: DIMACS instances solved to optimality

ID	Branch-and-Price			$[C_\gamma]$		$[\bar{C}_\gamma]$	
	$OG_{[D_\gamma^S]}$ (%)	$T_{[D_\gamma^S]}$ (sec.)	#Cols	$OG_{[C_\gamma]}$ (%)	$T_{[C_\gamma]}$ (sec.)	$OG_{[\bar{C}_\gamma]}$ (%)	$T_{[\bar{C}_\gamma]}$ (sec.)
D_{13}	0.00	526.42	53615	0.00	70.13	0.00	638.28
D_{14}	13.88	limit	1004282	0.00	1523.36	0.00	1728.64
D_{15}	36.50	limit	1137644	17.46	limit	0.00	2443.88
D_{16}	0.00	1917.41	238162	0.00	6323.84	53.33	limit
D_{20}	0.00	0.30	0	0.00	774.34	0.00	0.63
D_{22}	0.00	< 0.005	0	0.00	0.13	0.00	0.14
D_{23}	61.46	limit	11751925	0.00	583.88	0.00	28.22
D_{24}	0.00	0.02	0	0.00	8.56	0.00	0.20
D_{28}	0.00	184.66	214746	0.00	2.89	0.00	4.23
D_{29}	218.97	limit	5904127	231.25	limit	0.00	374.58
D_{30}	0.00	912.08	211144	1589.14	limit	280.00	limit
D_{32}	0.00	0.03	0	0.00	8.80	0.00	0.14
D_{33}	0.00	0.34	0	0.00	591.05	0.00	0.44
D_{34}	0.00	< 0.005	0	0.00	0.06	0.00	0.06
D_{35}	0.00	2619.77	866671	2500.00	limit	2060.00	limit
D_{51}	36.50	limit	2358977	852.31	limit	0.00	102.05
D_{52}	10.06	limit	2644217	172.73	limit	0.00	18.06
D_{53}	0.00	0.02	0	0.00	3.38	0.00	0.14
D_{54}	0.00	0.02	0	0.00	3.33	0.00	0.17
D_{55}	0.00	0.02	0	0.00	3.16	0.00	0.22
D_{60}	0.00	0.05	0	0.00	39.50	0.00	0.70
D_{62}	1.33	limit	1439865	1.54	limit	0.00	2.08

the sparse graph instances are disconnected (actually, four of them can easily be made connected since they are composed by only two connected components, one of which consisting of at most 3 vertices) and Proposition 1 is able to recognize the 67.92% of the 53 connected (or easily connectable) γ -quasi-cliques found. In particular, 15 connected solutions fulfill the sufficient condition (36) and other 21 additional cases are certified by Proposition 1, generally for higher values of γ ; e.g., all primal solutions are proved connected for $\gamma = 0.9$.

6. Conclusions and perspectives

In this paper a new MIP reformulation $[D_\gamma]$ for the γ -QCP, obtained by decomposing star inequalities, has been presented. The bound provided by $[D_\gamma]$ is as good as that computed by the tightest formulation $[C_\gamma]$ reported in the literature and experiments show that also the surrogate relaxation $[D_\gamma^S]$

roughly provide the same bounds. However, $[D_\gamma^S]$ seems to be more scalable since the column generation procedure becomes much faster as the density of graphs increases. On dense graphs the best dual bound is provided by formulation $[\bar{C}_\gamma]$, that models the γ -QCP by recalling the concept of complementary $(1-\gamma)$ -quasi-clique. Nevertheless, $[\bar{C}_\gamma]$ is computationally demanding for sufficiently large graphs and solving $[D_\gamma^S]$ by column generation remains a valid alternative, considering that the required CPU time is extremely limited.

Furthermore, the importance of taking into account the connectivity of MCRP solutions has been discussed. We presented a new sufficient condition to verify the connectivity of γ -quasi-cliques and, more in general, of solutions of graph optimization problems that lack of an explicit constraint of connectivity. For the γ -QCP, our result dominates the previous one reported in the literature and tests showed that it is also quite effective in practice. Indeed, it was able to prove the connectivity of primal solutions in the 49.87% of the total cases, whereas the benchmark condition was limited to the 27.00% of the instances.

Computational experiments highlighted that the γ -QCP is a challenging problem in practice and for several instances the optimality gap is very large. Although the star-based reformulation helps to compute good dual bounds in shorter time and can be successfully embedded in an exact procedure, yet our basic implementation of a branch-and-price algorithm does not definitively outperform the CPLEX MIP-solver on $[\bar{C}_\gamma]$ or $[C_\gamma]$. As future work, firstly we aim to enhance time performance of $[D_\gamma]$ by exploring a dynamic strategy based on lazy constraints to lighten formulation, where edge-flow constraints (19) are checked on the fly to ensure feasibility. Then, we are interested to the implementation of polyhedral cuts (e.g., generalized neighborhood, matching, forest) in order to tighten formulations $[D_\gamma]$ and $[D_\gamma^S]$, and to the enhancement of the pricing strategy in order to reduce the number of generated columns. On this ground, we look at the design and implementation of a full *branch-and-cut-and-price* procedure able to solve challenging instances of γ -QCP to optimality.

Acknowledgments

The authors are very grateful to the editors and the anonymous reviewers for their insightful and constructive comments and for helping the authors to significantly improve the presentation of this work.

References

- [1] Abello, J., Resende, M.G.C. and Sudarsky, S.: *Massive Quasi-Clique Detection*. LATIN 2002: Theoretical Informatics, Springer, 598–612, 2002.
- [2] Akkoyunlu, E.A.: *The Enumeration of Maximal Cliques for Large Graphs*. SIAM Journal of Computing, 2(1): 1–6, 1973.
- [3] Almeida, M.T. and Carvalho, F.D.: *Integer models and upper bounds for the 3-club problem*. Networks, 60(3): 155–166, 2012.
- [4] Bhattacharyya, M. and Bandyopadhyay, S.: *Mining the Largest Quasi-clique in Human Protein Interactome*. 2009 International Conference on Adaptive and Intelligent Systems, 194–199, 2009.
- [5] Bollobás, B. and Thomason, A.G., *Hereditary and monotone properties of graphs*, in: R.L. Graham and J. Nešetřil, eds., The mathematics of Paul Erdős, II, Algorithms and Combinatorics 14 (Springer-Verlag, 1997) 70–78.
- [6] Bourgeois, N., Giannakos, A., Lucarelli, G., Milis, I., Paschos, V. Th. and Pottié, O.: *The max quasi-independent set problem*. CSR 2010: Computer Science – Theory and Applications, pp 60-71, 2010.
- [7] Brandstädt, A., Le, V. B., Spinrad, J. P., *Graph Classes: A Survey*. Philadelphia, PA: SIAM, p. 18, 1987.
- [8] Brunato, M., Hoss, H.H. and Battiti, R.: *On Effectively Finding Maximal Quasi-cliques in Graphs*. Learning and Intelligent Optimization, Springer, 41–55, 2008.

- [9] Chataigner, F., Manić, G., Wakabayashi, Y., and Yuster, R.: *Approximation algorithms and hardness results for the clique packing problem*. Discrete Applied Mathematics, 157(7): 1396–1406, 2009.
- [10] Cornaz, D., Furini, F., Lacroix, M., Malaguti, E., Mahjoub, A.R., and Martin S.: *The vertex k -cut problem*. Discrete Optimization, 31: 8–28, 2019.
- [11] Dantzig, G.B. and Wolfe, P.: *Decomposition Principle for Linear Programs*. Operations Research, 8(1): 101–111, 1960.
- [12] Lübbecke, M.E., and Desrosiers, J.: *Selected Topics in Column Generation*. Operations Research, 53(6): 1007–1023, 2005.
- [13] Dolan, E. D., and Moré, J. J.: *Benchmarking optimization software with performance profiles*. Mathematical programming, 91(2): 201–213, 2002
- [14] Erdős, P. and Rényi, A.: *On Random Graphs. I*. Publicationes Mathematicae, 6: 290–297, 1959.
- [15] Fortunato, S.: *Community detection in graphs*. Physics Report, 486(3–5): 75–174, 2010.
- [16] Liu, G. and Wong, L.: *Effective Pruning Techniques for Mining Quasi-Clique*. ECML PKDD '08 Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases - Part II: 33–49, 2008.
- [17] Gschwind, T., Irnich, S., Furini, F., and Wolfer Calvo, R.: *Social network analysis and community detection by decomposing a graph into relaxed cliques*. Technical Report LM-2017-06, Chair of Logistics Management, Gutenberg School of Management and Economics, Johannes Gutenberg University Mainz, Mainz, Germany.
- [18] Krislock, N., Malick, J. and Roupin, F.: *Computational results of a semidefinite branch-and-bound algorithm for k -cluster*. Computers & Operations Research, 66: 153–159, 2016.

- [19] Marinelli, F., Pizzuti, A. and Rossi, F.: *A star-based reformulation for the maximum quasi-clique problem*. Proceedings of the 16th CTW on Graphs and Combinatorial Optimization: 118–121, 2018.
- [20] Moradi, E. and Balasundaram, B.: *Finding a maximum k -club using the k -clique formulation and canonical hypercube cuts*. Optimization Letters, 12(8): 1947–1957, 2018.
- [21] Rossi, R.A. and Ahmed, N.K.: *The network Data Repository with Interactive Graph Analytics and Visualization*. Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, <http://networkrepository.com>.
- [22] Pajouh, F.M., Miao, Z. and Balasundaram, B.: *A branch-and-bound approach for maximum quasi-cliques*. Annals of Operations Research. 216: 145–161, 2014.
- [23] Pardalos, P.M. and Xue, J.: *The maximum clique problem*. Journal of Global Optimization. Optim. 4(3): 301–328, 1994.
- [24] Pastukhov, G., Veremyev, A., Boginski, V. and Prokopyev, Oleg A.: *On maximum degree-based γ -quasi-clique problem: Complexity and exact approaches*. Networks 71(2): 136–152, 2018.
- [25] Pattilo, J., Youssef, N. and Butenko, S.: *On clique relaxation models in network analysis*. European Journal of Operational Research. 226(1): 9–18, 2013.
- [26] Pattilo, J., Veremyev, A., Butenko, S. and Boginski, V.: *On the maximum quasi-clique problem*. Discrete Applied Mathematics. 161: 244–257, 2013.
- [27] Carvajal, R., Constantino, M., Goycoolea, M., Vielma, J. P. and Andrés Weintraub: *Imposing connectivity constraints in forest planning models*. Operations Research, 61(4): 824–836, 2013.

- [28] San Segundo, P., Coniglio S., Furini, F. and Ljubić, I.: *A new branch-and-bound algorithm for the maximum edge-weighted clique problem*. European Journal of Operational Research, 278(1), 76–90, 2019.
- [29] Taccari, L.: *Integer programming formulations for the elementary shortest path problem*. European Journal of Operational Research, 252(1): 122–130, 2016.
- [30] Tsourakakis, C., Bonchi, F., Gionis, A., Gullo, F. and Tsiarli M.: *Denser than the densest subgraph: extracting optimal quasi-cliques with quality guarantees*. KDD '13 Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining: 104-112, 2013.
- [31] Veremyev, A., Prokopyev, O.A., Butenko, S., and Pasiliao, E.L.: *Exact MIP-based approaches for finding maximum quasi-cliques and dense subgraphs*. Computational Optimization and Applications, 64: 177–214, 2016.
- [32] Wang, Y., Buchanan, A., and Butenko, S.: *On imposing connectivity constraints in integer programs*. Mathematical Programming, 166(1): 241–271, 2017.

Appendix

In this section we report the detailed numerical results discussed in Section 5. Table 2 lists the percentage optimality gaps obtained with the linear relaxations of the integer programs. Each entry is the average computed on the elements of one of the four considered classes of random graphs. Bold numbers indicate whenever a formulation strictly dominates all the others. Attributes of sparse and dense graphs (order, size and density) are indicated in Tables 3 and 4, respectively.

Tables 5 and 6 show the percentage optimality gaps OG_{k_U} and $OG_{\bar{k}_U}$ between the value $|Q^l|$ provided by the greedy vertex elimination heuristic and the dual bounds k_U and \bar{k}_U respectively. A “-” mark indicates an instance for which the dual bounds are trivially equal to $|V|$ and the primal solution is optimal.

Percentage optimality gaps and CPU running times for solving the continuous relaxations of integer programs $[D_\gamma^S]$, $[C_\gamma]$ and $[\bar{C}_\gamma]$ are reported in Tables 7 and 8 (sparse graphs), and Tables 9-12 (DIMACS dense graphs); bold values refer to strictly better gaps or CPU times.

Finally, Table 13 reports the optimality gaps for the DIMACS instances for which neither the branch-and-price nor CPLEX running on the MIPs $[C_\gamma]$ and $[\bar{C}_\gamma]$ have been able to solve the instance within the time limit of 7200 seconds. A dash indicates that even the solution of the continuous relaxation was not available within the time limit.

Table 2: percentage optimality gaps on random graphs

γ	$R_{0.2}$				$R_{0.4}$			
	$OG_{[C_\gamma]}$ (%)	$OG_{[D_\gamma^S]}$ (%)	$OG_{[\tilde{C}_\gamma]}$ (%)	$OG_{[P_\gamma]}$ (%)	$OG_{[C_\gamma]}$ (%)	$OG_{[D_\gamma^S]}$ (%)	$OG_{[\tilde{C}_\gamma]}$ (%)	$OG_{[P_\gamma]}$ (%)
0.5	66.91	66.05	97.50	141.38	26.63	26.60	22.01	26.90
0.6	86.92	86.22	123.92	203.69	62.82	62.79	63.60	66.61
0.7	110.36	109.35	157.11	279.57	99.66	99.62	110.46	112.51
0.8	119.65	118.93	169.10	335.80	134.41	134.17	154.58	160.82
0.9	137.61	135.98	193.50	415.98	196.31	194.72	222.65	245.88
0.95	174.43	171.86	232.25	518.15	204.80	201.08	230.78	264.36
γ	$R_{0.6}$				$R_{0.8}$			
	$OG_{[C_\gamma]}$ (%)	$OG_{[D_\gamma^S]}$ (%)	$OG_{[\tilde{C}_\gamma]}$ (%)	$OG_{[P_\gamma]}$ (%)	$OG_{[C_\gamma]}$ (%)	$OG_{[D_\gamma^S]}$ (%)	$OG_{[\tilde{C}_\gamma]}$ (%)	$OG_{[P_\gamma]}$ (%)
0.5	0.00	0.00	0.0	0.00	0.00	0.00	0.0	0.00
0.6	1.75	1.75	0.65	1.82	0.00	0.00	0.0	0.00
0.7	38.73	38.73	20.79	29.61	0.00	0.00	0.0	0.00
0.8	97.55	97.53	67.13	75.14	1.28	1.28	0.45	1.40
0.9	183.67	182.66	137.23	143.51	61.76	61.70	21.35	33.71
0.95	238.74	234.95	181.89	186.83	124.47	123.26	54.56	65.72

Table 3: Attributes of sparse graphs

ID	Name	$ V $	$ E $	d
I_1	USAir97	332	2126	0.0387
I_2	Harvard500	500	2043	0.0164
I_3	Email	1133	5451	0.0085
I_4	Homer	561	1628	0.0104
I_5	SmallW	396	994	0.0127
I_6	Erdos971	472	1314	0.0118
I_7	Netscience	1589	2742	0.0022
I_8	C.Elegans	453	2025	0.0198
I_9	Erdos02	6927	8472	0.0004
I_{10}	Geom	7343	11898	0.0004
I_{11}	ca-HepTh	9877	25973	0.0005
I_{12}	ca-GrQc	5242	14484	0.0011
I_{13}	AS-735	7716	12572	0.0004
I_{14}	PGPgiantcompo	10680	24316	0.0004
I_{15}	EVA	8497	6711	0.0002
I_{16}	California	9664	15969	0.0003

Table 4: Attributes of DIMACS graphs

ID	Name	$ V $	$ E $	d	ID	Name	$ V $	$ E $	d
D_1	brock200-1	200	14834	0.75	D_{33}	MANN-a45	1035	533115	1.00
D_2	brock200-2	200	9876	0.50	D_{34}	MANN-a9	45	918	0.93
D_3	brock200-3	200	12048	0.61	D_{35}	p-hat300-1	300	10933	0.24
D_4	brock200-4	200	13089	0.66	D_{36}	p-hat300-2	300	21928	0.49
D_5	brock400-1	400	59723	0.75	D_{37}	p-hat300-3	300	33390	0.74
D_6	brock400-2	400	59786	0.75	D_{38}	p-hat500-1	500	31569	0.25
D_7	brock400-3	400	59681	0.75	D_{39}	p-hat500-2	500	62946	0.50
D_8	brock400-4	400	59765	0.75	D_{40}	p-hat500-3	500	93800	0.75
D_9	brock800-1	800	207505	0.65	D_{41}	p-hat700-1	700	60999	0.25
D_{10}	brock800-2	800	208166	0.65	D_{42}	p-hat700-2	700	121728	0.50
D_{11}	brock800-3	800	207333	0.65	D_{43}	p-hat700-3	700	183010	0.75
D_{12}	brock800-4	800	207643	0.65	D_{44}	p-hat1000-1	1000	122253	0.24
D_{13}	c-fat200-1	200	1534	0.08	D_{45}	p-hat1000-2	1000	244799	0.49
D_{14}	c-fat200-2	200	3235	0.16	D_{46}	p-hat1000-3	1000	371746	0.74
D_{15}	c-fat200-5	200	8473	0.43	D_{47}	p-hat1500-1	1500	284923	0.25
D_{16}	c-fat500-1	500	4459	0.04	D_{48}	p-hat1500-2	1500	568960	0.51
D_{17}	c-fat500-10	500	46627	0.37	D_{49}	p-hat1500-3	1500	847244	0.75
D_{18}	c-fat500-2	500	9139	0.07	D_{50}	san1000	1000	250500	0.50
D_{19}	c-fat500-5	500	23191	0.19	D_{51}	san200-0.7-1	200	13930	0.70
D_{20}	hamming10-2	1024	518656	0.99	D_{52}	san200-0.7-2	200	13930	0.70
D_{21}	hamming10-4	1024	434176	0.83	D_{53}	san200-0.9-1	200	17910	0.90
D_{22}	hamming6-2	64	1824	0.90	D_{54}	san200-0.9-2	200	17910	0.90
D_{23}	hamming6-4	64	704	0.35	D_{55}	san200-0.9-3	200	17910	0.90
D_{24}	hamming8-2	256	31616	0.97	D_{56}	san400-0.5-1	400	39900	0.50
D_{25}	hamming8-4	256	20864	0.64	D_{57}	san400-0.7-1	400	55860	0.70
D_{26}	johnson16-2-4	120	5460	0.76	D_{58}	san400-0.7-2	400	55860	0.70
D_{27}	johnson32-2-4	496	107880	0.88	D_{59}	san400-0.7-3	400	55860	0.70
D_{28}	johnson8-2-4	28	210	0.56	D_{60}	san400-0.9-1	400	71820	0.90
D_{29}	johnson8-4-4	70	1855	0.77	D_{61}	sanr200-0.7	200	13868	0.70
D_{30}	keller4	171	9435	0.65	D_{62}	sanr200-0.9	200	17863	0.90
D_{31}	keller5	776	225990	0.75	D_{63}	sanr400-0.5	400	39984	0.50
D_{32}	MANN-a27	378	70551	0.99	D_{64}	sanr400-0.7	400	55869	0.70

Table 5: percentage optimality gaps obtained with the dual bounds k_U and \bar{k}_U

ID	$\gamma = 0.5$		$\gamma = 0.7$		$\gamma = 0.8$		$\gamma = 0.9$	
	$OG_{k_U}^*$ (%)	$OG_{\bar{k}_U}^*$ (%)	$OG_{k_U}^*$ (%)	$OG_{\bar{k}_U}^*$ (%)	$OG_{k_U}^*$ (%)	$OG_{\bar{k}_U}^*$ (%)	$OG_{k_U}^*$ (%)	$OG_{\bar{k}_U}^*$ (%)
D_1	-	-	-	-	73.87	68.47	333.33	297.62
D_2	1.53	1.02	354.05	294.59	582.61	460.87	1133.33	858.33
D_3	-	-	129.63	116.05	370.27	316.22	680.95	557.14
D_4	-	-	47.33	44.27	262.00	232.00	677.27	577.27
D_5	-	-	-	-	109.78	103.80	613.73	556.86
D_6	-	-	-	-	116.20	109.50	574.07	520.37
D_7	-	-	-	-	112.09	106.04	586.79	532.08
D_8	-	-	-	-	109.19	102.70	574.07	520.37
D_9	-	-	150.81	142.35	800.00	717.50	2090.32	1780.65
D_{10}	-	-	139.44	131.99	768.67	691.57	2093.55	1790.32
D_{11}	-	-	154.97	146.36	767.47	687.95	2090.32	1780.65
D_{12}	-	-	154.13	145.87	847.37	761.84	2241.38	1913.79
D_{13}	160.00	16.67	247.37	31.58	287.50	37.50	346.15	46.15
D_{14}	96.55	15.52	152.63	26.32	181.25	31.25	226.92	46.15
D_{15}	24.32	15.54	62.50	28.13	84.81	35.44	110.77	47.69
D_{16}	282.86	17.14	413.64	31.82	488.89	44.44	566.67	53.33
D_{17}	33.33	15.74	74.64	28.23	97.11	35.84	128.37	48.23
D_{18}	189.39	16.67	285.71	30.95	331.43	37.14	393.10	48.28
D_{19}	85.98	15.24	142.45	27.36	177.01	36.78	219.72	49.30
D_{20}	-	-	-	-	-	-	-	-
D_{21}	-	-	-	-	-	-	84.59	77.26
D_{22}	-	-	-	-	-	-	-	-
D_{23}	65.63	40.63	462.50	300.00	740.00	460.00	900.00	525.00
D_{24}	-	-	-	-	-	-	-	-
D_{25}	-	-	57.42	50.32	221.13	187.32	923.81	766.67
D_{26}	-	-	-	-	244.12	235.29	1000.00	920.00
D_{27}	-	-	-	-	-	-	590.14	581.69
D_{28}	-	-	212.50	175.00	360.00	280.00	450.00	325.00
D_{29}	-	-	-	-	100.00	97.06	540.00	490.00
D_{30}	-	-	41.38	37.07	227.66	200.00	752.94	641.18
D_{31}	-	-	-	-	59.32	54.66	516.52	466.96
D_{32}	-	-	-	-	-	-	-	-
D_{33}	-	-	-	-	-	-	-	-
D_{34}	-	-	-	-	-	-	-	-
D_{35}	231.75	182.54	785.00	580.00	1169.23	830.77	1850.00	1250.00
D_{36}	1.02	0.68	42.86	33.14	105.26	84.21	268.33	211.67
D_{37}	-	-	-	-	21.94	18.99	122.95	108.20
D_{38}	277.66	223.40	1150.00	862.50	1552.94	1111.76	2550.00	1750.00
D_{39}	-	-	37.22	28.48	88.15	70.14	240.00	190.91
D_{40}	-	-	-	-	17.76	15.33	107.73	94.09
D_{41}	329.57	266.96	1337.93	1003.45	2343.75	1681.25	3988.89	2744.44
D_{42}	0.29	0.14	37.21	28.37	91.67	73.26	220.99	173.46
D_{43}	-	-	-	-	20.07	17.41	115.54	101.01
D_{44}	402.88	326.62	1746.88	1306.25	2810.53	2010.53	4636.36	3172.73
D_{45}	1.02	0.61	42.91	33.16	103.12	82.60	284.38	226.04
D_{46}	-	-	-	-	23.12	20.05	131.30	115.52
D_{47}	396.74	324.19	2212.82	1674.36	3736.36	2704.55	5585.71	3864.29
D_{48}	-	-	36.22	27.67	85.83	68.22	231.56	183.48
D_{49}	-	-	-	-	18.29	15.85	111.08	97.54

Table 6: percentage optimality gaps obtained with the dual bounds k_U and \bar{k}_U

ID	$\gamma = 0.5$		$\gamma = 0.7$		$\gamma = 0.8$		$\gamma = 0.9$	
	OG_{k_U} (%)	$OG_{\bar{k}_U}$ (%)	OG_{k_U} (%)	$OG_{\bar{k}_U}$ (%)	OG_{k_U} (%)	$OG_{\bar{k}_U}$ (%)	OG_{k_U} (%)	$OG_{\bar{k}_U}$ (%)
D_{50}	-	-	36.89	18.77	40.75	15.48	2094.12	1611.76
D_{51}	-	-	-	-	41.67	33.33	58.56	41.44
D_{52}	-	-	-	-	16.88	11.88	27.54	18.12
D_{53}	-	-	-	-	-	-	-	-
D_{54}	-	-	-	-	-	-	-	-
D_{55}	-	-	-	-	-	-	-	-
D_{56}	-	-	36.29	18.55	41.07	16.07	1319.05	1009.52
D_{57}	-	-	-	-	49.60	40.80	66.82	48.82
D_{58}	-	-	-	-	46.67	38.04	63.72	46.05
D_{59}	-	-	-	-	41.13	33.21	58.56	41.89
D_{60}	-	-	-	-	-	-	-	-
D_{61}	-	-	1.53	1.53	165.71	151.43	433.33	375.76
D_{62}	-	-	-	-	-	-	1.53	1.53
D_{63}	-	-	576.00	486.00	1164.00	932.00	2192.31	1669.23
D_{64}	-	-	-	-	252.83	232.08	802.56	707.69
I_1	37.31	14.93	59.18	22.45	69.77	23.26	97.14	31.43
I_2	143.24	40.54	192.31	42.31	195.83	33.33	191.30	21.74
I_3	492.00	184.00	792.86	278.57	800.00	261.54	746.15	223.08
I_4	145.45	69.70	277.78	138.89	326.67	153.33	328.57	128.57
I_5	125.00	57.14	211.76	94.12	257.14	107.14	327.27	127.27
I_6	217.39	95.65	369.23	169.23	470.00	210.00	575.00	250.00
I_7	238.71	22.58	270.83	20.83	277.27	13.64	271.43	4.76
I_8	200.00	83.33	322.22	127.78	373.33	140.00	458.33	158.33
I_9	922.22	361.11	1318.18	472.73	1522.22	522.22	1612.50	525.00
I_{10}	395.45	109.09	607.69	169.23	616.67	158.33	608.70	130.43
I_{11}	475.00	44.64	615.79	60.53	628.57	54.29	627.27	45.45
I_{12}	197.53	16.05	269.09	29.09	272.55	23.53	265.31	12.24
I_{13}	522.22	202.78	691.67	241.67	831.58	273.68	1013.33	313.33
I_{14}	327.40	53.42	438.78	71.43	448.89	64.44	452.38	54.76
I_{15}	1722.22	766.67	2660.00	1080.00	3150.00	1200.00	2950.00	1025.00
I_{16}	873.08	342.31	1683.33	616.67	1566.67	533.33	1466.67	450.00

Table 7: percentage optimality gaps on sparse graphs

ID	$\gamma = 0.5$			$\gamma = 0.7$			$\gamma = 0.8$			$\gamma = 0.9$		
	$ Q^* $	$OG_{[C_\gamma]}$ (%)	$OG_{[D_\gamma^*]}$ (%)									
I_1	67	0.00	0.00	49	0.00	0.00	43	0.00	0.00	35	8.57	11.43
I_2	37	10.81	10.81	26	15.38	15.38	24	8.33	8.33	23	4.35	4.35
I_3	25	28	28	14	64.29	64.29	13	53.85	53.85	13	38.46	38.46
I_4	33	3.03	6.06	18	33.33	38.89	15	40.00	46.67	14	35.71	35.71
I_5	28	3.57	7.14	17	23.53	23.53	14	28.57	35.71	11	45.45	54.55
I_6	23	13.04	13.04	13	46.15	46.15	10	60.00	70.00	8	87.5	87.5
I_7	31	0.00	0.00	24	4.17	4.17	22	4.55	4.55	21	0.00	0.00
I_8	30	3.33	3.33	18	22.22	22.22	15	26.67	33.33	12	41.67	50.00
I_9	18	11.11	16.67	11	36.36	36.36	9	44.44	55.56	8	37.50	50.00
I_{10}	44	0.00	2.27	26	23.08	23.08	24	16.67	16.67	23	8.7	8.7
I_{11}	56	0.00	0.00	38	10.53	10.53	35	8.57	8.57	33	3.03	3.03
I_{12}	81	2.47	2.47	55	12.73	12.73	51	7.84	7.84	49	2.04	2.04
I_{13}	36	0.00	0.00	24	8.33	8.33	19	21.05	21.05	15	33.33	33.33
I_{14}	73	2.74	4.11	49	12.24	12.24	45	6.67	6.67	42	2.38	2.38
I_{15}	9	0.00	0.00	5	40.00	40.00	4	50.00	50.00	4	25.00	25.00
I_{16}	26	34.62	34.62	12	108.33	108.33	12	83.33	83.33	12	58.33	66.67

Table 8: CPU times on sparse graphs (in sec.)

ID	$\gamma = 0.5$		$\gamma = 0.7$		$\gamma = 0.8$		$\gamma = 0.9$	
	$T_{[C_\gamma]}$	$T_{[D_\gamma^S]}$	$T_{[C_\gamma]}$	$T_{[D_\gamma^S]}$	$T_{[C_\gamma]}$	$T_{[D_\gamma^S]}$	$T_{[C_\gamma]}$	$T_{[D_\gamma^S]}$
I_1	0.45	0.39	0.72	0.36	0.83	1.2	0.88	0.27
I_2	1.06	0.31	0.98	0.64	1.53	0.52	1.11	0.33
I_3	11.48	8.41	12.13	6.55	11.33	8.28	12.02	6.22
I_4	0.66	0.64	0.59	0.89	0.70	0.61	0.83	0.55
I_5	0.09	0.23	0.23	0.22	0.31	0.14	0.25	0.09
I_6	0.17	1.61	0.30	1.11	0.22	0.92	0.17	0.59
I_7	0.97	1.11	0.95	1.02	1.30	0.77	1.00	1.19
I_8	0.95	0.59	1.05	0.56	0.80	0.56	0.94	0.36
I_9	5.98	2.28	6.38	17.11	6.94	15.86	6.59	16.27
I_{10}	11.17	30.69	10.11	36.36	9.69	31.7	9.55	31.36
I_{11}	225.55	210.61	291.54	180.00	278.91	253.8	348.85	263.52
I_{12}	27.28	22.39	21.67	20.84	18.88	19.59	21.95	22.70
I_{13}	8.55	12.55	7.49	11.95	7.47	10.20	7.72	13.09
I_{14}	62.99	123.19	56.25	117.36	65.65	110.98	60.97	115.92
I_{15}	1.75	3.22	1.17	3.39	1.19	3.08	1.11	3.36
I_{16}	100.61	43.09	112.52	42.09	79.42	40.89	120.03	42.58

Table 9: percentage optimality gaps on DIMACS graphs

ID	$\gamma = 0.5$				$\gamma = 0.7$				$\gamma = 0.8$				$\gamma = 0.9$			
	$ Q^* $	$OG_{[C_\gamma]}$ (%)	$OG_{[D_\gamma^S]}$ (%)	$OG_{[C_\gamma]}$ (%)	$ Q^* $	$OG_{[C_\gamma]}$ (%)	$OG_{[D_\gamma^S]}$ (%)	$OG_{[C_\gamma]}$ (%)	$ Q^* $	$OG_{[C_\gamma]}$ (%)	$OG_{[D_\gamma^S]}$ (%)	$OG_{[C_\gamma]}$ (%)	$ Q^* $	$OG_{[C_\gamma]}$ (%)	$OG_{[D_\gamma^S]}$ (%)	$OG_{[C_\gamma]}$ (%)
D_1	200	0.00	0.00	0.00	200	0.00	0.00	0.00	111	67.95	67.95	48.25	42	294.81	294.80	194.65
D_2	196	1.29	1.02	0.76	37	283.99	284.00	252.74	23	441.07	441.07	405.87	12	822.77	822.68	790.74
D_3	200	0.00	0.00	0.00	81	113.71	113.71	90.83	37	309.72	309.72	244.22	21	542.22	542.18	428.70
D_4	200	0.00	0.00	0.00	131	43.50	43.50	33.97	50	229.22	229.22	173.28	22	565.60	565.58	421.42
D_5	400	0.00	0.00	0.00	400	0.00	0.00	0.00	184	103.41	103.41	78.68	51	552.54	552.53	381.89
D_6	400	0.00	0.00	0.00	400	0.00	0.00	0.00	179	109.31	109.31	83.64	54	516.93	516.92	355.36
D_7	400	0.00	0.00	0.00	400	0.00	0.00	0.00	182	105.50	105.50	80.34	53	527.47	527.46	363.22
D_8	400	0.00	0.00	0.00	400	0.00	0.00	0.00	185	102.45	102.45	77.76	54	516.72	516.71	355.48
D_9	800	0.00	0.00	0.00	307	141.72	141.72	119.83	80	-	711.82	562.65	31	-	1762.58	1353.49
D_{10}	800	0.00	0.00	0.00	322	131.20	131.20	111.16	83	-	684.96	541.40	31	-	1768.51	1356.53
D_{11}	800	0.00	0.00	0.00	302	145.52	145.52	123.39	83	-	681.82	538.86	31	-	1761.04	1354.19
D_{12}	800	0.00	0.00	0.00	303	145.08	145.08	123.29	76	-	755.11	598.65	29	-	1892.36	1454.60
D_{13}	30	11.31	11.24	16.67	19	26.58	26.58	31.58	16	32.73	32.72	37.5	13	46.15	46.15	46.15
D_{14}	58	13.27	13.23	15.52	38	24.19	24.19	26.32	32	29.46	29.46	31.25	26	42.09	41.96	46.15
D_{15}	148	15.18	15.17	8.09	96	27.12	27.12	24.34	79	35.33	35.33	35.44	65	46.37	46.36	47.69
D_{16}	35	13.13	13.06	17.14	22	29.56	29.56	31.82	18	39.56	39.56	44.44	15	49.56	49.44	53.33
D_{17}	324	15.44	15.44	10.32	209	27.96	27.96	28.23	173	35.34	35.34	35.84	141	47.68	47.68	48.23
D_{18}	66	12.80	12.76	16.67	42	27.24	27.24	30.95	35	33.99	33.99	37.14	29	44.15	44.13	48.28
D_{19}	164	13.74	13.73	15.24	106	25.95	25.95	27.36	87	34.43	34.43	36.78	71	46.58	46.56	49.30
D_{20}	1024	0.00	0.00	0.00	1024	0.00	0.00	0.00	1024	0.00	0.00	0.00	1024	0.00	0.00	0.00
D_{21}	1024	0.00	0.00	0.00	1024	0.00	0.00	0.00	1024	0.00	0.00	0.00	532	-	77.26	31.66
D_{22}	64	0.00	0.00	0.00	64	0.00	0.00	0.00	64	0.00	0.00	0.00	64	0.00	0.00	0.00
D_{23}	32	40.63	40.63	36.63	8	300.00	300.00	300.00	5	460.00	460.00	460.00	4	535.87	525.00	525.00
D_{24}	256	0.00	0.00	0.00	256	0.00	0.00	0.00	256	0.00	0.00	0.00	256	0.00	0.00	0.00
D_{25}	256	0.00	0.00	0.00	155	50.32	50.32	32.77	71	187.32	187.32	131.24	21	767.20	766.67	575.90
D_{26}	120	0.00	0.00	0.00	120	0.00	0.00	0.00	34	235.29	235.29	195.61	10	921.10	920.00	631.37
D_{27}	496	0.00	0.00	0.00	496	0.00	0.00	0.00	496	0.00	0.00	0.00	71	-	581.69	484.44
D_{28}	28	0.00	0.00	0.00	8	175.00	175.00	136.27	5	280.00	280.00	228.57	4	341.35	325.00	273.60
D_{29}	70	0.00	0.00	0.00	70	0.00	0.00	0.00	34	97.06	97.06	74.93	10	498.87	490.00	326.80
D_{30}	171	0.00	0.00	0.00	116	36.75	36.75	26.70	47	195.60	195.60	148.38	17	627.12	627.07	481.76
D_{31}	776	0.00	0.00	0.00	776	0.00	0.00	0.00	472	-	54.46	36.62	115	-	463.62	320.96
D_{32}	378	0.00	0.00	0.00	378	0.00	0.00	0.00	378	0.00	0.00	0.00	378	0.00	0.00	0.00
D_{33}	1035	0.00	0.00	0.00	1035	0.00	0.00	0.00	1035	0.00	0.00	0.00	1035	0.00	0.00	0.00
D_{34}	45	0.00	0.00	0.00	45	0.00	0.00	0.00	45	0.00	0.00	0.00	45	0.00	0.00	0.00

Table 10: percentage optimality gaps on DIMACS graphs

ID	$\gamma = 0.5$				$\gamma = 0.7$				$\gamma = 0.8$				$\gamma = 0.9$			
	$ Q $	$OG_{[c_\gamma]}$ (%)	$OG_{[D_\gamma^S]}$ (%)	$OG_{[c_\gamma]}$ (%)	$ Q $	$OG_{[c_\gamma]}$ (%)	$OG_{[D_\gamma^S]}$ (%)	$OG_{[c_\gamma]}$ (%)	$ Q $	$OG_{[c_\gamma]}$ (%)	$OG_{[D_\gamma^S]}$ (%)	$OG_{[c_\gamma]}$ (%)	$ Q $	$OG_{[c_\gamma]}$ (%)	$OG_{[D_\gamma^S]}$ (%)	$OG_{[c_\gamma]}$ (%)
D_{35}	63	134.29	134.62	182.54	20	428.52	429.25	580.00	13	612.39	612.23	830.77	8	930.53	928.99	1250.00
D_{36}	293	0.27	0.30	0.48	175	20.08	20.18	24.44	114	61.40	61.53	67.02	60	172.78	172.24	179.95
D_{37}	300	0.00	0.00	0.00	300	0.00	0.00	0.00	237	17.83	17.83	10.09	122	103.55	103.31	63.42
D_{38}	94	170.06	170.27	223.40	24	656.70	657.55	862.50	17	835.49	835.71	1111.76	10	1314.75	1315.29	1750.00
D_{39}	500	0.00	0.00	0.00	309	16.82	16.92	20.19	211	49.75	49.87	53.04	110	155.43	154.76	156.72
D_{40}	500	0.00	0.00	0.00	500	0.00	0.00	0.00	411	14.36	14.36	7.74	220	89.95	89.63	53.59
D_{41}	115	204.43	204.66	266.96	29	763.27	763.94	1003.45	16	1269.89	1269.82	1681.25	9	2066.02	2066.68	2744.44
D_{42}	696	0.12	0.11	0.13	430	15.84	15.99	19.98	288	60.28	51.58	56.43	162	139.28	138.95	143.94
D_{43}	700	0.00	0.00	0.00	700	0.00	0.00	0.00	563	16.27	16.27	9.11	296	-	96.32	58.76
D_{44}	139	252.98	253.25	-	32	996.08	996.89	-	19	1515.94	1516.25	2010.53	11	2382.06	2382.23	3172.73
D_{45}	980	-	0.11	0.34	585	-	19.91	24.55	385	-	59.44	-	192	-	183.47	-
D_{46}	1000	0.00	0.00	0.00	1000	0.00	0.00	0.00	783	-	18.82	10.79	393	-	110.22	69.30
D_{47}	215	-	254.42	-	39	-	1296.10	-	22	-	2064.86	2704.55	14	-	2924.86	3864.29
D_{48}	1500	0.00	0.00	-	936	-	16.03	-	642	-	48.03	-	339	-	148.42	-
D_{49}	1500	0.00	0.00	0.00	1500	0.00	0.00	0.00	1230	-	14.88	-	650	-	92.97	-
D_{50}	1000	0.00	0.00	0.00	618	-	15.97	-	562	-	11.61	-	34	-	1540.19	-
D_{51}	200	0.00	0.00	0.00	200	0.00	0.00	0.00	132	32.67	32.67	11.09	111	40.34	40.34	5.85
D_{52}	200	0.00	0.00	0.00	200	0.00	0.00	0.00	160	9.45	9.45	3.27	138	12.88	12.87	2.26
D_{53}	200	0.00	0.00	0.00	200	0.00	0.00	0.00	200	0.00	0.00	0.00	200	0.00	0.00	0.00
D_{54}	200	0.00	0.00	0.00	200	0.00	0.00	0.00	200	0.00	0.00	0.00	200	0.00	0.00	0.00
D_{55}	200	0.00	0.00	0.00	200	0.00	0.00	0.00	200	0.00	0.00	0.00	200	0.00	0.00	0.00
D_{56}	400	0.00	0.00	0.00	248	15.32	15.32	5.71	224	11.77	11.77	3.99	21	960.31	960.30	917.76
D_{57}	400	0.00	0.00	0.00	400	0.00	0.00	0.00	250	40.05	40.05	15.92	211	47.55	47.55	10.79
D_{58}	400	0.00	0.00	0.00	400	0.00	0.00	0.00	255	37.30	37.30	13.65	215	44.81	44.80	8.64
D_{59}	400	0.00	0.00	0.00	400	0.00	0.00	0.00	265	32.12	32.12	11.18	222	40.24	40.24	6.01
D_{60}	400	0.00	0.00	0.00	400	0.00	0.00	0.00	400	0.00	0.00	0.00	400	0.00	0.00	0.00
D_{61}	200	0.00	0.00	0.00	196	1.53	1.53	1.07	70	149.07	149.07	109.07	33	369.96	369.88	256.77
D_{62}	200	0.00	0.00	0.00	200	0.00	0.00	0.00	200	0.00	0.00	0.00	196	1.53	1.53	0.78
D_{63}	400	0.00	0.00	0.00	50	473.19	473.19	422.24	25	903.60	903.60	830.65	13	1616.41	1616.24	1544.51
D_{64}	400	0.00	0.00	0.00	400	0.00	0.00	0.00	106	230.36	230.36	175.33	39	698.42	698.41	501.94

Table 11: CPU times on DIMACS graphs (in sec.)

ID	$\gamma = 0.5$			$\gamma = 0.7$			$\gamma = 0.8$			$\gamma = 0.9$		
	$T_{[C_\gamma]}$	$T_{[D_\gamma^*]}$	$T_{[C_\gamma]}$	$T_{[C_\gamma]}$	$T_{[D_\gamma^*]}$	$T_{[C_\gamma]}$	$T_{[C_\gamma]}$	$T_{[D_\gamma^*]}$	$T_{[C_\gamma]}$	$T_{[D_\gamma^*]}$	$T_{[C_\gamma]}$	$T_{[D_\gamma^*]}$
D_1	3.35	< 0.005	0.03	0.39	< 0.005	0.03	14.71	0.06	0.09	28.92	0.05	0.48
D_2	21.51	0.08	0.11	3.26	0.05	9.78	6.86	0.05	10.11	11.22	0.05	11.03
D_3	2.57	0.02	0.08	4.54	0.03	0.61	14.24	0.03	0.56	19.06	0.03	1.03
D_4	3.31	0.02	0.06	4.01	0.06	0.11	13.63	0.05	0.37	20.98	0.03	0.42
D_5	21.33	0.02	0.28	3.53	0.03	0.33	398.93	0.13	24.80	655.91	0.19	58.59
D_6	34.20	0.02	0.31	3.53	0.03	0.30	366.06	0.13	24.21	264.52	0.19	67.08
D_7	32.31	0.03	0.30	3.53	0.03	0.31	369.46	0.14	22.68	1029.45	0.20	46.15
D_8	32.78	0.03	0.31	3.67	0.02	0.31	324.50	0.13	36.60	385.29	0.20	66.5
D_9	464.48	0.09	5.69	5172.18	0.81	2782.22	—	0.89	4565.12	—	0.84	3222.22
D_{10}	428.80	0.08	5.62	4560.08	0.80	2328.41	—	0.86	2987.82	—	0.91	4226.71
D_{11}	419.67	0.08	5.77	5342.64	0.75	2465.00	—	1.03	3075.39	—	0.94	2893.40
D_{12}	483.92	0.08	5.83	5085.10	0.77	967.21	—	1.00	3012.10	—	1.02	2930.90
D_{13}	1.44	0.03	0.05	0.22	0.03	0.09	0.64	0.02	0.08	1.76	0.02	0.09
D_{14}	6.24	0.02	0.09	0.84	0.02	0.09	7.52	0.02	0.05	6.02	0.02	0.08
D_{15}	37.10	0.05	5.97	5.41	0.05	11.39	22.18	0.05	3.37	37.47	0.03	0.14
D_{16}	10.59	0.11	0.51	1.44	0.11	0.58	1.86	0.11	1.01	6.15	0.09	0.66
D_{17}	684.63	0.23	706.45	128.34	0.19	179.45	748.23	0.17	1.50	530.97	0.22	1.22
D_{18}	39.30	0.08	0.73	7.04	0.08	0.56	28.67	0.08	0.56	30.89	0.06	0.50
D_{19}	103.20	0.17	1.29	14.68	0.14	0.94	344.14	0.13	0.84	121.57	0.14	0.58
D_{20}	907.80	0.17	0.05	144.86	0.16	0.05	432.11	0.16	0.05	1568.37	0.16	0.05
D_{21}	697.12	0.13	4.06	121.14	0.13	3.98	352.89	0.13	3.88	—	2.03	2400.82
D_{22}	0.55	0.02	< 0.005	0.11	< 0.005	< 0.005	0.84	0.02	0.02	1.15	0.02	0.02
D_{23}	0.09	0.02	0.02	0.03	< 0.005	0.09	0.03	0.02	0.16	0.09	< 0.005	0.16
D_{24}	6.79	0.02	0.00	1.03	0.02	0.02	5.01	0.02	0.02	12.23	0.02	< 0.005
D_{25}	6.10	0.02	0.14	10.08	0.09	6.19	5.80	0.05	16.65	45.52	0.05	20.76
D_{26}	8.72	0.02	0.02	1.44	< 0.005	0.02	18.83	0.02	0.02	23.48	0.02	0.02
D_{27}	35.24	0.03	0.17	8.53	0.05	0.19	32.81	0.05	0.16	—	0.25	0.45
D_{28}	0.02	0.02	< 0.005	< 0.005	< 0.005	0.02	0.02	< 0.005	< 0.005	0.03	< 0.005	< 0.005
D_{29}	0.67	0.02	< 0.005	0.14	< 0.005	0.02	1.28	0.02	< 0.005	2.03	< 0.005	< 0.005
D_{30}	4.51	0.02	0.03	7.29	0.05	0.05	14.74	0.03	0.17	36.07	0.02	0.28
D_{31}	576.38	0.08	2.76	38.19	0.08	2.79	—	0.81	710.62	—	0.91	1025.27
D_{32}	17.50	0.03	< 0.005	3.73	0.03	< 0.005	13.15	0.05	0.02	34.21	0.03	0.02
D_{33}	1091.77	0.16	0.03	162.02	0.16	0.03	655.49	0.16	< 0.005	1785.99	0.16	0.02
D_{34}	0.03	0.02	< 0.005	0.03	< 0.005	< 0.005	0.05	< 0.005	< 0.005	0.11	< 0.005	< 0.005

Table 12: CPU times on DIMACS graphs (in sec.)

ID	$\gamma = 0.5$			$\gamma = 0.7$			$\gamma = 0.8$			$\gamma = 0.9$		
	$T_{[C_\gamma]}$	$T_{[D_\gamma^*]}$	$T_{[C_\gamma]}$	$T_{[C_\gamma]}$	$T_{[D_\gamma^*]}$	$T_{[C_\gamma]}$	$T_{[C_\gamma]}$	$T_{[D_\gamma^*]}$	$T_{[C_\gamma]}$	$T_{[D_\gamma^*]}$	$T_{[C_\gamma]}$	$T_{[D_\gamma^*]}$
D_{35}	14.23	0.19	56.92	7.75	0.17	1.67	9.80	0.11	1.23	18.08	0.11	0.83
D_{36}	127.59	0.42	25.83	69.89	0.16	79.52	44.03	0.14	93.56	62.76	0.13	81.73
D_{37}	8.92	0.02	0.38	12.55	0.02	0.80	157.88	0.09	25.09	205.91	0.13	43.80
D_{38}	144.52	0.59	802.59	87.38	0.69	6.98	101.23	0.41	5.44	147.20	0.41	5.06
D_{39}	49.06	0.03	9.86	655.58	0.53	285.92	685.61	0.59	589.22	626.69	0.55	428.13
D_{40}	70.56	0.05	2.28	30.25	0.03	2.53	1891.25	0.31	27.94	2032.66	0.24	132.73
D_{41}	559.87	1.72	3522.42	160.02	1.56	21.70	143.17	1.19	18.89	527.61	1.81	15.17
D_{42}	879.42	3.19	808.44	2564.84	1.39	3341.45	5765.97	1.50	5142.14	3406.08	1.86	2095.45
D_{43}	218.23	0.06	7.03	75.52	0.06	7.48	6168.50	0.80	316.52	—	0.83	552.95
D_{44}	4819.48	3.97	—	930.89	3.61	—	3174.97	2.70	97.44	3121.17	2.64	81.22
D_{45}	—	8.10	80.14	—	4.13	3268.00	—	3.67	—	—	4.06	—
D_{46}	2071.77	0.12	18.44	156.02	0.13	18.53	—	2.03	908.89	—	2.25	2858.16
D_{47}	—	21.86	—	—	19.49	—	—	11.80	514.92	—	10.64	360.02
D_{48}	1927.84	0.19	—	—	14.13	—	—	13.88	—	—	12.50	—
D_{49}	332.20	0.25	88.16	430.66	0.27	85.75	—	7.03	—	—	5.89	—
D_{50}	501.26	0.11	24.51	—	1.80	—	—	2.05	—	—	1.81	—
D_{51}	3.18	0.02	0.05	3.01	< 0.005	0.05	19.75	0.05	0.14	32.28	0.05	0.27
D_{52}	2.45	0.02	0.06	3.20	0.02	0.05	14.46	0.06	0.08	21.08	0.06	0.08
D_{53}	3.45	< 0.005	< 0.005	3.17	0.02	0.02	3.79	0.02	< 0.005	8.25	< 0.005	0.02
D_{54}	3.60	0.02	0.02	3.68	0.02	0.03	5.09	0.02	0.02	8.28	0.02	0.02
D_{55}	3.88	< 0.005	0.02	3.87	0.02	0.02	5.23	< 0.005	0.02	7.41	0.02	< 0.005
D_{56}	25.44	0.02	0.97	205.10	0.14	230.12	267.29	0.19	315.14	225.80	0.16	282.57
D_{57}	25.55	0.03	0.41	40.48	0.03	0.34	665.64	0.17	63.74	2105.47	0.16	102.88
D_{58}	30.30	0.03	0.41	27.86	0.02	0.41	609.03	0.14	72.77	2513.60	0.14	100.31
D_{59}	26.27	0.03	0.44	31.56	0.03	0.39	714.61	0.14	69.36	2467.72	0.14	80.61
D_{60}	31.68	0.03	0.08	25.77	0.02	0.08	32.73	0.02	0.08	59.50	0.03	0.05
D_{61}	3.32	< 0.005	0.05	5.79	0.05	0.06	20.90	0.05	0.25	18.77	0.03	0.28
D_{62}	3.39	0.02	0.03	4.77	0.02	< 0.005	5.16	0.02	0.02	14.20	0.05	< 0.005
D_{63}	24.43	0.02	0.97	204.69	0.16	232.10	251.02	0.14	209.34	202.55	0.16	201.34
D_{64}	29.30	0.03	0.44	38.61	0.02	0.42	823.79	0.22	71.26	535.68	0.14	64.24

Table 13: DIMACS instances not solved within time limit

ID	Branch-and-Price		MIPs	
	$OG_{[D_\gamma^s]}$ (%)	#Cols	$OG_{[C_\gamma]}$ (%)	$OG_{[\bar{C}_\gamma]}$ (%)
D_1	251.11	2930023	3180.00	218.79
D_2	40.00	4205293	2050.80	520.83
D_3	486.36	2843859	1791.43	692.60
D_4	278.93	2333289	2799.80	314.25
D_5	62.07	1232015	6540.00	2116.90
D_6	468.94	1100520	10998.67	2262.60
D_7	473.68	1338862	8175.00	2662.88
D_8	459.27	1046925	10966.67	2108.60
D_9	1455.18	299444	-	7900.00
D_{10}	1459.42	296895	-	7900.00
D_{11}	1496.80	310185	-	8788.89
D_{12}	1593.68	311608	-	7900.00
D_{17}	44.38	624741	6800.00	3383.33
D_{18}	27.58	82466	516.67	59.26
D_{19}	39.39	234664	44.81	58.21
D_{21}	23.31	257701	-	753.20
D_{25}	458.90	1103522	5865.67	1067.64
D_{26}	550.00	6089940	1110.38	240.00
D_{27}	580.68	986462	-	1146.43
D_{31}	457.65	397155	-	5230.33
D_{36}	24.59	1772277	1504.90	1591.89
D_{37}	71.71	1461076	1199.05	170.41
D_{38}	1016.67	764961	4528.67	3600.00
D_{39}	143.78	913014	13850.00	501.53
D_{40}	6.36	936627	20750.00	150.49
D_{41}	1472.96	361658	9549.50	4166.67
D_{42}	132.07	487322	19250.00	8650.00
D_{43}	94.89	481375	-	5647.38
D_{44}	1806.42	209762	49900.00	5900.00
D_{45}	176.61	258266	-	-
D_{46}	109.41	243074	-	941.67
D_{47}	2889.91	94781	-	7828.57
D_{48}	143.38	107727	-	-
D_{49}	92.92	73421	-	-
D_{50}	1535.27	307602	-	-
D_{56}	942.81	1448182	3599.33	1390.29
D_{57}	45.24	1177209	10233.33	2087.80
D_{58}	43.25	1225530	7650.00	2080.00
D_{59}	38.64	1370705	6100.00	1996.09
D_{61}	294.44	3045987	1812.13	312.24
D_{63}	461.11	1232796	7298.33	2482.13
D_{64}	628.57	1404281	10264.33	2601.88