The minimum quasi-clique partitioning problem: Complexity, formulations, and a computational study

Article *in* Information Sciences · August 2022

DOI: 10.1016/j.ins.2022.08.073

Rafael A. Melo
Universidade Federal da Bahia

Celso C. Ribeiro
Universidade Federal Fluminense

José Angel Riveaux Merino
Universidade Federal Fluminense

The minimum quasi-clique partitioning problem: Complexity, formulations, and a computational study

Rafael A. Melo[a], Celso C. Ribeiro[b], Jose A. Riveaux[b]

[a]Institute of Computing, Universidade Federal da Bahia, Salvador, 40170-115, BA, Brazil
[b]Institute of Computing, Universidade Federal Fluminense, Niterói, 24210-346, RJ, Brazil

Abstract

Given a simple graph $G = (V, E)$ and a real constant $\gamma \in (0, 1]$, a γ-clique (or γ-quasi-clique) is a subset $V' \subseteq V$ inducing a subgraph with edge density at least γ. The minimum quasi-clique (or γ-clique) partitioning problem (MQCPP) consists in partitioning the vertices of the graph in γ-cliques to minimize the number of elements in the partition. In this paper, we formally introduce the minimum quasi-clique partitioning problem, which has not yet been addressed in the literature from an optimization point of view. We show by using a reduction from the unweighted maximum cut problem that even deciding whether a graph can be partitioned into two γ-cliques is NP-complete. This result contrasts with that of the clique partitioning problem, whose decision version is polynomially solvable for bipartition. We propose four integer programming formulations and a multi-start greedy randomized heuristic to provide initial feasible solutions for MQCPP. Computational experiments show that two formulations that employ the principles of representatives outperform the others regarding the best-obtained solutions and the number of instances solved optimally within the imposed time limit. Furthermore, the results also demonstrate that the instances with medium values of γ are more challenging for the proposed formulations than those with larger or lower values.

Keywords: quasi-clique partitioning, maximum quasi-clique, computational complexity, integer programming, combinatorial optimization, network clustering

1. Introduction

1.1. Preliminaries

Let $G = (V, E)$ be a simple undirected graph with vertex set $V = \{v_1, \ldots, v_n\}$ and edge set $E \subset V \times V$. G is a complete graph if an edge in E connects every two different vertices in V. A clique is a complete subgraph of a graph. The complement of G is the graph $\bar{G} = (V, \bar{E})$ with $\bar{E} = \{uv \mid uv \notin E\}$. Denote by anti-edge (or non-edge) of G an edge in $\bar{E}$. Besides, given a subset $S \subseteq V$, define $\bar{S} = V \setminus S$.

Email addresses: rafael.melo@ufba.br (Rafael A. Melo), celso@ic.uff.br (Celso C. Ribeiro), jangel.riveaux@gmail.com (Jose A. Riveaux)

The graph $G[S]$ induced in G by $S \subseteq V$ is that with vertex set S and edge set formed by all edges of E with both endpoints in S. For any $S \subseteq V$, the subset $E(S) \subseteq E$ is formed by all edges of G with both endpoints in S. Also, consider $W \subseteq V$ such that $S \cap W = \emptyset$, and denote by $E(S, W)$ the set of all edges in G with one endpoint in S and another in W. A partition of a graph (or graph partition) is a partition of its vertex set.

The density of a graph G is defined as $d(G) = \frac{|E|}{|V|\cdot(|V|-1)/2} = \frac{2|E|}{|V|\cdot(|V|-1)}$. The degree of a vertex $v \in V$, $deg_G(v)$, is given by the number of vertices adjacent to v in G. Given a graph $G = (V, E)$ and a threshold $\gamma \in (0, 1]$, a γ-clique (or γ-quasi-clique) is any subset $C \subseteq V$ such that the density of $G[C]$ is greater than or equal to γ. A γ-clique C is maximal if there is no other γ-clique C' such that $C \subset C'$. The minimum quasi-clique (or γ-clique) partitioning problem (MQCPP) consists in finding the minimum number $\mathcal{K}$ of quasi-cliques in which G can be partitioned.

1.2. Literature review

Problems related to finding dense subgraphs in networks have appeared in several applications, including telecommunications [2], biology [41], social networks [40], and as approaches for solving other challenging combinatorial optimization problems [5]. Among the critical problems of obtaining large dense subgraphs, we can highlight the maximum clique problem and the maximum quasi-clique problem. Besides, partitioning graphs into dense subgraphs finds applications in several areas, such as bioinformatics [18], quantum computing [45], data mining [16], and community detection [13, 47, 48]. The minimum quasi-clique partitioning problem lies in this family of problems.

The maximum clique problem is a classical NP-hard problem that consists in obtaining a maximum cardinality clique in a graph. Theoretical and computational aspects of the problem and its variants have been extensively studied throughout the last decades [6, 23, 46]. The clique partitioning problem and its derivations also received a lot of attention in the literature [7, 10, 17, 32]. We notice that the clique partitioning problem is strongly related to graph coloring [19], as a clique partition in a graph determines a vertex coloring [24] in its complement. However, the clique concept may be considered too restrictive for applications such as network analysis [33] and, thus, different clique relaxations have been studied, such as quasi-cliques.

Problems consisting of obtaining maximum quasi-cliques, or at least maximal ones, attracted the attention of several researchers during the last decades. Abello et al. [2] proposed an approach that employs preprocessing and a greedy randomized adaptive search procedure (GRASP) metaheuristic for obtaining quasi-cliques in massive sparse graphs. Tsourakakis, Bonchi, Gionis, Gullo, and Tsiarli [42] defined the problem of obtaining optimal quasi-cliques. They proposed a greedy approximation algorithm and a local search heuristic for obtaining such structures. Pattillo, Veremyev, Butenko, and Boginski [34] studied the maximum quasi-clique problem (MQCP). The authors showed that the problem is NP-hard for any $0 < \gamma < 1$, proposed analytical upper bounds, and presented integer programming formulations. Veremyev, Prokopyev, Butenko, and Pasiliao [43] proposed new integer programming formulations for MQCP and compared them with those presented in Pattillo et al. [34], both theoretically and computationally. Pinto, Ribeiro, Rosseti, and Plastino [36] elaborated a biased random-key genetic algorithm (BRKGA) for MQCP. Ribeiro and Riveaux [38] provided an exact backtracking-based algorithm for MQCP. Zhou, Benlic, and Wu [49] described a memetic algorithm for MQCP that employs tabu search for local improvement and opposition-based learning (OBL) to enhance its search mechanism. Pinto, Ribeiro, Riveaux, and Rosseti [37] proposed a BRKGA metaheuristic with an exact local search strategy that combines the methods presented in Pinto et al. [36] and Ribeiro and Riveaux [38]. Marinelli, Pizzuti, and Rossi [25] proposed a new formulation for MQCP with an exponential number of variables to provide strong linear relaxation bounds using column generation. Peng, Wu, Wang, and Wu [35] devised an artificial bee colony algorithm for MQCP. The problem of obtaining maximal degree-based quasi-cliques, i.e., those in which each vertex has a degree of at least $\gamma(|V|-1)$, was addressed in Sanei-Mehri, Das, Hashemi, and Tirthapura [39]. The authors proved the NP-completeness of determining if a given degree-based quasi-clique is maximal and proposed a heuristic to the problem of obtaining the k largest degree-based quasi-cliques in a graph.

Analogously to the clique partitioning problem, which involves partitioning the graph into cliques, the quasi-clique partitioning problem consists of partitioning the graph into quasi-cliques. As observed in the literature, the problem of clustering networks can be modeled as minimum clique partition problems since all elements in the cluster are tightly related. However, due to the restrictiveness of cliques, it may be worth modeling clusters by k-cores (maximal subgraphs with degree at least k) or clique relaxations such as s-plexes (subgraphs in which every vertex is nonadjacent to at most $s-1$ other vertices), s-clubs (sets of vertices inducing subgraphs of diameter at most s), or quasi-cliques [22, 44]. To the best of our knowledge, differently from the minimum clique partitioning, the minimum quasi-clique partitioning problem was not yet formally addressed in the literature despite its potentially significant applicability. As far as we know, the only work considering the quasi-clique partitioning of a graph is Basu, Sengupta, Maulik, and Bandyopadhyay [4]. The authors proposed a game-theoretical approach for partitioning a graph into (λ, γ)-cliques (i.e., quasi-cliques characterized not only by their densities but also by the degrees of their vertices) associated with communities. However, the problem is not analyzed from an optimization point of view. Instead, the partitions are evaluated according to network quality measures regarding the obtained communities.

1.3. Contributions and organization

The principal contributions of our paper can be summarized as follows. Firstly, we formally introduce the minimum quasi-clique partitioning problem (MQCPP) and show that it is NP-hard even when the optimal solution is equal to two, i.e., when the instance admits a partition into two quasi-cliques. Secondly, we present and compare computationally four compact integer programming formulations (with polynomial numbers of variables and constraints) for the problem. Thirdly, we propose a multi-start greedy randomized heuristic to provide initial feasible solutions for the formulations.

The remainder of this paper is organized as follows. The computational complexity of MQCPP is addressed in Section 2. Integer programming formulations are presented in

Section 3. Section 4 shows the proposed multi-start greedy randomized heuristic. The computational experiments are summarized in Section 5. Concluding remarks are discussed in the last section.

2. Computational complexity

In this section, we show that the minimum quasi-clique partitioning problem (MQCPP) is NP-hard even for a very restrictive case, namely, when the number of quasi-cliques in the optimal solution is equal to two. To do so, consider the decision version of MQCPP, which can be defined as follows:

Quasi-clique partitioning problem (QCPP)
Instance: Graph $G = (V, E)$, threshold value $\gamma \in (0, 1]$, and an integer $\mathcal{K} \geq 2$.
Question: Is there a partition of G into $\mathcal{K}$ γ-cliques?

Observe that, when $\gamma = 1$, QCPP corresponds to the well-known clique partitioning problem (CPP), which was shown to be NP-complete for $\mathcal{K} > 2$ [21]. Thus, QCPP is also NP-complete for $\mathcal{K} > 2$. It should be noticed, however, that CPP can be solved in polynomial time (and thus is in P) for $\mathcal{K} = 2$, since it amounts to checking whether the complement $\bar{G}$ of G is bipartite or not. In the following, we show that QCPP is NP-complete when $\gamma \in (0, 1)$ even for $\mathcal{K} = 2$. In this direction, consider the following particular case of QCPP:

Quasi-clique bipartitioning problem (QCBP)
Instance: Graph $G = (V, E)$ and threshold value $\gamma \in (0, 1)$.
Question: Is there a partition of G into two γ-cliques?

In what follows, Theorem 1 shows that QCBP is NP-complete. The proof uses a reduction from the NP-complete decision version of the unweighted maximum cut problem [15], also known as the simple maximum cut problem, which can be stated as:

Unweighted maximum cut problem (MCP)
Instance: Graph $G = (V, E)$ and integer L.
Question: Is there $P \subset V$ such that $|E(P, \bar{P})| \geq L$?

Theorem 1. *QCBP is NP-complete.*

Proof. We first show that QCBP belongs to NP. Given any solution for QCBP, represented by the partition $(S, \bar{S})$ with $S \subset V$ and $S \neq \emptyset$, the densities of the subgraphs $G[S]$ and $G[\bar{S}]$ induced in G by S and $\bar{S}$, respectively, can be computed in linear-time $O(|V| + |E|)$. Therefore, QCBP is NP.

We present a polynomial transformation from the unweighted maximum cut problem to the quasi-clique bipartitioning problem, i.e., MCP $\propto$ QCBP. Therefore, as the former is known to be NP-complete, this implies that the latter is also NP-complete, as we already showed it belongs to NP.

The polynomial transformation works as follows. Given an input graph $G = (V, E)$ for MCP, we generate for QCBP an input graph $G' = (V', E')$ with $2n(n^2 + 1) = 2n^3 + 2n$ vertices such that:

$$
\begin{aligned}
V' &= \{w_1^0, \ldots, w_1^{n^2}\} \cup \ldots \cup \{w_n^0, \ldots, w_n^{n^2}\} \cup \{\breve{w}_1^0, \ldots, \breve{w}_1^{n^2}\} \cup \ldots \cup \{\breve{w}_n^0, \ldots, \breve{w}_n^{n^2}\}
\end{aligned}
$$

and

$$
\begin{aligned}
E' &= \{w_h^i w_k^j, \breve{w}_h^i \breve{w}_k^j : \ h, k = 1, \ldots, n; \ j = 0, \ldots, n^2; \ i = 1, \ldots, n^2; \ \ \text{with } i \neq j \text{ or } h \neq k\} \\
&\cup \{w_h^i \breve{w}_k^j, \breve{w}_h^i w_k^j : \ h, k = 1, \ldots, n : k \neq h; \ j = 0, \ldots, n^2; \ i = 1, \ldots, n^2\} \\
&\cup \{w_h^0 \breve{w}_k^0, \breve{w}_h^0 w_k^0 : \ h, k = 1, \ldots, n : v_h v_k \in E\}.
\end{aligned}
$$

We observe that, for every $h = 1, \ldots, n$, each vertex $w_h^i \in V'$, with $i = 1, \ldots, n^2$, is adjacent to all other vertices in V', except for vertices $\breve{w}_h^j \in V'$, with $j = 0, \ldots, n^2$. Additionally, notice that $\{w_h^0 : h = 1, \ldots, n\}$ and $\{\breve{w}_h^0 : h = 1, \ldots, n\}$ are independent sets. Observe that G' can be obtained from G in polynomial time as it has $O(n^3)$ vertices and $O(n^6)$ edges. This construction is illustrated in Example 1.

To finalize the polynomial transformation, given the integer L for MCP, define γ for QCBP as

$$
\gamma = \frac{2\left[\binom{n^3}{2} + n^4 + L\right]}{n(n^2 + 1)[n(n^2 + 1) - 1]}. \tag{1}
$$

Propositions 1 and 2, which will be detailed in the following, show that there is a cut of size greater than or equal to L in $G = (V, E)$ if and only if there is a γ-clique bipartition in $G' = (V', E')$. Thus, implying that QCBP is NP-complete. □

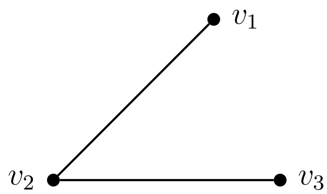
Example 1. Figure 1(a) exemplifies an input graph for MCP, while Figure 1(b) shows the corresponding graph G' constructed from the one in Figure 1(a). To simplify the illustration, the clique formed by the vertices $\{w_h^i : i = 1, \ldots, n^2\}$ is represented by a single super-vertex W_h in Figure 1(b), for every $h = 1, \ldots, n$. Analogously, the clique formed by the vertices $\{\breve{w}_h^i : i = 1, \ldots, n^2\}$, for every $h = 1, \ldots, n$, is represented by a unique super-vertex $\breve{W}_h$. △

Proposition 1. *Given $G = (V, E)$ and $G' = (V', E')$ as defined in Theorem 1, if there exists a cut of $G = (V, E)$ with size greater than or equal to L, then there exists a γ-clique bipartitioning of $G' = (V', E')$.*

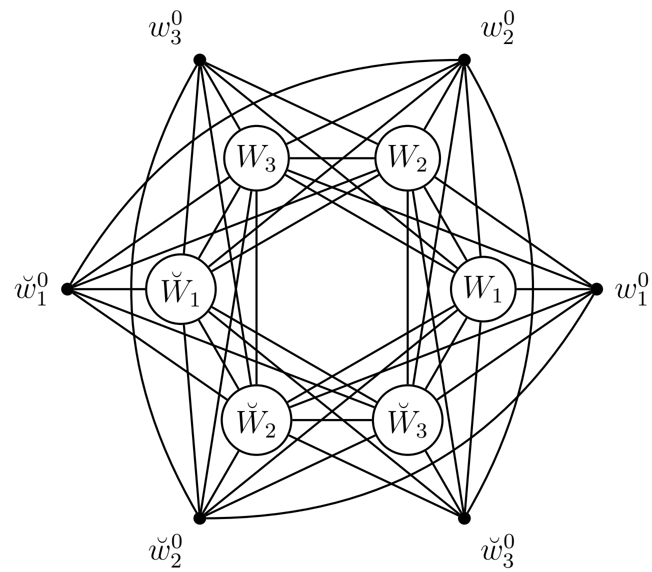
Proof. Assume there is a subset $P \subset V$ defining a cut $(P, \bar{P})$ for which $|E(P, \bar{P})| \geq L$. Define the bipartition $(S, \bar{S})$, with

$$
S = \{w_h^0, \ldots, w_h^{n^2} : h = 1, \ldots, n; \ v_h \in P\} \cup \{\breve{w}_h^0, \ldots, \breve{w}_h^{n^2} : h = 1, \ldots, n; \ v_h \in \bar{P}\}.
$$

Notice that $|S| = n(n^2 + 1)$. Therefore, given the construction of G', $S \setminus \{w_h^0, \breve{w}_h^0 : h = 1, \ldots, n\}$ induces a clique K_{n^3} with n^3 vertices in G' as the anti-edge associated with the pair w_h^i and $\breve{w}_h^i$ is not present for any $i \in \{1, \ldots, n^2\}$ and $h \in \{1, \ldots, n\}$. Thus, the vertices of such a set are all pairwise connected in G', guaranteeing the existence of $\binom{n^3}{2}$ edges in $E'(S)$. Also due to the construction of G', all the vertices in the subsets $W^0 = \{w_h^0 : h = 1, \ldots, n; v_h \in P\}$



(a) $G = (V, E)$



(b) $G' = (V', E')$

Figure 1: Example of the two graphs involved in the polynomial transformation MCP $\propto$ QCBP.

and $\breve{W}^0 = \{\breve{w}_h^0 : h = 1, \ldots, n; v_h \in \bar{P}\}$ are adjacent to all the vertices in K_{n^3}, ensuring the existence of $n \cdot n^3 = n^4$ edges in $E'(S)$. Moreover, if there are at least L edges between P and $\bar{P}$ in G, then there must be L or more edges in $E'(W^0 \cap S, \breve{W}^0 \cap S)$, corresponding to the edges $w_h^0 \breve{w}_{h'}^0$ such that $v_h v_{h'} \in E$. Consequently, as these sets of guaranteed edges are disjoint, $|E'(S)| \geq \binom{n^3}{2} + n^4 + L$, implying that

$$d(G'[S]) = \frac{2|E'(S)|}{|S| \cdot (|S| - 1)} \geq \frac{2\left[\binom{n^3}{2} + n^4 + L\right]}{n(n^2+1)[n(n^2+1) - 1]}.$$

Notice that using the same arguments in a symmetric way, it follows that $|\bar{S}| = |S|$ and $|E'(\bar{S})| = |E'(S)|$. Hence,

$$d(G'[\bar{S}]) = \frac{2|E'(\bar{S})|}{|\bar{S}| \cdot (|\bar{S}| - 1)} = \frac{2|E'(S)|}{|S| \cdot (|S| - 1)} \geq \frac{2[\binom{n^3}{2} + n^4 + L]}{n(n^2+1)[n(n^2+1) - 1]}.$$

Therefore, $G'[S]$ and $G'[\bar{S}]$ are γ-cliques. This property is illustrated in Example 2.

$\square$

Example 2. Consider again the graph illustrated in Figure 1(a), for which $n = 3$, and assume $L = 2$. We show that if there exists a cut of G with size two, then equation (1) implies that there will be a γ-clique bipartitioning in G' for

$$\gamma = \frac{2 \cdot \left[\binom{3^3}{2} + 3^4 + 2\right]}{3 \cdot (3^2+1) \cdot [3 \cdot (3^2+1) - 1]} = \frac{868}{870}.$$

Notice that the set $P = \{v_1, v_3\}$ defining the cut $(\{v_1, v_3\}, \{v_2\})$, illustrated in Figure 2, is a solution for MCP. Given the construction of G', taking the partition $(\{v_1, v_3\}, \{v_2\})$, we obtain $S = \{w_1^0, w_3^0, \breve{w}_2^0\} \cup W_1 \cup W_3 \cup \breve{W}_2$ and $\bar{S} = \{\breve{w}_1^0, \breve{w}_3^0, w_2^0\} \cup \breve{W}_1 \cup \breve{W}_3 \cup W_2$, with their corresponding edges illustrated in the continuous lines of Figures 3(a) and 3(b), respectively. Both the induced subgraphs $G'[S]$ and $G'[\bar{S}]$ have 30 vertices and $\binom{3^3}{2} + 3^4 + 2 = 434$ edges. Thus, their densities are equal to $\frac{2 \cdot 434}{30 \cdot 29} = \frac{868}{870} = \gamma$.

$\triangle$

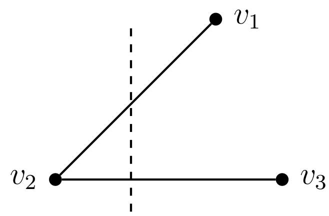
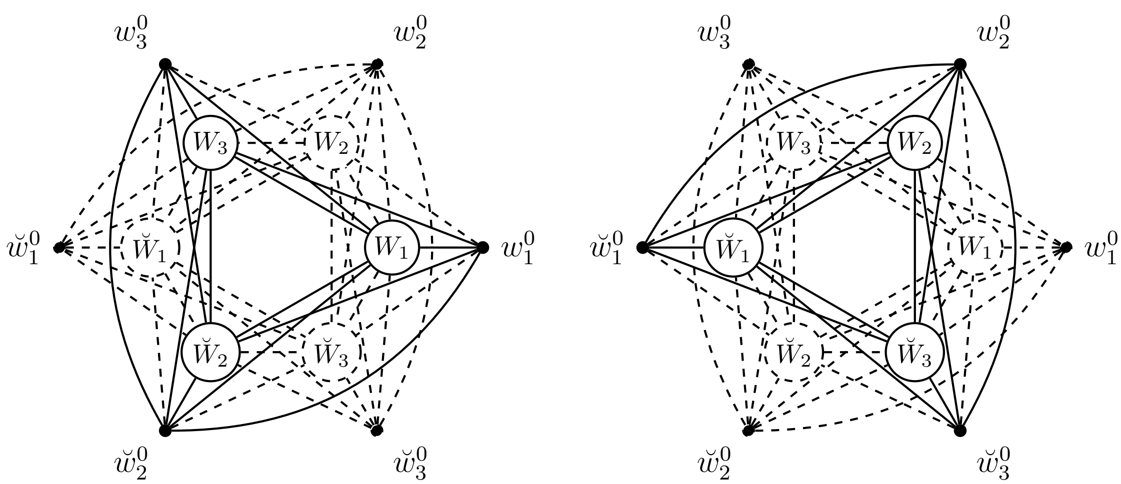


Figure 2: Partition $(\{v_1, v_3\}, \{v_2\})$ defining a cut for G.



(a) $G'[S]$

(b) $G'[\bar{S}]$

Figure 3: Bipartition of graph G'.

Proposition 2. *Given $G = (V, E)$ and $G' = (V', E')$ as defined in Theorem 1, if there exists a bipartition of G' in γ-cliques S and $\bar{S}$, then there is a cut of $G = (V, E)$ with size greater than or equal to L.*

Proof. Let $S \subset V'$ and denote by $(S, \bar{S})$ a bipartition of V' such that $G'[S]$ and $G'[\bar{S}]$ are γ-cliques. Without loss of generality, suppose that $|S| \geq |V'|/2$. Since G' has $2n(n^2 + 1)$ vertices and because of the quasi-hereditary property of γ-cliques [34], S must contain a γ-clique S' of G' with size $|V'|/2 = n(n^2 + 1) = n^3 + n$. Consequently, equation (1) implies that the number of edges in $G'[S']$ is greater than or equal to $\binom{n^3}{2} + n^4 + L$.

Notice that, since $|S'| = n^3 + n$ and the number of edges in $G'[S']$ is greater than or equal to $\binom{n^3}{2} + n^4 + L$, the maximum number of anti-edges in $G'[S']$ is

$$\binom{n^3+n}{2} - \left[\binom{n^3}{2} + n^4 + L\right] = \binom{n}{2} - L.$$

Lemmas 1, 2, 3, and 4, which will be detailed next, show, respectively, that:

(a) S' does not simultaneously contain one vertex from W_h and another from $\breve{W}_h$ for any $h = 1, \ldots, n$;

(b) for every $h = 1, \ldots, n$, $W_h \cap S' \neq \emptyset$ and $|W_h \cap S'| \geq n^2 - n$ if and only if $\breve{W}_h \cap S' = \emptyset$;

(c) $|(W_1 \cup \ldots \cup W_n \cup \breve{W}_1 \cup \ldots \cup \breve{W}_n) \cap S'| \leq n^3$; and

(d) for every $h = 1, \ldots, n$, either $W_h \cup \{w_h^0\} \subset S'$ or $\breve{W}_h \cup \{\breve{w}_h^0\} \subset S'$.

Notice that the construction of G' together with point (d) (given by Lemma 4) imply that, for every $h = 1, \ldots, n$, exactly one of the sets $W_h \cup \{w_h^0\}$ or $\breve{W}_h \cup \{\breve{w}_h^0\}$ is in S'. Thus, every one of the n^3 vertices in $(W_1 \cup \ldots \cup W_n \cup \breve{W}_1 \cup \ldots \cup \breve{W}_n) \cap S'$ is adjacent to every other vertex in S' (we recall that there are $n^3 + n$ of them), ensuring $\binom{n^3}{2} + n^4$ of the edges in $G'[S']$. Because $G'[S']$ has at least $\binom{n^3}{2} + n^4 + L$ edges, L or more edges must exist between the n vertices of $(w_1^0, \ldots, w_n^0, \breve{w}_1^0, \ldots, \breve{w}_n^0) \cap S$. Since both sets $\{w_h^0 : h = 1, \ldots, n\}$ and $\{\breve{w}_h^0 : h = 1, \ldots, n\}$ are independent, these edges can only exist between the vertices of $\{w_h^0 : h = 1, \ldots, n\} \cap S'$ and $\{\breve{w}_h^0 : h = 1, \ldots, n\} \cap S'$, and correspond to those built from the edges in the input graph G of MCP. Consequently, considering $P = \{v_h \in V : w_h^0 \in S'\}$, it follows that the partition $(P, \bar{P})$ defines a cut with at least L edges in G for MCP. $\square$

Lemma 1. *Let S' be defined as in Proposition 2. S' does not simultaneously contain one vertex from W_h and another from $\breve{W}_h$, for any $h = 1, \ldots, n$.*

Proof. Observe that the $2n$ vertices $w_1^0, \ldots, w_n^0, \breve{w}_1^0, \ldots, \breve{w}_n^0$ are the only vertices of V' that do not belong to $W_1 \cup \ldots \cup W_n \cup \breve{W}_1 \cup \ldots \cup \breve{W}_n$. This implies that $|(W_1 \cup \ldots \cup W_n \cup \breve{W}_1 \cup \ldots \cup \breve{W}_n) \cap S'| \geq n^3 - n$. Notice that the set $(W_1 \cup \ldots \cup W_n \cup \breve{W}_1 \cup \ldots \cup \breve{W}_n) \cap S'$ can be seen as the union of subsets $(W_i \cup \breve{W}_i) \cap S'$ for $i = 1, \ldots, n$, as $W_1, \ldots, W_n, \breve{W}_1, \ldots, \breve{W}_n$ are disjoint sets. Let $\Gamma \subseteq \{1, \ldots, n\}$ contain every index ℓ such that S' contains vertices from both W_ℓ and $\breve{W}_\ell$. Besides, let $\bar{\Gamma} = \{1, \ldots, n\} \setminus \Gamma$ be the set containing every index ℓ such that S' contains vertices from W_ℓ if and only if S' does not contain vertices from $\breve{W}_\ell$.

Firstly, we show that if $\Gamma \neq \emptyset$, then there must be at least one $\ell' \in \Gamma$ for which $|(W_{\ell'} \cup \breve{W}_{\ell'}) \cap S'| \geq n^2 - n$. Suppose by contradiction that $|(W_\ell \cup \breve{W}_\ell) \cap S'| < n^2 - n$ for every $\ell \in \Gamma$. Let $p = |\bar{\Gamma}|$ and $q = |\Gamma|$, and notice that $p + q = n$. Thus, the number of vertices in $(W_1 \cup \ldots \cup W_n \cup \breve{W}_1 \cup \ldots \cup \breve{W}_n) \cap S'$ is at most $pn^2 + q(n^2 - n - 1)$. To see why notice that there can be at most n^2 vertices for each $\ell \in \bar{\Gamma}$, which are the n^2 vertices of the corresponding set (either W_ℓ or $\breve{W}_\ell$), and at most $(n^2 - n - 1)$ vertices for each $\ell \in \Gamma$, given the supposition. However, as $q = |\Gamma| \geq 1$, it follows that $pn^2 + q(n^2 - n - 1) \leq (n-1)n^2 + (n^2 - n - 1) =$

$n^3 - n - 1 < n^3 - n$. Therefore, we reach a contradiction as there must exist $\ell' \in \Gamma$ such that $|(W_{\ell'} \cup \breve{W}_{\ell'}) \cap S'| \geq n^2 - n$.

We now show that if $\Gamma \neq \emptyset$, then the number of anti-edges in $G'[S']$ is strictly greater than $\binom{n}{2} - L$, implying that $G'[S']$ cannot be a γ-clique. Recall that, given the construction of G', there are no edges between the vertices of $W_{\ell'}$ and those of $\breve{W}_{\ell'}$, i.e., $E'(W_{\ell'}, \breve{W}_{\ell'}) = \emptyset$. All the anti-edges between the vertices of $W_{\ell'} \cap S'$ and $\breve{W}_{\ell'} \cap S'$ will be anti-edges in $G'[S']$, and their number can be computed as $|W_{\ell'} \cap S'| \cdot |\breve{W}_{\ell'} \cap S'|$. Since $|(W_{\ell'} \cup \breve{W}_{\ell'}) \cap S'| \geq n^2 - n$, the number of anti-edges in $G'[S']$ between the vertices of $W_{\ell'}$ and $\breve{W}_{\ell'}$ is at least $\{\min\ xy \mid x + y \geq n^2 - n,\ x \geq 1,\ y \geq 1\}$, whose solution is $n^2 - n - 1$ and is achieved when either $x = 1$ or $y = 1$ (this can be checked by inspection). This implies that there are at least $n^2 - n - 1 > \binom{n}{2} - L$ anti-edges in $G'[(W_h \cup \breve{W}_h) \cap S']$. Thus, Γ has to be empty, and the result follows. □

Lemma 2. *Let S' be defined as in Proposition 2. For every $h = 1, \ldots, n$, $W_h \cap S' \neq \emptyset$ and $|W_h \cap S'| \geq n^2 - n$ if and only if $\breve{W}_h \cap S' = \emptyset$.*

Proof. If $W_h \cap S' \neq \emptyset$ then, by Lemma 1, no vertex of $\breve{W}_h$ belongs to S', i.e., $\breve{W}_h \cap S' = \emptyset$. On the other hand, if $\breve{W}_h \cap S' = \emptyset$ then, for every $\ell = 1, \ldots n$ with $\ell \neq h$, at most n^2 vertices of either W_ℓ or $\breve{W}_\ell$ may belong to S', adding up to a total of at most $n^2(n-1) = n^3 - n^2$ vertices. Therefore, as $|S'| = n^3 + n$, at least $n^2 - n$ vertices of W_h must belong to S', i.e., $S' \cap W_h \neq \emptyset$. □

Lemma 3. *Let S' be defined as in Proposition 2. $|(W_1 \cup \ldots \cup W_n \cup \breve{W}_1 \cup \ldots \cup \breve{W}_n) \cap S'| \leq n^3$.*

Proof. We know from Lemma 2 that, for each $h = 1, \ldots, n$, only vertices of either W_h or $\breve{W}_h$ may belong to S'. Since the sets W_h and $\breve{W}_h$ have exactly n^2 vertices each, the number of vertices in $|(W_1 \cup \ldots \cup W_n \cup \breve{W}_1 \cup \ldots \cup \breve{W}_n) \cap S'|$ is not greater than $n \cdot n^2 = n^3$. □

Lemma 4. *Let S' be defined as in Proposition 2. For every $h = 1, \ldots, n$, either $W_h \cup \{w_h^0\} \subset S'$ or $\breve{W}_h \cup \{\breve{w}_h^0\} \subset S'$.*

Proof. Notice that Lemma 2 implies that either $|W_h \cap S'| \geq n^2 - n$ or $|\breve{W}_h \cap S'| \geq n^2 - n$. Assume by contradiction that there is some $h \geq 1$ such that either $w_h^0 \in S'$ and $\breve{W}_h \cap S' \neq \emptyset$ or $\breve{w}_h^0 \in S'$ and $W_h \cap S' \neq \emptyset$. This, together with the fact that $E'(w_h^0, \breve{W}_h) = E'(\breve{w}_h^0, W_h) = \emptyset$, implies that the number of anti-edges in $G[S']$ is at least $n^2 - n > \binom{n}{2} - L$. Therefore, the number of edges in $G'[S']$ is strictly smaller than $\binom{n^3}{2} + n^4 + L$, which contradicts the fact that $G'[S']$ is a γ-clique. Moreover, given Lemma 3 and the fact that $|S'| = n(n^2 + 1) = n^3 + n$, it follows that for every $h = 1, \ldots, n$ either $W_h \cup \{w_h^0\} \subset S'$ (contributing with $n^2 + 1$ vertices each) or $\breve{W}_h \cup \{\breve{w}_h^0\} \subset S'$ (contributing with $n^2 + 1$ vertices each). □

As QCBP is a special case of QCPP, Corollary 1 holds as a consequence of Theorem 1.

Corollary 1. *QCPP is NP-complete.*

Thus, Corollary 1 implies that the minimum quasi-clique partition problem (MQCPP) is NP-hard even when its optimal solution is equal to two.

3. Integer programming formulations

Integer programming approaches have been successfully applied recently to various challenging graph-related problems [11, 26, 29]. This section presents four compact integer programming formulations for the minimum quasi-clique partitioning problem (MQCPP). The first formulation, presented in Section 3.1, models quasi-cliques and graph partitions in natural ways, following ideas similar to those used in, respectively, Pattillo et al. [34] and Méndez-Díaz and Zabala [31]. The second formulation, described in Section 3.2, relies on the idea of quasi-clique size decompositions [43]. The third formulation, detailed in Section 3.3, uses the principles of formulations by representatives [8, 14] to model partitions in an attempt to reduce symmetry. The fourth formulation, given in Section 3.4, combines the ideas of quasi-clique size decompositions and formulations by representatives.

In the remainder of the section, denote the vertex set of $G = (V, E)$ by $V = \{1, \ldots, n\}$ for the sake of notation simplification. Additionally, define UB as any valid upper bound on the number of quasi-cliques in an optimal partition. Note that any feasible solution for the problem can be used to determine UB. Besides, define UB_k as an upper bound on the size of any quasi-clique in the graph. The following bound can be used [34]:

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

3.1. Standard formulation

A standard (or natural) integer programming formulation for MQCPP can be established using the decision variables described in the following. For every $i \in \{1, \ldots, UB\}$, define the binary variable

$$y_i = \begin{cases} 1, & \text{if a quasi-clique indexed by } i \text{ belongs to the partition,} \\ 0, & \text{otherwise.} \end{cases}$$

Besides, for every $i \in \{1, \ldots, UB\}$ and $v \in V$, consider the binary variable

$$x_{iv} = \begin{cases} 1, & \text{if vertex } v \text{ belongs to the quasi-clique } i, \\ 0, & \text{otherwise.} \end{cases}$$

Furthermore, for any $i \in \{1, \ldots, UB\}$ and $u, v \in V$, with $u < v$, define the binary variable

$$w_{iuv} = \begin{cases} 1, & \text{if vertices } u \text{ and } v \text{ are both in quasi-clique } i, \\ 0, & \text{otherwise.} \end{cases}$$

Thus, MQCPP can be formulated as the following integer program:

$$(STD) \quad \min \sum_{i=1}^{UB} y_i \tag{3}$$

$$\sum_{i=1}^{UB} x_{iv} = 1, \qquad \forall\, v \in V, \tag{4}$$

$$x_{iv} \le y_i, \qquad \forall\, i = 1, \dots, UB, \quad v \in V, \tag{5}$$

$$x_{iu} + x_{iv} \le w_{iuv} + 1, \qquad \forall\, i = 1, \dots, UB, \quad u, v \in V : u < v, \tag{6}$$

$$w_{iuv} \le x_{iu}, \qquad \forall\, i = 1, \dots, UB, \quad u, v \in V : u < v, \tag{7}$$

$$w_{iuv} \le x_{iv}, \qquad \forall\, i = 1, \dots, UB, \quad u, v \in V : u < v, \tag{8}$$

$$\sum_{u \in V} \sum_{\substack{v \in V \\ u<v, uv \in E}} w_{iuv} \ge \gamma \cdot \sum_{u \in V} \sum_{\substack{v \in V \\ u<v}} w_{iuv}, \qquad \forall\, i = 1, \dots, UB, \tag{9}$$

$$y_i \ge y_{i+1}, \qquad \forall\, i = 1, \dots, UB - 1, \tag{10}$$

$$y_i \in \{0, 1\}, \qquad \forall\, i = 1, \dots, UB, \tag{11}$$

$$x_{iv} \in \{0, 1\}, \qquad \forall\, i = 1, \dots, UB, \quad v \in V, \tag{12}$$

$$w_{iuv} \in \{0, 1\}, \qquad \forall\, i = 1, \dots, UB, \quad u, v \in V : u < v. \tag{13}$$

The objective function (3) minimizes the number of quasi-cliques in the partition. Constraints (4) establish that each vertex belongs to exactly one quasi-clique. Constraints (5) guarantee that a vertex may belong to a quasi-clique only if this quasi-clique belongs to the partition. Constraints (6)-(8) enforce that if both vertices u and v are part of the quasi-clique indexed by i, then the existence or not of the edge between them will affect the density of this quasi-clique. Next, constraints (9) enforce that the density of every quasi-clique should be greater than or equal to γ. Constraints (10) are symmetry breaking constraints. Finally, constraints (11)-(13) ensure the integrality requirements of the variables.

We remark that whenever one wants to avoid equality constraints in the formulation, constraints (4) can be replaced by

$$\sum_{i=1}^{UB} x_{iv} \ge 1, \qquad \forall\, v \in V, \tag{14}$$

$$x_{iv} + x_{i'v} \le 1, \qquad \forall\, v \in V, \quad i, i' = 1, \dots, UB : i < i'. \tag{15}$$

3.2. Formulation using quasi-clique size decompositions

In this section, we present a formulation using the idea of Veremyev et al. [43] to model the quasi-cliques in each element of the partition. The authors showed through computational experiments that such an approach was practical for finding maximum quasi-cliques in sparse graphs. Besides the y and x variables already defined in Section 3.1, the formulation uses the following decision variables. Consider, for every $i \in \{1, \dots, UB\}$ and $k \in \{1, \dots, UB_k\}$, the binary variable

$$z_{ik} = \begin{cases} 1, & \text{if a quasi-clique indexed by } i \text{ has size } k, \\ 0, & \text{otherwise.} \end{cases}$$

Additionally, for every $i \in \{1, \dots, UB\}$ and every $e = uv \in E$, consider the binary variable

$$o_{iuv} = \begin{cases} 1, & \text{if edge } e = uv \text{ is in the quasi-clique indexed by } i, \\ 0, & \text{otherwise.} \end{cases}$$

In this way, MQCPP can be formulated as

$$
\begin{aligned}
(STDs) \quad & \min \sum_{i=1}^{UB} y_i && && (16)\\
& (4)-(5),(10)-(12), && \\
& o_{iuv} \le x_{iu}, && \forall\ i=1,\dots,UB, \quad e=uv\in E, && (17)\\
& o_{iuv} \le x_{iv}, && \forall\ i=1,\dots,UB, \quad e=uv\in E, && (18)\\
& \sum_{v=1}^{n} x_{iv} = \sum_{k=1}^{UB_k} k\cdot z_{ik}, && \forall\ i=1,\dots,UB, && (19)\\
& \sum_{k=1}^{UB_k} z_{ik} = y_i, && \forall\ i=1,\dots,UB, && (20)\\
& \sum_{e=uv\in E} o_{iuv} \ge \gamma\cdot\sum_{k=1}^{UB_k}\frac{k\cdot(k-1)}{2} z_{ik}, && \forall\ i=1,\dots,UB, && (21)\\
& z_{ik}\in\{0,1\}, && \forall\ i=1,\dots,UB, \quad k=1,\dots,UB_k, && (22)\\
& o_{iuv}\in\{0,1\}, && \forall\ i=1,\dots,UB, \quad e=uv\in E. && (23)
\end{aligned}
$$

Constraints (17)-(18) guarantee that an edge is in a given quasi-clique only if both its extremities belong to that partition. Constraints (19) force the number of vertices in each quasi-clique to be equal to the value determined by the z variables. Constraints (20) determine that a given quasi-clique is nonempty only if it is used. Constraints (21) ensure the density of every quasi-clique is greater than or equal to γ. Constraints (22)-(23) ensure the integrality requirements on the variables are met.

3.3. Formulation by representatives

In this section, we provide a formulation by representatives for MQCPP. Such an approach has been successfully applied in the literature to formulate different partition problems [3, 9, 14, 28, 30].

Define, for every two vertices $u, v \in V$ with $u \le v$, the binary variable

$$
X_{uu} = \begin{cases} 1, & \text{if vertex } u \text{ is the representative of a quasi-clique belonging to the partition,} \\ 0, & \text{otherwise;} \end{cases}
$$

$$
X_{uv} = \begin{cases} 1, & \text{if vertex } v \text{ belongs to the quasi-clique represented by vertex } u \text{, with } u < v, \\ 0, & \text{otherwise.} \end{cases}
$$

Furthermore, for any triplet $u, v, v' = 1, \dots, n : u \le v < v'$, consider the binary variable

$$
W_{uvv'} = \begin{cases} 1, & \text{if vertices } v \text{ and } v' \text{ are both in the quasi-clique represented by vertex } u, \\ 0, & \text{otherwise.} \end{cases}
$$

A formulation by representatives for MQCPP can be given by:

$$
\begin{aligned}
(REP) \quad & \min \sum_{u \in V} X_{uu} && && (24)\\
& \sum_{u=1}^{v} X_{uv} = 1, && \forall\, v \in V, && (25)\\
& X_{uv} \leq X_{uu}, && \forall\, u, v \in V : u < v, && (26)\\
& X_{uv} + X_{uv'} \leq W_{uvv'} + 1, && \forall\, u, v, v' \in V : u \leq v < v', && (27)\\
& W_{uvv'} \leq X_{uv}, && \forall\, u, v, v' \in V : u \leq v < v', && (28)\\
& W_{uvv'} \leq X_{uv'}, && \forall\, u, v, v' \in V : u \leq v < v', && (29)\\
& \sum_{\substack{v \in V \\ u \leq v}} \sum_{\substack{v' \in V \\ v < v', vv' \in E}} W_{uvv'} \geq \gamma \cdot \sum_{\substack{v \in V \\ u \leq v}} \sum_{\substack{v' \in V \\ v < v'}} W_{uvv'}, && \forall\, u = 1, \ldots, n-1, && (30)\\
& X_{uv} \in \{0, 1\}, && \forall\, u, v \in V : u \leq v, && (31)\\
& W_{uvv'} \in \{0, 1\}, && \forall\, u, v, v' \in V : u \leq v < v'. && (32)
\end{aligned}
$$

The objective function (24) minimizes the number of quasi-cliques in the partition. Constraints (25) establish that each vertex belongs to exactly one quasi-clique. Constraints (26) guarantee that a vertex may be represented by vertex v only if v is the representative vertex of a quasi-clique that belongs to the partition. Constraints (27)-(29) enforce that if vertices v and v' appear in the quasi-clique represented by vertex u, then the existence or absence of the edge between them will affect the density of this quasi-clique. Constraints (30) enforce that the density of every quasi-clique should be greater than or equal to γ. Lastly, constraints (31)-(32) ensure the integrality requirements of the variables.

Similarly to what was observed earlier in Section 3.1, if one wishes to avoid equality constraints in formulation REP, constraints (25) can be replaced by

$$
\begin{aligned}
& \sum_{u=1}^{v} X_{uv} \geq 1, && \forall\, v \in V, && (33)\\
& X_{uv} + X_{u'v} \leq 1, && \forall\, u, u', v \in V : u < u' \leq v. && (34)
\end{aligned}
$$

3.4. Formulation by representatives using quasi-clique size decomposition

The next formulation combines the ideas of quasi-clique size decompositions and formulations by representatives, already applied in Sections 3.2 and 3.3, respectively.

To formulate the integer program, for every $u \in V$ and $k \in \{1, \ldots, UB_k\}$, consider the binary variable

$$
Z_{uk} = \begin{cases} 1, & \text{if a quasi-clique represented by vertex } u \text{ has size } k, \\ 0, & \text{otherwise.} \end{cases}
$$

Also, for every $u \in V$ and every $e = vv' \in E$, with $u \leq v < v'$, define the binary variable

$$
O_{uvv'} = \begin{cases} 1, & \text{if edge } e = vv' \text{ is in the quasi-clique represented by } u, \\ 0, & \text{otherwise.} \end{cases}
$$

Consequently, MQCPP can be formulated as

$$
\begin{aligned}
(REPs) \quad & \min \sum_{u \in V} X_{uu} && && (35)\\
& (25)-(26),(31)-(32), && &&\\
& O_{uvv'} \leq X_{uv}, && \forall\, u \in V, \quad e = vv' \in E : u \leq v < v', && (36)\\
& O_{uvv'} \leq X_{uv'}, && \forall\, u \in V, \quad e = vv' \in E : u \leq v < v', && (37)\\
& \sum_{v=u}^{n} X_{uv} = \sum_{k=1}^{UB_k} k \cdot Z_{uk}, && \forall\, u \in V, && (38)\\
& \sum_{k=1}^{UB_k} Z_{uk} = X_{uu}, && \forall\, u \in V, && (39)\\
& \sum_{\substack{e=vv' \in E\\ u \leq v < v'}} O_{uvv'} \geq \gamma \cdot \sum_{k=1}^{UB_k} \frac{k \cdot (k-1)}{2} Z_{uk}, && \forall\, u \in V, && (40)\\
& Z_{uk} \in \{0,1\}, && \forall\, u \in V, \quad k = 1, \ldots, UB_k, && (41)\\
& O_{uvv'} \in \{0,1\}, && \forall\, u \in V, \quad e = vv' \in E : u \leq v < v'. && (42)
\end{aligned}
$$

The explanations for constraints (36)-(42) are similar to those for constraints (17)-(23), with the difference that the variables correspond to the representative vertices. Notice that the occurrences of UB_k in formulation REPs can be replaced by $\min\{UB_k, n - v + 1\}$.

4. Multi-start greedy randomized heuristic

This section describes a multi-start greedy randomized heuristic to provide initial feasible solutions (warm starts) to the proposed formulations. It is based on the HCB constructive heuristic [36] for the maximum quasi-clique problem. HCB is a greedy randomized heuristic based on the potential differences introduced in Abello, Pardalos, and Resende [1]. The reader is referenced to the authors' original work [36], in which more details can be obtained.

The proposed heuristic is detailed in Algorithm 1. It takes as inputs the graph G, the value of γ, and a parameter α used to define the size of the restricted candidate lists for the HCB constructive heuristic. Line 1 initializes the best known solution with n quasi-cliques, one for each vertex in V. Each iteration of the outer while loop in lines 2-10 builds a new greedy randomized solution. While a stopping condition is not reached, line 3 initializes the set of vertices that were not yet included in the current partition with all the vertices of G. Line 4 sets the current partial solution S as an empty set. Next, each iteration of the inner while loop in lines 5-8 adds a new maximal quasi-clique to the partial solution S. While there are still vertices that are not in the partial solution (i.e., F is not empty), line 6 calls the HCB heuristic to generate a new maximal quasi-clique Q in $G[F]$. After that, line 7 adds Q to the partial solution S. Line 8 updates the set F by removing the elements in Q. After the inner loop, in lines 9-10, the best solution is updated whenever its cardinality is greater than that of the new constructed quasi-clique partition S. Finally, the best obtained solution S^{best} is returned in line 11.

Algorithm 1: *Multi-start Greedy Randomized* (G, γ, α)

1 $S^{\text{best}} \leftarrow \{\{v_1\}, \ldots, \{v_n\}\}$;
2 **while** *stopping condition is not met* **do**
3     $F \leftarrow V$;
4     $S \leftarrow \emptyset$;
5   **while** $F \neq \emptyset$ **do**
6         $Q \leftarrow HCB(G[F], \gamma, \alpha)$;
7         $S \leftarrow S \cup \{Q\}$;
8         $F \leftarrow F \setminus Q$;
9   **if** $|S| < |S^{\text{best}}|$ **then**
10         $S^{\text{best}} \leftarrow S$;
11 **return** S^{best};

5. Computational experiments

This section summarizes the performed computational experiments. All the tests were executed on a machine running under Ubuntu GNU/Linux, with an Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz processor and 16Gb of RAM. The multi-start greedy randomized heuristic was implemented in C++. The integer programming formulations were tackled using the MIP solver Gurobi 9.0.2.

5.1. Benchmark instances

Each benchmark instance corresponds to an input graph and a value for γ. The considered input graphs are displayed in Table 1, sorted by $|V|$. The graphs correspond to: real networks that are commonly used in the literature, which were recently used for the longest induced path problem and referenced in Matsypura, Veremyev, Prokopyev, and Pasiliao [27]; instances from the DIMACS Implementation Challenges [12]; and instances from the Moviegalaxies data set [20] (https://doi.org/10.7910/DVN/T4HBA3), which represent social connectivity of the characters in movies. The table shows the input graphs, their numbers of vertices and edges, and their densities. For each of the 23 input graphs, instances were considered for every $\gamma \in \{0.999, 0.950, 0.900, 0.800, 0.700, 0.600, 0.500, 0.400, 0.300\}$, leading to a total of 207 instances. We remark that we selected these graphs because they are widely used in the literature and also considering the limitations of the integer programming approaches, which have $O(|V|^3)$ variables each.

Table 1: Input graphs used as benchmarks.

Input graph	$\|V\|$	$\|E\|$	$d(G)$
Memento	14	19	0.2088
The_X_Files	24	41	0.1486
Alien_3	25	77	0.2567
high-tech	33	91	0.1723
karate	34	78	0.1390
mexican	35	117	0.1966
sawmill	36	62	0.0984
tailorS1	39	158	0.2132
chesapeake	39	170	0.2294
Batman_Returns	51	124	0.0973
attiro	59	128	0.0748
krebs	62	153	0.0809
dolphins	62	159	0.0841
prison	67	142	0.0642
sanjuansur	75	144	0.0519
jean	77	254	0.0868
3-FullIns_3	80	346	0.1095
david	87	406	0.1085
myciel6	95	755	0.1691
4-FullIns_3	114	541	0.0840
ieeebus	118	179	0.0259
sfi	118	200	0.0290
anna	138	493	0.0522

5.2. Tested approaches and settings

We compare the standard formulation (STD), the formulation using quasi-clique size decompositions (STDs), the formulation by representatives (REP), and the formulation by representatives using quasi-clique size decomposition (REPs), all of them presented in Section 3. The solutions obtained by the multi-start greedy randomized heuristic (MSH), described in Section 4, were provided as initial feasible solutions for each execution of all the formulations.

We defined a time limit of 30 seconds as the stopping criterion for obtaining an initial feasible solution with the heuristic MSH. Similarly to Pinto et al. [36], we set $\alpha = 0.09$. For solving the formulations, Gurobi was executed with the standard configurations using a single thread, and a time limit of 3600 seconds was imposed on each run.

5.3. Computational results

This section summarizes the computational results. Individualized results for each instance are available in Appendix A. Table 2 summarizes the results obtained by the formulations assembled by the input graphs. The first column identifies the input graph, followed by the number of instances (i.e., the number of values for γ). All reported averages in other columns of the table are taken over the nine instances with different values of γ for the specified input graph. The third column gives the average best solution value obtained by MSH within 30 seconds. Next, for each formulation, the table provides the average best solution value, the number of instances exactly solved, the average time in seconds for solving those that were optimally solved, and the average gap in percentage for the instances that were not solved to optimality within 3600 seconds. In each line, the smallest average best solution value (column best) and the largest number of instances solved to optimality (column #opt) are shown in boldface. The results show that formulation REPv outperforms the other formulations when it comes to the average best solution values. At the same time, while REP is slightly better than REPv regarding the number of instances solved to optimality. Besides, it can be seen that the heuristic MSH achieved, in low computational times, solutions whose best values were, in several cases, not far from the smallest ones reported in the table. Additionally, STD was the formulation that faced the most difficulties in improving the solutions obtained by MSH. It can also be observed from this table that REPv obtains lower average open gaps for most of the input graphs.

Table 3 summarizes the results obtained by the formulations, categorized by the values of γ. The first column shows each considered value of γ, followed by the number of instances (i.e., the number of input graphs). The remaining columns are the same as those in Table 2, except that all reported averages are taken over the 23 instances corresponding to the different input graphs for each specific value of γ. It is noticeable from this table that the two formulations by representatives (REP and REPv) dominate the others when it comes to the average best solution values, with REP outperforming all the others for larger values of γ and REPv showing the best performance for medium and lower values of γ. Regarding the number of instances solved to optimality, REP performed the best for those with $\gamma \geq 0.800$. For $\gamma \leq 0.700$, STDv and REPv achieved more optimal solutions for three values of γ each. It should be noticed that STDv achieved the largest number of instances solved to optimality for the three smallest values of γ.

Figure 4 shows the fraction in percent of the instances solved to optimality by at least one of the formulations. The plot indicates that considering the input graphs used in the benchmark, the instances with large values of γ (0.999, 0.950, and 0.900) tend to be easier to solve. In contrast, those with intermediary values (0.700 and 0.600) appear to be the most difficult. Furthermore, the plot shows that the proposed approaches could solve nearly 59% of the instances to optimality.

Table 2: Summary of the results obtained by the four formulations displayed by the input graphs.

		MSH	STD				STDv				REP				REPv			
Input graph	#inst	best	best	#opt	time (s)	gap (%)	best	#opt	time (s)	gap (%)	best	#opt	time (s)	gap (%)	best	#opt	time (s)	gap (%)
Memento	9	8.4	**7.8**	**9**	10.6		**7.8**	**9**	1.1		**7.8**	**9**	0.5		**7.8**	**9**	0.1	
The_X_Files	9	10.0	**9.8**	3	11.4	35.1	**9.8**	3	0.7	19.8	**9.8**	**9**	53.5		**9.8**	**9**	3.9	
Alien_3	9	7.0	**6.6**	5	151.4	29.3	**6.6**	5	81.2	28.7	**6.6**	**9**	173.5		**6.6**	**9**	32.4	
high-tech	9	10.6	**10.4**	2	76.2	51.4	**10.4**	3	62.9	45.3	**10.4**	6	447.6	27.8	**10.4**	**9**	572.4	
karate	9	13.8	13.1	2	6.8	58.5	13.0	3	24.8	42.3	**12.9**	7	434.2	41.4	**12.9**	**9**	32.5	
mexican	9	8.1	8.1	3	12.3	47.1	**8.0**	4	179.5	39.8	8.1	**6**	112.9	36.0	8.1	**6**	1320.6	20.4
sawmill	9	12.8	**12.8**	1	1.3	57.5	**12.8**	3	33.9	36.7	**12.8**	6	54.7	23.6	**12.8**	**9**	45.9	
tailorS1	9	10.3	10.2	3	283.0	62.8	10.1	2	0.5	49.9	**9.8**	**6**	479.7	49.0	9.9	2	20.8	47.5
chesapeake	9	10.8	10.6	3	749.5	61.5	10.4	3	25.1	53.0	**10.3**	**4**	99.5	45.5	**10.3**	3	871.4	34.6
Batman_Returns	9	14.3	14.3	1	11.5	69.4	13.6	2	33.2	39.2	13.7	4	260.8	52.8	**13.3**	**7**	1477.5	28.4
attiro	9	18.8	18.8	1	14.6	76.6	**18.6**	1	805.3	41.1	18.8	4	367.4	51.4	**18.6**	**5**	729.2	20.7
krebs	9	23.0	22.8	1	227.5	80.1	21.7	1	392.8	51.3	22.0	**3**	129.3	55.2	**21.3**	1	1625.7	47.9
dolphins	9	20.3	20.3	1	7.5	79.4	19.3	1	1277.3	50.8	19.8	**3**	32.3	53.6	**19.2**	0		34.4
prison	9	18.9	18.9	1	6.0	77.3	18.7	0		34.7	18.8	**4**	407.9	55.8	**18.6**	3	1916.2	23.2
sanjuansur	9	24.7	24.7	1	32.2	82.7	24.6	0		39.9	24.3	4	435.3	55.9	**24.2**	**5**	1244.1	21.6
jean	9	26.8	26.3	**1**	17.5	85.0	24.1	**1**	213.1	57.6	25.9	**1**	1.7	63.9	**23.0**	**1**	323.3	53.3
3-FullIns_3	9	26.0	25.9	1	38.3	81.4	25.9	0		65.5	**25.4**	**2**	1277.4	64.3	26.0	0		69.8
david	9	27.7	27.2	1	188.9	87.0	24.6	0		59.7	26.6	**2**	1501.1	76.3	**22.8**	0		58.8
myciel6	9	39.6	38.9	0		81.3	38.2	0		83.5	36.9	**1**	2.5	89.3	**35.3**	0		82.0
4-FullIns_3	9	38.7	38.6	0		79.8	38.6	0		75.2	**38.0**	**1**	7.1	74.0	38.4	0		74.2
ieeebus	9	45.9	45.4	0		82.9	45.4	0		43.8	45.2	3	218.3	64.1	**44.3**	**5**	2377.4	36.2
sfi	9	53.4	53.4	0		83.6	50.6	0		54.3	51.7	1	5.1	58.8	**47.8**	**3**	1652.4	37.9
anna	9	67.7	67.7	**1**	264.5	95.3	65.8	0		80.7	65.7	**1**	9.8	76.4	**62.3**	0		77.9
Average		23.4	23.2		111.1	70.2	22.5		223.7	49.7	22.7		283.1	55.7	**21.9**		838.0	45.2
Total	207			41				41				**96**				95		

Table 3: Summary of the results obtained by the four formulations displayed by the values of γ.

		MSH	STD				STDv				REP				REPv			
γ	#inst	best	best	#opt	time (s)	gap (%)	best	#opt	time (s)	gap (%)	best	#opt	time (s)	gap (%)	best	#opt	time (s)	gap (%)
0.999	23	31.5	**31.2**	19	42.8	6.7	31.5	1	1.1	53.6	**31.2**	**23**	1.7		31.5	14	883.7	59.7
0.950	23	31.4	31.4	1	0.1	77.8	31.4	1	1.7	55.5	**31.1**	**18**	335.1	27.6	31.3	13	648.7	61.7
0.900	23	30.8	30.8	1	13.2	79.8	30.8	1	1.3	55.8	**30.3**	**16**	101.7	33.9	30.8	13	1023.4	63.1
0.800	23	28.9	28.7	1	14.1	77.9	28.9	1	1.2	56.0	**28.2**	**12**	579.0	48.3	28.6	10	831.1	55.8
0.700	23	27.4	27.1	1	19.9	75.6	27.1	2	194.8	58.0	26.8	6	399.3	56.5	**26.7**	**8**	953.1	55.6
0.600	23	19.7	19.5	2	380.3	72.4	19.0	3	228.6	44.7	18.9	5	742.9	67.2	**18.0**	**7**	598.2	48.0
0.500	23	16.7	16.5	2	16.7	69.0	15.4	**8**	60.3	48.5	15.7	3	113.9	68.2	**13.7**	7	489.5	42.2
0.400	23	13.7	13.4	6	528.8	69.9	11.3	**10**	7.7	41.1	12.8	5	505.7	68.6	**10.1**	**10**	38.3	42.0
0.300	23	10.1	9.7	8	21.1	68.1	7.3	**14**	192.3	36.4	8.9	8	129.5	69.4	**6.5**	13	267.2	39.2
Average		23.4	23.2		111.1	70.2	22.5		223.7	49.7	22.7		283.1	55.7	**21.9**		838.0	45.2
Total	207			41				41				**96**				95		

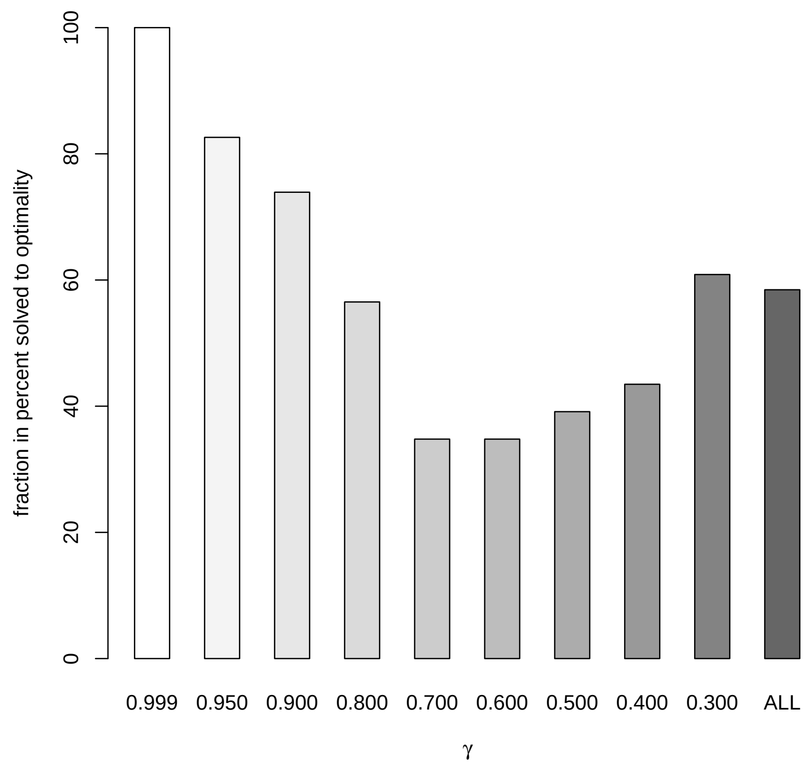


Figure 4: Fraction in percent of the instances solved to optimality by at least one formulation within the time of 3600 seconds.

6. Concluding remarks

In this paper, we considered the minimum quasi-clique partitioning problem (MQCPP), which has applicability in clustering and network analysis. We formally introduced the problem and showed that its decision version is NP-complete even when one asks for a partition into two γ-cliques, i.e., a bipartition. This interesting result contrasts with that for the minimum clique partitioning problem, whose decision version is NP-complete for obtaining a partition into more than two cliques but polynomially solvable for two cliques.

Furthermore, we proposed four integer programming formulations and a multi-start greedy randomized heuristic for MQCPP. The computational experiments showed that the formulation by representatives (REP) and the formulation by representatives using the idea of clique-size decomposition (REPs) are those with the best performances. While REP solved more instances to optimality, REPs outperformed the others regarding the best-obtained solution values. Besides, the results also indicate that instances with intermediate values of γ tend to be more challenging to be solved by the proposed formulations. Furthermore, the experiments showed that the proposed heuristic could obtain, in low computational times, feasible solutions of reasonable quality to be offered to the formulations. Finally, the results indicate that the proposed methods seem promising for partitioning real networks into dense subgraphs and, consequently, to be used as clustering approaches in practice.

We believe that our work opens promising prospects for future research. One of them is the study of other metaheuristics for MQCPP. Besides, one could analyze how to extend our approaches for partitioning graphs into other families of dense subgraphs, such as the more general (λ, γ)-cliques. In addition, this work also provides an alternative strategy for new exact community detection methods in social and communication networks.

Acknowledgments

The work of Rafael A. Melo was supported by the State of Bahia Research Foundation (FAPESB) and the Brazilian National Council for Scientific and Technological Development (CNPq). Part of this work was developed while the author was a visiting scholar at the Department of Computer Science of Universidade Federal de Minas Gerais, Brazil. The work of Celso C. Ribeiro was partially supported by CNPq research grants 309869/2020-0 and 425778/2016-9 and by FAPERJ research grant E-26/200.926/2021. A CNPq scholarship sponsored the work of Jose A. Riveaux. This work was also partially sponsored by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), under Finance Code 001.

References

[1] J. Abello, P. M. Pardalos, and M. G. C. Resende. On maximum clique problems in very large graphs. In J. M. Abello and J. S. Vitter, editors, *External Memory Algorithms*, pages 119–130. American Mathematical Society, 1999.

[2] J. Abello, M. Resende, and S. Sudarsky. Massive quasi-clique detection. In J. Abello and J. Vitter, editors, *Proceedings of the 5th Latin American Symposium on the Theory of Informatics*, volume 2286 of *Lecture Notes in Computer Science*, pages 598–612. Springer, Berlin, 2002.

[3] L. Bahiense, Y. A. M. Frota, T. F. Noronha, and C. C. Ribeiro. A branch-and-cut algorithm for the equitable coloring problem using a formulation by representatives. *Discrete Applied Mathematics*, 164:34–46, 2014.

[4] S. Basu, D. Sengupta, U. Maulik, and S. Bandyopadhyay. A strong nash stability based approach to minimum quasi clique partitioning. In *2014 Sixth International Conference on Communication Systems and Networks*, pages 1–6, Bangalore, 2014. IEEE.

[5] C. Blum, M. Djukanovic, A. Santini, H. Jiang, C.-M. Li, F. Manyà, and G. R. Raidl. Solving longest common subsequence problems via a transformation to the maximum clique problem. *Computers & Operations Research*, 125:105089, 2021.

[6] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo. The maximum clique problem. In P. M. Pardalos, D.-Z. Du, and R. L. Graham, editors, *Handbook of Combinatorial Optimization*, pages 1–74. Springer, 1999.

[7] J. Brimberg, S. Janićijević, N. Mladenović, and D. Urošević. Solving the clique partitioning problem as a maximally diverse grouping problem. *Optimization Letters*, 11: 1123–1135, 2017.

[8] M. Campêlo, R. Corrêa, and Y. Frota. Cliques, holes and the vertex coloring polytope. *Information Processing Letters*, 89:159–164, 2004.

[9] M. Campêlo, V. Campos, and R. Corrêa. On the asymmetric representatives formulation for the vertex coloring problem. *Electronic Notes in Discrete Mathematics*, 19:337–343, 2005.

[10] S. G. De Amorim, J.-P. Barthélemy, and C. C. Ribeiro. Clustering and clique partitioning: Simulated annealing and tabu search approaches. *Journal of Classification*, 9: 17–41, 1992.

[11] M. Dell'Amico, R. Montemanni, and S. Novellani. Exact models for the flying sidekick traveling salesman problem. *International Transactions in Operational Research*, 29: 1360–1393, 2022.

[12] DIMACS. Implementation challenges, 2021. Online reference at http://dimacs.rutgers.edu/Challenges/ last visited on November 27, 2021.

[13] M. El-Moussaoui, T. Agouti, A. Tikniouine, and M. El -Adnani. A comprehensive literature review on community detection: Approaches and applications. *Procedia Computer Science*, 151:295–302, 2019.

[14] Y. Frota, N. Maculan, T. F. Noronha, and C. C. Ribeiro. A branch-and-cut algorithm for partition coloring. *Networks: An International Journal*, 55:194–204, 2010.

[15] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete problems. In *Proceedings of the Sixth Annual ACM Symposium on Theory of Computing*, pages 47–63, Seattle, 1974. ACM.

[16] F. Glaria, C. Hernández, S. Ladra, G. Navarro, and L. Salinas. Compact structure for sparse undirected graphs based on a clique graph partition. *Information Sciences*, 544: 485–499, 2021.

[17] J. Gramm, J. Guo, F. Hüffner, and R. Niedermeier. Data reduction and exact algorithms for clique cover. *ACM Journal of Experimental Algorithmics*, 13:2.2–2.15, 2009.

[18] H. Hu, X. Yan, Y. Huang, J. Han, and X. J. Zhou. Mining coherent dense subgraphs across massive biological networks for functional discovery. *Bioinformatics*, 21:i213–i221, 2005.

[19] T. R. Jensen and B. Toft. *Graph Coloring Problems*, volume 39. Wiley, 2011.

[20] J. Kaminski, M. Schober, R. Albaladejo, O. Zastupailo, and C. Hidalgo. Moviegalaxies - Social networks in movies, 2018. Online reference at https://doi.org/10.7910/DVN/T4HBA3 last visited on November 27, 2021.

[21] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85—103. Plenum Press, 1972.

[22] V. E. Lee, N. Ruan, R. Jin, and C. Aggarwal. A survey of algorithms for dense subgraph discovery. In C. Aggarwal and H. Wang, editors, *Managing and Mining Graph Data*, volume 40 of *Advances in Database Systems*, pages 303–336. Springer, 2010.

[23] C.-M. Li, H. Jiang, and F. Manyà. On minimization of the number of branches in branch-and-bound algorithms for the maximum clique problem. *Computers & Operations Research*, 84:1–15, 2017.

[24] E. Malaguti and P. Toth. A survey on vertex coloring problems. *International Transactions in Operational Research*, 17:1–34, 2010.

[25] F. Marinelli, A. Pizzuti, and F. Rossi. LP-based dual bounds for the maximum quasi-clique problem. *Discrete Applied Mathematics*, 296:118–140, 2021.

[26] R. G. Marzo, R. A. Melo, C. C. Ribeiro, and M. C. Santos. New formulations and branch-and-cut procedures for the longest induced path problem. *Computers & Operations Research*, 139:105627, 2022.

[27] D. Matsypura, A. Veremyev, O. A. Prokopyev, and E. L. Pasiliao. On exact solution approaches for the longest induced path problem. *European Journal of Operational Research*, 278:546–562, 2019.

[28] R. A. Melo and C. C. Ribeiro. Improved solutions for the freight consolidation and containerization problem using aggregation and symmetry breaking. *Computers & Industrial Engineering*, 85:402–413, 2015.

[29] R. A. Melo, M. F. Queiroz, and C. C. Ribeiro. Compact formulations and an iterated local search-based matheuristic for the minimum weighted feedback vertex set problem. *European Journal of Operational Research*, 289:75–92, 2021.

[30] R. A. Melo, M. F. Queiroz, and M. C. Santos. A matheuristic approach for the b-coloring problem using integer programming and a multi-start multi-greedy randomized metaheuristic. *European Journal of Operational Research*, 295:66–81, 2021.

[31] I. Méndez-Díaz and P. Zabala. A branch-and-cut algorithm for graph coloring. *Discrete Applied Mathematics*, 154:826–847, 2006.

[32] M. Oosten, J. H. Rutten, and F. C. Spieksma. The clique partitioning problem: facets and patching facets. *Networks: An International Journal*, 38:209–226, 2001.

[33] J. Pattillo, N. Youssef, and S. Butenko. On clique relaxation models in network analysis. *European Journal of Operational Research*, 226:9–18, 2013.

[34] J. Pattillo, A. Veremyev, S. Butenko, and V. Boginski. On the maximum quasi-clique problem. *Discrete Applied Mathematics*, 161:244–257, 2013.

[35] B. Peng, L. Wu, Y. Wang, and Q. Wu. Solving maximum quasi-clique problem by a hybrid artificial bee colony approach. *Information Sciences*, 578:214–235, 2021.

[36] B. Q. Pinto, C. C. Ribeiro, I. Rosseti, and A. Plastino. A biased random-key genetic algorithm for the maximum quasi-clique problem. *European Journal of Operational Research*, 271:849–865, 2018.

[37] B. Q. Pinto, C. C. Ribeiro, J. A. Riveaux, and I. Rosseti. A BRKGA-based matheuristic for the maximum quasi-clique problem with an exact local search strategy. *RAIRO: Recherche Opérationnelle*, 55:S741 – S763, 2021.

[38] C. C. Ribeiro and J. Riveaux. An exact algorithm for the maximum quasi-clique problem. *International Transactions in Operational Research*, 26:2199–2229, 2019.

[39] S.-V. Sanei-Mehri, A. Das, H. Hashemi, and S. Tirthapura. Mining largest maximal quasi-cliques. *ACM Transactions on Knowledge Discovery from Data*, 15:1–21, 2021.

[40] J. H. Seo and M. H. Kim. Finding influential communities in networks with multiple influence types. *Information Sciences*, 548:254–274, 2021.

[41] V. Spirin and L. A. Mirny. Protein complexes and functional modules in molecular networks. *Proceedings of the National Academy of Sciences*, 100:12123–12128, 2003.

[42] C. Tsourakakis, F. Bonchi, A. Gionis, F. Gullo, and M. Tsiarli. Denser than the densest subgraph: extracting optimal quasi-cliques with quality guarantees. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 104–112, Chicago, 2013.

[43] A. Veremyev, O. A. Prokopyev, S. Butenko, and E. L. Pasiliao. Exact MIP-based approaches for finding maximum quasi-clique and dense subgraphs. *Computational Optimization and Applications*, 64:177–214, 2016.

[44] A. Verma and S. Butenko. Network clustering via clique relaxations: A community based approach. In D. A. Bader, H. Meyerhenke, P. Sanders, and D. Wagner, editors, *Graph Partitioning and Graph Clustering*, volume 588 of *Contemporary Mathematics*, pages 129–139. American Mathematical Society, 2013.

[45] V. Verteletskyi, T.-C. Yen, and A. F. Izmaylov. Measurement optimization in the variational quantum eigensolver using a minimum clique cover. *The Journal of Chemical Physics*, 152:124114, 2020.

[46] Q. Wu and J.-K. Hao. A review on algorithms for maximum clique problems. *European Journal of Operational Research*, 242:693–709, 2015.

[47] Z. Yang, R. Algesheimer, and C. J. Tessone. A comparative analysis of community detection algorithms on artificial networks. *Scientific Reports*, 6:30750, 2016.

[48] X. Zhao, J. Liang, and J. Wang. A community detection algorithm based on graph compression for large-scale social networks. *Information Sciences*, 551:358–372, 2021.

[49] Q. Zhou, U. Benlic, and Q. Wu. An opposition-based memetic algorithm for the maximum quasi-clique problem. *European Journal of Operational Research*, 286:63–83, 2020.

Appendix A Detailed results

Table 4 presents the results obtained by each of the formulations for each instance. The first two columns indicate the input graph and the value of γ. The third column gives the value of the best solution obtained by the heuristic MSH within 30 seconds. The next columns show, for each of the formulations (STD, STDv, REP, and REPv), the value of the best obtained solution, the best bound achieved by the solver, and the running time in seconds.

Table 4: Detailed results obtained by the four formulations for the minimum quasi-clique partitioning problem.

		MSH	STD			STDv			REP			REPv		
Input graph	γ	best	best	bbound	time (s)	best	bbound	time (s)	best	bbound	time (s)	best	bbound	time (s)
Memento	0.999	**10**	**10**	10	0.024	**10**	10	1.123	**10**	10	0.004	**10**	10	0.103
Memento	0.950	**10**	**10**	10	0.050	**10**	10	1.747	**10**	10	0.004	**10**	10	0.097
Memento	0.900	**10**	**10**	10	13.215	**10**	10	1.260	**10**	10	0.009	**10**	10	0.130
Memento	0.800	**9**	**9**	9	14.072	**9**	9	1.163	**9**	9	0.049	**9**	9	0.108
Memento	0.700	**9**	**9**	9	19.878	**9**	9	1.467	**9**	9	0.112	**9**	9	0.137
Memento	0.600	**8**	**8**	8	24.206	**8**	8	1.984	**8**	8	0.332	**8**	8	0.118
Memento	0.500	8	**6**	6	15.974	**6**	6	0.650	**6**	6	0.468	**6**	6	0.141
Memento	0.400	7	**5**	5	7.290	**5**	5	0.641	**5**	5	1.144	**5**	5	0.261
Memento	0.300	5	**3**	3	0.710	**3**	3	0.108	**3**	3	2.197	**3**	3	0.180
The_X_Files	0.999	**14**	**14**	14	0.095	**14**	12	3600.004	**14**	14	0.016	**14**	14	1.454
The_X_Files	0.950	**14**	**14**	9	3600.006	**14**	11	3600.004	**14**	14	0.040	**14**	14	2.320
The_X_Files	0.900	**14**	**14**	7	3600.005	**14**	11	3600.004	**14**	14	0.147	**14**	14	2.918
The_X_Files	0.800	**13**	**13**	8	3600.004	**13**	10	3600.005	**13**	13	0.802	**13**	13	3.622
The_X_Files	0.700	12	**11**	7	3600.005	**11**	8	3600.004	**11**	11	2.202	**11**	11	7.873
The_X_Files	0.600	**9**	**9**	6	3600.005	**9**	8	3600.004	**9**	9	49.561	**9**	9	9.815
The_X_Files	0.500	**6**	**6**	5	3600.004	**6**	6	1.897	**6**	6	110.959	**6**	6	4.863
The_X_Files	0.400	5	**4**	4	31.956	**4**	4	0.274	**4**	4	300.454	**4**	4	2.397
The_X_Files	0.300	**3**	**3**	3	2.106	**3**	3	0.050	**3**	3	17.577	**3**	3	0.065
Alien_3	0.999	**11**	**11**	11	0.121	**11**	7	3600.004	**11**	11	0.033	**11**	11	35.667
Alien_3	0.950	**11**	**11**	8	3600.005	**11**	7	3600.004	**11**	11	0.164	**11**	11	64.696
Alien_3	0.900	**10**	**10**	6	3600.005	**10**	8	3600.004	**10**	10	1.172	**10**	10	46.149
Alien_3	0.800	**9**	**9**	6	3600.006	**9**	7	3600.005	**9**	9	68.530	**9**	9	72.105
Alien_3	0.700	8	**6**	5	3600.005	**6**	6	388.034	**6**	6	106.320	**6**	6	36.423
Alien_3	0.600	**5**	**5**	5	736.402	**5**	5	17.340	**5**	5	1069.271	**5**	5	31.611
Alien_3	0.500	4	**3**	3	17.415	**3**	3	0.259	**3**	3	230.227	**3**	3	4.113
Alien_3	0.400	3	**2**	2	3.255	**2**	2	0.192	**2**	2	85.667	**2**	2	0.402
Alien_3	0.300	**2**	**2**	2	0.052	**2**	2	0.020	**2**	2	0.107	**2**	2	0.035
high-tech	0.999	**16**	**16**	16	0.256	**16**	9	3600.009	**16**	16	0.072	**16**	16	103.647
high-tech	0.950	**16**	**16**	6	3600.109	**16**	8	3600.010	**16**	16	0.810	**16**	16	180.802
high-tech	0.900	**15**	**15**	5	3600.003	**15**	7	3600.010	**15**	15	3.426	**15**	15	326.564
high-tech	0.800	**14**	**14**	5	3600.002	**14**	7	3600.010	**14**	14	111.734	**14**	14	558.308
high-tech	0.700	**12**	**12**	5	3600.002	**12**	6	3600.010	**12**	12	2120.085	**12**	12	1163.487
high-tech	0.600	9	**8**	4	3600.191	**8**	6	3600.011	**8**	6	3600.008	**8**	8	827.532
high-tech	0.500	**6**	**6**	4	3600.011	**6**	6	187.855	**6**	4	3600.031	**6**	6	1981.515
high-tech	0.400	**4**	**4**	3	3600.011	**4**	4	0.711	**4**	3	3600.008	**4**	4	7.156
high-tech	0.300	**3**	**3**	3	152.091	**3**	3	0.242	**3**	3	449.544	**3**	3	2.866

Continued on next page

Table 4 – continued from previous page

		MSH	STD			STDv			REP			REPv		
Input graph	γ	best	best	bbound	time (s)	best	bbound	time (s)	best	bbound	time (s)	best	bbound	time (s)
karate	0.999	**20**	**20**	20	0.214	**20**	11	3600.010	**20**	20	0.061	**20**	20	18.524
karate	0.950	**20**	**20**	6	3600.012	**20**	11	3600.010	**20**	20	0.409	**20**	20	25.864
karate	0.900	**19**	**19**	6	3600.005	**19**	10	3600.011	**19**	19	4.338	**19**	19	23.378
karate	0.800	18	**17**	5	3600.008	18	9	3600.011	**17**	17	26.329	**17**	17	52.001
karate	0.700	17	**15**	5	3600.007	**15**	8	3600.011	**15**	15	138.245	**15**	15	69.826
karate	0.600	13	11	4	3600.002	**10**	8	3600.011	**10**	10	2304.545	**10**	10	55.491
karate	0.500	9	8	4	3600.002	**7**	7	67.598	**7**	4	3600.088	**7**	7	40.334
karate	0.400	**5**	**5**	4	3600.012	**5**	5	6.612	**5**	3	3600.236	**5**	5	6.603
karate	0.300	**3**	**3**	3	13.399	**3**	3	0.067	**3**	3	565.466	**3**	3	0.236
mexican	0.999	**13**	**13**	13	0.336	**13**	7	3600.012	**13**	13	0.092	**13**	13	1135.882
mexican	0.950	**13**	**13**	6	3600.003	**13**	7	3600.012	**13**	13	2.147	**13**	13	1173.879
mexican	0.900	**12**	**12**	5	3600.004	**12**	7	3600.012	**12**	12	11.767	**12**	12	2294.478
mexican	0.800	**10**	**10**	5	3600.007	**10**	6	3600.012	**10**	10	383.898	**10**	10	3319.170
mexican	0.700	**8**	**8**	5	3600.018	**8**	6	3600.013	**8**	6	3600.015	**8**	7	3600.007
mexican	0.600	7	7	4	3600.015	**6**	6	666.593	7	4	3600.002	7	5	3600.007
mexican	0.500	**5**	**5**	3	3600.014	**5**	5	51.217	**5**	3	3600.028	**5**	4	3600.008
mexican	0.400	**3**	**3**	3	36.522	**3**	3	0.134	**3**	3	279.281	**3**	3	0.332
mexican	0.300	**2**	**2**	2	0.102	**2**	2	0.048	**2**	2	0.151	**2**	2	0.076
sawmill	0.999	**18**	**18**	18	1.308	**18**	12	3600.013	**18**	18	0.077	**18**	18	14.403
sawmill	0.950	**18**	**18**	5	3600.028	**18**	12	3600.014	**18**	18	0.239	**18**	18	15.805
sawmill	0.900	**18**	**18**	5	3600.020	**18**	11	3600.013	**18**	18	1.492	**18**	18	30.594
sawmill	0.800	**16**	**16**	5	3600.003	**16**	10	3600.013	**16**	16	6.127	**16**	16	26.443
sawmill	0.700	**16**	**16**	4	3600.018	**16**	8	3600.013	**16**	16	28.970	**16**	16	43.042
sawmill	0.600	**11**	**11**	4	3600.002	**11**	8	3600.014	**11**	11	290.999	**11**	11	146.263
sawmill	0.500	**8**	**8**	4	3600.003	**8**	8	98.927	**8**	7	3600.005	**8**	8	96.526
sawmill	0.400	**6**	**6**	4	3600.015	**6**	6	2.348	**6**	4	3600.002	**6**	6	39.373
sawmill	0.300	**4**	**4**	3	3600.003	**4**	4	0.339	**4**	3	3600.002	**4**	4	0.286
tailorS1	0.999	**17**	**17**	17	2.667	**17**	7	3600.017	**17**	17	0.170	**17**	8	3600.010
tailorS1	0.950	**17**	**17**	5	3600.117	**17**	6	3600.017	**17**	17	6.537	**17**	7	3600.010
tailorS1	0.900	**15**	**15**	4	3600.020	**15**	6	3600.017	**15**	15	38.344	**15**	7	3600.010
tailorS1	0.800	13	13	4	3600.003	13	6	3600.017	**12**	12	971.212	**12**	6	3600.010
tailorS1	0.700	11	11	4	3600.076	11	5	3600.017	**10**	5	3600.003	11	5	3600.011
tailorS1	0.600	8	8	4	3600.004	8	5	3600.017	**7**	3	3600.002	**7**	4	3600.010
tailorS1	0.500	6	6	3	3600.005	**5**	4	3600.016	**5**	3	3600.002	**5**	4	3600.010
tailorS1	0.400	4	**3**	3	846.108	**3**	3	0.923	**3**	3	1861.806	**3**	3	41.549
tailorS1	0.300	**2**	**2**	2	0.140	**2**	2	0.066	**2**	2	0.204	**2**	2	0.108
chesapeake	0.999	**17**	**17**	17	0.516	**17**	7	3600.017	**17**	17	0.142	**17**	17	2603.069
chesapeake	0.950	**17**	**17**	5	3600.003	**17**	6	3600.017	**17**	17	77.271	**17**	12	3600.011
chesapeake	0.900	17	17	5	3600.003	17	6	3600.016	**16**	16	320.062	17	10	3600.010
chesapeake	0.800	15	14	4	3600.004	14	6	3600.016	**13**	8	3600.002	**13**	8	3600.010
chesapeake	0.700	**12**	**12**	4	3600.003	**12**	5	3600.017	**12**	4	3600.002	**12**	6	3600.010
chesapeake	0.600	8	8	4	3600.071	**7**	6	3600.018	**7**	3	3600.009	**7**	5	3600.011
chesapeake	0.500	**5**	**5**	3	3600.003	**5**	5	74.254	**5**	3	3600.003	**5**	4	3600.010
chesapeake	0.400	4	**3**	3	2247.916	**3**	3	0.874	4	3	3600.002	**3**	3	11.122
chesapeake	0.300	**2**	**2**	2	0.142	**2**	2	0.069	**2**	2	0.362	**2**	2	0.100

Continued on next page

Table 4 – continued from previous page

Input graph	γ	MSH best	STD best	STD bbound	STD time (s)	STDv best	STDv bbound	STDv time (s)	REP best	REP bbound	REP time (s)	REPv best	REPv bbound	REPv time (s)
Batman_Returns	0.999	**20**	**20**	20	11.467	**20**	12	3600.005	**20**	20	0.251	**20**	20	1177.708
Batman_Returns	0.950	**20**	**20**	5	3600.074	**20**	12	3600.039	**20**	20	8.742	**20**	20	1479.540
Batman_Returns	0.900	**20**	**20**	4	3600.011	**20**	10	3600.038	**20**	20	49.210	**20**	20	2992.159
Batman_Returns	0.800	**17**	**17**	4	3600.008	**17**	10	3600.038	**17**	17	985.040	**17**	13	3600.024
Batman_Returns	0.700	16	16	4	3600.007	16	8	3600.038	**15**	9	3600.017	**15**	10	3600.023
Batman_Returns	0.600	11	11	4	3600.007	**10**	8	3600.039	**10**	4	3600.005	**10**	10	3116.494
Batman_Returns	0.500	11	11	3	3600.007	9	6	3600.040	9	3	3600.038	**8**	8	1299.324
Batman_Returns	0.400	8	8	3	3600.008	**6**	6	63.842	7	3	3600.005	**6**	6	273.627
Batman_Returns	0.300	6	6	3	3600.013	**4**	4	2.650	5	3	3600.005	**4**	4	3.402
attiro	0.999	**27**	**27**	27	14.635	**27**	15	3600.008	**27**	27	0.423	**27**	27	608.158
attiro	0.950	**27**	**27**	4	3613.708	**27**	13	3600.008	**27**	27	16.097	**27**	27	452.162
attiro	0.900	**27**	**27**	4	3600.012	**27**	14	3600.008	**27**	27	41.534	**27**	27	375.218
attiro	0.800	**24**	**24**	4	3614.460	**24**	11	3600.060	**24**	24	1411.460	**24**	24	693.967
attiro	0.700	**21**	**21**	3	3602.443	**21**	11	3600.060	**21**	17	3600.007	**21**	18	3600.039
attiro	0.600	**14**	**14**	4	3600.011	**14**	9	3600.062	**14**	9	3600.024	**14**	11	3600.036
attiro	0.500	**12**	**12**	3	3600.011	**12**	9	3600.062	**12**	3	3600.056	**12**	9	3600.038
attiro	0.400	10	10	3	3600.011	**9**	7	3600.062	10	3	3600.010	**9**	7	3600.038
attiro	0.300	7	7	3	3600.012	**6**	6	805.260	7	3	3600.007	**6**	6	1516.723
krebs	0.999	**33**	**33**	33	227.467	**33**	12	3600.010	**33**	33	0.672	**33**	21	3600.006
krebs	0.950	**33**	33	5	3600.015	**33**	13	3600.009	**33**	33	54.927	**33**	15	3600.008
krebs	0.900	**31**	**31**	4	3600.847	**31**	12	3600.011	**31**	31	332.220	**31**	14	3600.006
krebs	0.800	27	27	4	3600.016	27	11	3600.009	**26**	24	3600.088	27	12	3601.110
krebs	0.700	**25**	**25**	4	3600.031	**25**	10	3600.010	**25**	13	3600.008	**25**	10	3600.557
krebs	0.600	20	19	4	3604.280	20	9	3600.067	**18**	5	3600.010	**18**	8	3600.006
krebs	0.500	17	16	3	3600.207	13	8	3600.072	14	3	3600.009	**12**	7	3600.006
krebs	0.400	13	13	3	3600.013	**8**	7	3600.064	12	3	3600.013	**8**	6	3600.006
krebs	0.300	8	8	3	3600.014	**5**	5	392.798	6	3	3600.013	**5**	5	1625.738
dolphins	0.999	**28**	**28**	28	7.540	**28**	12	3600.010	**28**	28	0.433	**28**	23	3600.040
dolphins	0.950	**28**	**28**	4	3676.140	**28**	12	3600.010	**28**	28	29.184	**28**	18	3600.213
dolphins	0.900	**27**	**27**	4	3600.035	**27**	11	3600.010	**27**	27	67.411	**27**	17	3600.616
dolphins	0.800	**25**	**25**	4	3610.317	**25**	10	3600.010	**25**	23	3600.023	**25**	13	3600.007
dolphins	0.700	**23**	**23**	3	3601.896	**23**	10	3600.011	**23**	14	3600.025	**23**	11	3600.008
dolphins	0.600	18	18	4	3600.014	18	9	3600.009	**16**	5	3600.008	**16**	9	3600.006
dolphins	0.500	15	15	3	3600.122	**12**	7	3600.192	14	3	3600.011	**12**	8	3600.006
dolphins	0.400	11	11	3	3600.014	**8**	6	3600.073	10	3	3600.017	**8**	6	3600.006
dolphins	0.300	8	8	3	3600.014	**5**	5	1277.279	7	3	3600.011	6	5	3600.047
prison	0.999	**26**	**26**	26	5.970	**26**	15	3600.012	**26**	26	0.664	**26**	26	2952.480
prison	0.950	**26**	**26**	4	3601.860	**26**	14	3600.016	**26**	26	41.766	**26**	26	1091.526
prison	0.900	**25**	**25**	4	3600.020	**25**	14	3600.067	**25**	25	79.549	**25**	25	1704.530
prison	0.800	**24**	**24**	4	3600.021	**24**	12	3600.014	**24**	24	1509.422	**24**	21	3600.058
prison	0.700	**22**	**22**	4	3609.392	**22**	12	3600.012	**22**	17	3600.010	**22**	13	3600.008
prison	0.600	16	16	4	3600.084	**15**	11	3600.014	**15**	8	3600.011	**15**	11	3600.008
prison	0.500	13	13	3	3600.019	13	10	3600.091	13	3	3600.011	**12**	9	3600.007
prison	0.400	**10**	**10**	3	3600.017	**10**	8	3600.092	**10**	3	3600.011	**10**	8	3600.012
prison	0.300	8	8	3	3600.595	**7**	6	3600.087	8	3	3600.012	**7**	6	3600.007

Continued on next page

Table 4 – continued from previous page

		MSH	STD			STDv			REP			REPv		
Input graph	γ	best	best	bbound	time (s)	best	bbound	time (s)	best	bbound	time (s)	best	bbound	time (s)
sanjuansur	0.999	**35**	**35**	35	32.190	**35**	18	3600.018	**35**	35	1.059	**35**	35	837.826
sanjuansur	0.950	**35**	**35**	4	3600.032	**35**	18	3600.018	**35**	35	89.764	**35**	35	631.064
sanjuansur	0.900	**35**	**35**	4	3600.034	**35**	17	3600.017	**35**	35	176.950	**35**	35	740.888
sanjuansur	0.800	**31**	**31**	3	3600.027	**31**	15	3600.020	**31**	31	1473.526	**31**	31	1076.276
sanjuansur	0.700	**29**	**29**	4	3600.028	**29**	13	3600.017	**29**	20	3600.015	**29**	29	2934.521
sanjuansur	0.600	20	20	3	3605.590	20	13	3600.018	**18**	12	3600.018	**18**	14	3600.012
sanjuansur	0.500	16	16	3	3600.029	16	11	3600.130	**15**	4	3600.018	**15**	11	3600.011
sanjuansur	0.400	**12**	**12**	3	3600.029	**12**	9	3600.128	**12**	3	3600.014	**12**	9	3600.011
sanjuansur	0.300	9	9	3	3600.185	**8**	7	3600.119	9	3	3600.015	**8**	7	3600.544
jean	0.999	**35**	**35**	35	17.472	**35**	12	3600.020	**35**	35	1.709	**35**	16	3600.011
jean	0.950	**35**	**35**	4	3600.031	**35**	12	3600.019	**35**	32	3600.033	**35**	15	3600.015
jean	0.900	**33**	**33**	4	3600.032	**33**	11	3600.020	**33**	27	3600.014	**33**	14	3600.011
jean	0.800	31	**30**	3	3600.034	31	11	3600.022	**30**	11	3600.073	**30**	12	3600.013
jean	0.700	29	29	4	3600.030	29	9	3600.020	29	4	3600.015	**27**	10	3600.015
jean	0.600	25	24	4	3600.031	23	9	3600.484	24	3	3600.018	**20**	9	3600.012
jean	0.500	21	21	3	3600.030	18	8	3600.136	21	3	3600.018	**13**	7	3600.284
jean	0.400	18	18	3	3600.033	**8**	7	3600.136	14	3	3600.017	9	6	3600.124
jean	0.300	14	12	3	3600.036	**5**	5	213.099	12	2	3600.016	**5**	5	323.268
3-FullIns_3	0.999	38	**37**	37	38.251	38	9	3600.023	**37**	37	1.785	38	10	3600.013
3-FullIns_3	0.950	38	38	3	3600.161	38	9	3600.024	**37**	37	2552.954	38	9	3600.019
3-FullIns_3	0.900	38	38	3	3600.194	38	8	3600.023	**37**	30	3600.018	38	9	3600.021
3-FullIns_3	0.800	37	37	4	3600.208	37	8	3600.038	**36**	21	3600.025	37	8	3600.013
3-FullIns_3	0.700	36	36	3	3600.039	36	8	3600.030	**35**	4	3600.020	36	7	3600.013
3-FullIns_3	0.600	**17**	**17**	3	3600.034	**17**	6	3600.022	**17**	3	3600.020	**17**	6	3600.014
3-FullIns_3	0.500	**13**	**13**	3	3600.033	**13**	6	3600.023	**13**	3	3600.021	**13**	5	3600.013
3-FullIns_3	0.400	**10**	**10**	3	3600.035	**10**	5	3600.022	**10**	3	3600.016	**10**	4	3600.013
3-FullIns_3	0.300	7	7	3	3600.033	**6**	4	3600.161	7	2	3600.017	7	3	3600.014
david	0.999	**36**	**36**	36	188.911	**36**	11	3600.030	**36**	36	3.142	**36**	13	3600.021
david	0.950	35	35	4	3600.049	35	10	3600.030	**34**	34	2999.125	**34**	12	3600.017
david	0.900	**33**	**33**	3	3600.048	**33**	10	3600.031	**33**	24	3600.061	**33**	11	3600.025
david	0.800	31	31	3	3600.061	31	9	3600.032	**29**	9	3600.022	**29**	9	3600.017
david	0.700	29	29	3	3600.044	28	8	3600.031	29	3	3600.048	**26**	8	3600.017
david	0.600	26	26	3	3600.045	24	7	3600.032	24	3	3600.023	**18**	7	3600.023
david	0.500	23	22	3	3600.048	19	7	3600.032	21	3	3600.027	**14**	6	3600.017
david	0.400	20	20	3	3600.050	**9**	6	3600.033	19	2	3600.025	**9**	5	3600.017
david	0.300	16	13	3	3600.061	**6**	5	3600.188	14	2	3600.024	**6**	4	3600.017
myciel6	0.999	52	**48**	47	3600.331	52	7	3600.041	**48**	48	2.481	52	8	3600.023
myciel6	0.950	52	52	3	3600.058	52	6	3600.041	**48**	10	3600.029	51	7	3600.023
myciel6	0.900	52	52	3	3600.063	52	6	3600.040	**48**	7	3600.029	52	7	3600.020
myciel6	0.800	52	52	3	3600.062	52	5	3600.043	**48**	3	3600.030	52	6	3600.037
myciel6	0.700	**52**	**52**	3	3600.062	**52**	5	3600.042	**52**	3	3600.032	**52**	5	3600.024
myciel6	0.600	30	30	3	3600.065	30	4	3600.042	29	2	3600.032	**23**	4	3600.030
myciel6	0.500	26	26	3	3600.063	26	4	3600.041	22	2	3600.030	**18**	4	3600.023
myciel6	0.400	22	22	3	3600.060	22	3	3600.040	22	2	3600.031	**12**	3	3600.023
myciel6	0.300	18	16	2	3600.065	**6**	3	3600.042	15	2	3600.031	**6**	2	3600.022

Continued on next page

Table 4 – continued from previous page

		MSH	STD			STDv			REP			REPv		
Input graph	γ	best	best	bbound	time (s)	best	bbound	time (s)	best	bbound	time (s)	best	bbound	time (s)
4-FullIns_3	0.999	56	**55**	47	3600.107	56	12	3600.073	**55**	55	7.057	56	12	3600.039
4-FullIns_3	0.950	56	56	3	3600.403	56	11	3600.073	**55**	42	3600.054	56	12	3600.038
4-FullIns_3	0.900	56	56	3	3600.110	56	10	3600.075	**54**	33	3600.056	56	11	3600.038
4-FullIns_3	0.800	54	54	3	3600.111	54	9	3600.076	**53**	6	3600.056	54	10	3600.039
4-FullIns_3	0.700	53	53	3	3600.108	53	8	3600.074	**52**	3	3600.140	53	9	3600.038
4-FullIns_3	0.600	**26**	**26**	3	3600.109	**26**	8	3600.092	**26**	3	3600.059	**26**	8	3600.037
4-FullIns_3	0.500	21	21	3	3600.102	21	6	3600.080	21	2	3600.051	**20**	6	3600.050
4-FullIns_3	0.400	16	16	3	3600.104	**15**	5	3600.079	16	2	3600.060	**15**	5	3600.052
4-FullIns_3	0.300	**10**	**10**	3	3600.104	**10**	4	3600.072	**10**	2	3600.057	**10**	4	3600.041
ieeebus	0.999	58	**57**	52	3600.243	58	36	3600.079	**57**	57	4.933	**57**	57	2117.839
ieeebus	0.950	58	58	3	3600.116	58	36	3600.079	**57**	57	151.272	**57**	57	2112.832
ieeebus	0.900	58	**57**	3	3600.276	58	34	3600.184	**57**	57	498.775	**57**	57	1776.998
ieeebus	0.800	58	57	3	3600.246	58	27	3600.080	**57**	52	3600.075	**57**	57	2509.315
ieeebus	0.700	58	**57**	3	3600.114	58	27	3600.080	**57**	45	3600.069	**57**	57	3369.871
ieeebus	0.600	**40**	**40**	3	3600.114	**40**	23	3600.078	**40**	5	3600.061	**40**	24	3600.044
ieeebus	0.500	34	34	3	3600.116	33	18	3600.084	34	4	3600.058	**31**	20	3600.045
ieeebus	0.400	28	28	3	3600.118	26	15	3600.085	28	3	3600.109	**25**	16	3600.049
ieeebus	0.300	21	21	3	3600.117	20	12	3600.081	20	2	3600.146	**18**	12	3600.044
sfi	0.999	**65**	**65**	64	3600.114	**65**	29	3600.079	**65**	65	5.061	**65**	65	764.751
sfi	0.950	**65**	**65**	3	3600.121	**65**	28	3600.079	**65**	61	3600.056	**65**	65	1202.593
sfi	0.900	**65**	**65**	3	3600.334	**65**	27	3600.078	**65**	53	3600.217	**65**	65	2989.849
sfi	0.800	**61**	**61**	3	3600.120	**61**	24	3600.078	**61**	43	3600.194	**61**	36	3600.044
sfi	0.700	59	59	3	3600.135	59	22	3600.079	58	29	3600.059	**57**	29	3600.046
sfi	0.600	50	50	3	3600.113	48	19	3600.101	49	5	3600.059	**45**	25	3600.044
sfi	0.500	46	46	3	3600.119	41	17	3600.081	40	3	3600.146	**32**	21	3600.044
sfi	0.400	41	41	3	3600.119	35	15	3600.084	39	3	3600.062	**24**	16	3600.049
sfi	0.300	29	29	3	3600.116	**16**	13	3600.507	23	2	3600.060	**16**	12	3600.044
anna	0.999	**80**	**80**	80	264.461	**80**	18	3600.167	**80**	80	9.807	**80**	20	3600.070
anna	0.950	**79**	**79**	3	3600.199	**79**	17	3600.130	**79**	63	3600.096	**79**	19	3600.068
anna	0.900	79	79	3	3600.211	79	16	3600.137	**77**	54	3600.106	79	18	3600.076
anna	0.800	76	76	3	3600.203	76	15	3600.129	**74**	13	3600.095	76	16	3600.071
anna	0.700	73	73	3	3600.215	73	13	3600.133	**70**	3	3600.106	72	14	3600.074
anna	0.600	63	63	3	3600.227	**61**	11	3600.135	63	3	3600.091	62	12	3600.075
anna	0.500	60	60	3	3600.187	57	10	3600.134	60	2	3600.085	**51**	10	3600.074
anna	0.400	54	54	3	3600.183	48	9	3600.133	50	2	3600.232	**40**	8	3600.074
anna	0.300	45	45	3	3600.195	39	7	3600.134	38	2	3600.107	**22**	6	3600.074