Approximation algorithms and hardness results for the clique packing problem

F. Chataigner\textsuperscript{a}, G. Manić\textsuperscript{b}, Y. Wakabayashi\textsuperscript{a,\textsuperscript{*}}, R. Yuster\textsuperscript{c}

\textsuperscript{a} Instituto de Matemática e Estatística, Universidade de São Paulo, SP, Brazil
\textsuperscript{b} Instituto de Computação, Universidade Estadual de Campinas, SP, Brazil
\textsuperscript{c} Department of Mathematics, University of Haifa, Israel

\begin{abstract}
For a fixed family $\mathcal{F}$ of graphs, an $\mathcal{F}$-packing in a graph $G$ is a set of pairwise vertex-disjoint subgraphs of $G$, each isomorphic to an element of $\mathcal{F}$. Finding an $\mathcal{F}$-packing that maximizes the number of covered edges is a natural generalization of the maximum matching problem, which is just $\mathcal{F} = \{K_2\}$. In this paper we provide new approximation algorithms and hardness results for the $\mathcal{K}_r$-packing problem where $\mathcal{K}_r = \{K_2, K_3, \ldots, K_r\}$.

We show that already for $r = 3$ the $\mathcal{K}_r$-packing problem is APX-complete, and, in fact, we show that it remains so even for graphs with maximum degree 4. On the positive side, we give an approximation algorithm with approximation ratio at most 2 for every fixed $r$. For $r = 3$, 4, 5 we obtain better approximations. For $r = 3$ we obtain a simple $3/2$-approximation, achieving a known ratio that follows from a more involved algorithm of Halldórsson. For $r = 4$, we obtain a $(3/2 + \epsilon)$-approximation, and for $r = 5$ we obtain a $(25/14 + \epsilon)$-approximation.
\end{abstract}

\section{Introduction}

Let $\mathcal{F}$ be a fixed family of graphs. An $\mathcal{F}$-packing in a graph $G$ is a set of pairwise vertex-disjoint subgraphs of $G$, each isomorphic to an element of $\mathcal{F}$. We say that an $\mathcal{F}$-packing covers an edge (resp. vertex) of $G$ if one of the subgraphs of the packing contains that edge (resp. vertex). In this paper the $\mathcal{F}$-packing problem is the problem of finding the maximum number of edges that can be covered by an $\mathcal{F}$-packing. When $\mathcal{F} = \{K_2\}$, this simply corresponds to the maximum matching problem. Apart from its theoretical interest, this problem is also important from a practical point of view, as it arises naturally in applications such as scheduling.

Another related problem is that of finding an $\mathcal{F}$-packing in a graph $G$ that covers the maximum number of vertices. To avoid confusion, we refer to this problem as the $\mathcal{F}_V$-packing problem. This problem is NP-hard, even when $\mathcal{F}$ consists of a single graph that has a component with at least three vertices [5]; and also when $\mathcal{F}$ contains only complete graphs with at least three vertices [6]. On the other hand, the $\mathcal{F}_V$-packing problem is polynomially solvable for some non-trivial classes of families $\mathcal{F}$, and many important results in matching theory can be generalized to those cases. For example, when $\mathcal{F} = \{K_2, \ldots, K_r\}$, $r > 2$, Hell and Kirkpatrick [6] showed that this problem is in P.

Let $\mathcal{K}_r = \{K_2, \ldots, K_r\}$. In contrast with the above result of Hell and Kirkpatrick, we show, in Section 3, that the $\mathcal{K}_r$-packing problem is NPX-complete already for $r = 3$, and, in fact, already for graphs with maximum degree 4.

On the positive side, we show in Section 2 that a simple greedy algorithm yields a 2-approximation for the $\mathcal{K}_r$-packing problem. A modified greedy algorithm, which is based on application of the local search method of Hurkens and Schrijver [7]...
yields better approximation ratios for \( r = 4, 5 \). The analysis of these ratios is also somewhat more complicated than the analysis of the simple greedy algorithm. In particular, for \( r = 4 \) we obtain a \((3/2 + \varepsilon)\)-approximation and for \( r = 5 \) we obtain a \((25/14 + \varepsilon)\)-approximation.

In Section 4 we specifically address the \( K_3 \)-packing problem. We show, in fact, that a tighter analysis of the simple greedy algorithm yields a \( 3/2 \)-approximation for it. More generally, we show that there is a \((1 + \frac{1}{2} \rho)\)-approximation algorithm for the \( K_3 \)-packing problem, whenever there is a \( \rho \)-approximation algorithm for the triangle packing problem. In particular, for the class of graphs with maximum degree 4, using a result of [8] for the triangle packing problem, we derive a \( 1.4\)-approximation algorithm for the \( K_3 \)-packing problem for this class of graphs.

An extended abstract mentioning the results of this paper has appeared in the Proceedings of Eurocomb 2007 [3].

1.1. Basic definitions and notation

All graphs considered here are simple. If \( G \) is a graph, then \( V_G \) (resp. \( E_G \)) denotes its vertex (resp. edge) set. The number of vertices of \( G \) is denoted by \( n_G \), and the maximum degree by \( \Delta(G) \).

Let \( F \) be a fixed family of graphs. We recall that in the \( F \)-packing problems to be investigated in this paper we are interested in maximizing the number of edges that are covered. The set \{\( K_2, \ldots, K_r \)\} is abbreviated as \( K_r \). A complete graph of order \( k \) is called a \( k \)-clique. A 3-clique is called a triangle. If \( A \) is an \( F \)-packing of a graph \( G \), then the value of \( A \), denoted \( \text{val}_F(G, A) \) (or simply \( \text{val}(A) \)), is the number of edges of \( G \) that it covers. We denote by \( \mathcal{P}_A \) (resp. \( \mathcal{Q}_A, \mathcal{T}_A, \mathcal{E}_A \)) the collection of all 5-cliques (resp. 4-cliques, triangles, edges) in \( A \). Furthermore, we denote by \( G[A] \) the subgraph of \( G \) induced by the set of edges in \( A \). If \( G \) is an instance of the \( F \)-packing problem, then \( \text{opt}_F(G) \) denotes the value of an optimal solution for \( G \).

We call a graph irredundant if each of its edges belongs to some triangle.

In this paper we refer to a heuristic of Hurbek and Schrijver [7], denoted as HS(k, t), that finds a maximum set of vertex-disjoint \( k \)-cliques in a graph \( G \). It is a local search greedy heuristic that, starting with any collection of \( k \)-cliques, while possible, replaces at most \( p - 1 \) \( k \)-cliques in the current collection with a set of \( p \leq t \) disjoint \( k \)-cliques that are not in the current collection, and updates the current collection. The parameter \( t \) is an integer. Its approximation ratio is \( k/2 + \varepsilon \), where \( \varepsilon \) depends on \( t \).

Given a parameter \( \rho \geq 1 \), a \( \rho \)-approximation algorithm for a maximization problem \( \Pi \) is a polynomial-time algorithm that, for any instance \( I \) of \( \Pi \) produces a solution \( S \) whose value, \( \text{val}_\Pi(I, S) \), is at least \( \frac{1}{\rho} \text{opt}_\Pi(I) \), where \( \text{opt}_\Pi(I) \) is the value of an optimal solution for \( I \) (we also say that \( \rho \) is the approximation ratio). If such an algorithm exists, we say that \( \Pi \) belongs to \( \text{APX} \). Let \( \Pi_1 \) and \( \Pi_2 \) be optimization problems. An L-reduction from \( \Pi_1 \) to \( \Pi_2 \) consists of a pair of polynomial-time computable functions \((f, g)\) such that, for two fixed positive constants \( \alpha \) and \( \beta \) the following hold:

(C1) For every instance \( I_1 \) of \( \Pi_1, f(I_1) \) is an instance of \( \Pi_2 \), and \( |\text{opt}_{\Pi_2}(f(I_1))| \leq \alpha |\text{opt}_{\Pi_1}(I_1)| \).

(C2) Given an instance \( I_1 \) of \( \Pi_1 \), and any feasible solution \( A \) for \( f(I_1) \), we have that \( g(I_1, A) \) is a feasible solution for the instance \( I_1 \) of \( \Pi_1 \), and \( |\text{opt}_{\Pi_2}(I_1, g(I_1, A))| \leq \beta |\text{opt}_{\Pi_2}(f(I_1))| - \text{val}_{\Pi_2}(f(I_1), A) | \).

We denote by \( \Pi_1 \preceq \Pi_2 \) the existence of an L-reduction from \( \Pi_1 \) to \( \Pi_2 \). If \( \Pi_1 \preceq \Pi_2 \) and \( \Pi_2 \in \text{APX} \), then \( \Pi_1 \in \text{APX} \). A problem \( \Pi \) is \( \text{APX-hard} \) if, for every \( \Pi' \in \text{APX} \), we have \( \Pi' \preceq \Pi \). If an \text{APX-hard} problem belongs to the class \( \text{APX} \), then it is \( \text{APX-complete} \). It is known that an \( \text{APX-hard} \) problem does not admit a PTAS, unless \( P = \text{NP} \) [1].

One of the reductions we show in Section 3 considers the following restricted version of the MAX SAT problem, denoted here simply as SAT, known to be \( \text{APX-complete} \) [1]. Given a collection of disjunctive clauses \( C = \{c_1, c_2, \ldots, c_l\} \) over a set \( X = \{x_1, x_2, \ldots, x_n\} \) of variables, such that each clause has at most 2 literals, and each variable appears in at most 3 of the clauses (counting both positive and negated occurrences), find a truth assignment for the variables of \( X \) that satisfies as many clauses as possible.

2. Approximation algorithm for the \( K_r \)-packing problem

All algorithms we describe in this paper have a common structure. This common structure will be presented in a form of a generic algorithm, called here BASE. To distinguish the different algorithms we can derive from this basic algorithm, we assume that this algorithm calls a generic \text{PROCEDURE} \( \mathcal{P}_q \) that outputs a \( \{K_q\} \)-packing of a given input graph \( G \). The different algorithms are then obtained by substituting \( \mathcal{P}_q \) by specific algorithms.

**Algorithm BASE**

Input: A graph \( G \).

\text{Subroutine: \text{PROCEDURE} \( \mathcal{P}_q \) that outputs a \( \{K_q\} \)-packing of a given input graph}

Output: A \( \{K_2, \ldots, K_r\} \)-packing of \( G \).

1. for \( q = r \) downto \( 3 \) do
2. \( F_q \leftarrow \{K_q\} \)-packing of \( G \) output by the \text{PROCEDURE} \( \mathcal{P}_q \)
3. \( G \leftarrow G - F_q \)
4. \( F_2 \leftarrow \) a maximum matching in \( G \)
5. return \( F_2 \cup \ldots \cup F_q \)
2.1. A simple greedy algorithm

Denote by $g_r$ the simple Greedy Algorithm that consists of the algorithm Basic, in which the Procedure $P_q$ (called at line 2) is an algorithm that simply selects a maximal set of vertex-disjoint $q$-cliques in a graph. We say $g_r$ is a greedy algorithm because it selects first the larger cliques.

The next lemma is the key to obtaining the approximation ratio of $g_r$.

**Lemma 2.1.** Let $G$ be a graph, $r \geq 2$ an integer, and $\mathcal{A}$ a solution returned by the algorithm $g_r$ applied to $G$. If $C$ is a $q$-clique in $G$, where $2 \leq q \leq r$, then $\sum_{\mathcal{V} \in \mathcal{C}} d_{\mathcal{G}(\mathcal{A})}(v) \geq 1/2(q(q-1))$.

**Proof.** The proof is by induction on $r$. For $r = 2$, it suffices to note that if $C$ is a 2-clique in $G$ then at least one of its vertices intersects an edge of $\mathcal{A}$ (a maximal matching returned by the algorithm). Thus, $\sum_{\mathcal{V} \in \mathcal{C}} d_{\mathcal{G}(\mathcal{A})}(v) \geq 1$ and the lemma holds.

Suppose now that $r > 2$. Let $C$ be a $q$-clique of $G$, $2 \leq q \leq r$, and let $l$ be the number of vertices in the intersection of $C$ and $F_i$. Set $F_i := \bigcup_{l=2}^{r-1} F_i$. Then, we have that

$$\sum_{v \in \mathcal{V}} d_{\mathcal{G}(\mathcal{A})}(v) = \sum_{v \in \mathcal{V}} d_{\mathcal{G}(\mathcal{A})}(v) \geq \sum_{v \in \mathcal{V}} d_{\mathcal{G}(\mathcal{A})}(v) = l(r - 1) + \sum_{\mathcal{V} \in \mathcal{C}} d_{\mathcal{G}(\mathcal{A})}(v).$$

Note that $C - F_i$ is isomorphic to $K_{q-1}$. When $q < r$ we have that $q - l < r$. When $q = r$, since $\mathcal{P}_r$ is the algorithm that simply selects a maximal set of vertex-disjoint $r$-cliques, we have that $l \geq 1$, and again $q - l < r$. Thus, we can apply the induction hypothesis on the last term of the equation, obtaining

$$\sum_{v \in \mathcal{V}} d_{\mathcal{G}(\mathcal{A})}(v) \geq (l(r - 1) + 1/2(q - l)(q - l - 1)) = 1/2((l^2 + (2r - 2q - 1)l + q(q - 1))).$$

If $q < r$, then $2r - 2q - 1 > 0$, so the minimum for the right-hand side of (1) is reached at $l = 0$. In that case we have $\sum_{v \in \mathcal{V}} d_{\mathcal{G}(\mathcal{A})}(v) \geq 1/2q(q - 1)$. If, however, $q = r$, then the minimum for the right-hand side of (1) is reached at $l = \frac{1}{2}$. Since $\mathcal{P}_r$ selects a maximal set of vertex-disjoint $r$-cliques, we have that $l = 0$ is not possible. Thus, the minimum for the right-hand side of (1) is reached at $l = 1$ and is $\frac{1}{2}q(q - 1)$. So, the proof is now complete. ■

**Theorem 2.2.** For $r \geq 2$, the algorithm $g_r$ is a 2-approximation algorithm for the $K_r$-packing problem.

**Proof.** Let $\mathcal{A}$ be a solution returned by the algorithm $g_r$ applied to a graph $G$. Consider an optimal $K_r$-packing $\mathcal{O}$ in $G$. Applying Lemma 2.1 to each clique $C$ of $\mathcal{O}$, we get

$$2 \text{val}(\mathcal{A}) = \sum_{v \in \mathcal{V}} d_{\mathcal{G}(\mathcal{A})}(v) \geq \sum_{c \in \mathcal{O}} \sum_{v \in \mathcal{V}} d_{\mathcal{G}(\mathcal{A})}(v) \geq \sum_{c \in \mathcal{O}} \frac{1}{2}|\mathcal{V}_c|(|\mathcal{V}_c| - 1) = \sum_{c \in \mathcal{O}} |E_c| = \text{val}(\mathcal{O}).$$

The first inequality follows from the fact that in the first sum we consider the degrees in $\mathcal{G}(\mathcal{A})$ of all vertices of $G$, and in the second sum we consider the degrees in $\mathcal{G}(\mathcal{A})$ of those vertices in $G$ that belong to the cliques of $\mathcal{O}$ (we may have vertices in $\mathcal{G}(\mathcal{A})$ that do not belong to $\mathcal{O}$). The second inequality follows from Lemma 2.1. ■

**Remark 1.** In the proof of Lemma 2.1 we did not use the fact that $F_2$ is a maximum matching (see step 4 of the algorithm Basic). That is, we may substitute step 4 by “$F_2 \leftarrow$ a maximal matching in $G$”, and obtain the same result. In other words, if we consider that $g_r$ (the Greedy Algorithm) simply uses the Procedure $P_q$ for $q = r, \ldots, 2$, the statement of Theorem 2.2 holds.

**Remark 2.** We note that the upper bound 2 for the approximation ratio of algorithm $g_r$ is not tight. In Section 4 we show that the algorithm $g_3$ has, in fact, approximation ratio 3/2. The analysis is somewhat more delicate, however.

2.2. A modified greedy algorithm based on local search

Denote by $\mathcal{B}_3$, the algorithm Basic, in which the Procedure $P_q$ is the heuristic HS($q$, $t$), when $q = r$ (see Section 1.1), and for $2 < q < r$, $P_q$ is the algorithm that simply selects a maximal set of vertex-disjoint $q$-cliques.

**Theorem 2.3.** The algorithm $\mathcal{B}_4$ is a $(3/2 + \varepsilon)$-approximation algorithm for the $K_4$-packing problem.

**Proof.** Let $\mathcal{B}$ be an optimal solution and $\mathcal{B}$ be the solution returned by the algorithm $\mathcal{B}_4$. Thus, $\text{val}(\mathcal{B}) = 6|\mathcal{B}_0| + 3|\mathcal{B}_0| + |\mathcal{E}_o|$ and $\text{val}(\mathcal{B}) = 6|\mathcal{B}_0| + 3|\mathcal{B}_0| + |\mathcal{E}_o|.

Let $q_i$, $0 \leq i \leq 4$, be the number of 4-cliques of $\mathcal{B}_0$ that intersect precisely $i$ vertices of $\mathcal{B}_0 \cup \mathcal{B}_3$. Let $t$, $0 \leq t \leq 3$, be the number of triangles of $\mathcal{B}_3$ that intersect precisely $t$ vertices of $\mathcal{B}_0 \cup \mathcal{B}_3$. Note that since $\mathcal{P}_3$ selects a maximal set of vertex-disjoint 3-cliques, we have that $t_0 = 0$. Furthermore, since HS($4$, $t$) returns a maximal collection of 4-cliques, we have that $q_0 = 0$. Suppose now that $q_t > 0$. Then, there is a 4-clique, say $D$, in $\mathcal{B}_0$ that intersects precisely one vertex of $\mathcal{B}_0 \cup \mathcal{B}_3$. The three other vertices of $D$ would form a triangle that does not intersect $\mathcal{B}_0 \cup \mathcal{B}_3$, contradicting the fact that $\mathcal{P}_3$ selects a maximal set of vertex-disjoint 3-cliques. Thus, $q_t = 0$. 


Observe that the number of vertices of $Q_B \cup T_B$ covered by $Q_B \cup T_B$ is $2q_2 + 3q_3 + 4q_4 + t_1 + 2t_2 + 3t_3$. Thus, the number of vertices of $Q_B \cup T_B$ not covered by $Q_B \cup T_B$ is $w := 4|Q_B| + 3|T_B| - (2q_2 + 3q_3 + 4q_4 + t_1 + 2t_2 + 3t_3)$, hence, the number of edges of $\mathcal{E}_B$ with at least one endpoint in a clique of $Q_B \cup T_B$ is at most $w$.

Now, let $z := |\mathcal{E}_B| - w$. Note that at least $\max(0, z)$ edges of $\mathcal{E}_B$ are disjoint from $Q_B \cup T_B$. Furthermore, every triangle of $\mathcal{T}_B$ (resp. 4-clique of $Q_B$) that intersects precisely 1 vertex (resp. 2 vertices) of $Q_B \cup T_B$ contributes an edge that is disjoint from $Q_B \cup T_B$. Since $\mathcal{E}_B$ is a maximum matching of $G - \{v: v$ is a vertex in $Q_B \cup T_B\}$, we have

$$|\mathcal{E}_B| \geq q_2 + q_3 + q_4 + \max(0, z). \quad (2)$$

Using the facts that $|Q_B| = q_2 + q_3 + q_4$ and $|\mathcal{T}_B| = t_1 + t_2 + t_3$, we can rewrite $z$ obtaining

$$z = |\mathcal{E}_B| - 4|Q_B| - 3|T_B| + 3|Q_B| + 2|T_B| - q_2 - q_4 - t_1 - t_3. \quad (3)$$

Since $\text{val}(\mathcal{B}) = 6|Q_B| + 3|T_B| + |\mathcal{E}_B|$, using (2) we get

$$\text{val}(\mathcal{B}) \geq 6|Q_B| + 3|T_B| + q_2 + t_1 + \max(0, z) \geq 6|Q_B| + 3|T_B| + q_2 + t_1 + z.$$  

Now substituting the value of $z$ given in (3), we obtain

$$\text{val}(\mathcal{B}) \geq 2|Q_B| + 3|Q_B| + 2|T_B| + |\mathcal{E}_B|.$$

(4)

Combining the fact that $Q_B$ is the solution output by $\text{HS}(4, t)$, which has an approximation ratio $2 + \epsilon$, and the fact that $\text{opt}_{X4}(G) \geq |Q_B|$, we have

$$|Q_B| \geq \left(\frac{1}{2} - \epsilon'\right) \text{opt}_{X4}(G) \geq \left(\frac{1}{2} - \epsilon'\right) |Q_B|.$$

The above inequality together with (4) implies that

$$\text{val}(\mathcal{B}) \geq (4 - 2\epsilon') |Q_B| + 2|T_B| + |\mathcal{E}_B| \geq \left(\frac{2}{3} - \epsilon'\right) (6|Q_B| + 3|T_B| + |\mathcal{E}_B|) = \left(\frac{2}{3} - \epsilon'\right) \text{val}(\mathcal{G}).$$

**Theorem 2.4.** The algorithm $\mathcal{B}_5$ is a $(25/14 + \epsilon)$-approximation algorithm for the $K_3$-packing problem.

**Proof.** The proof is similar to the one presented for Theorem 2.3. Let $\mathcal{G}$ be an optimal solution and $\mathcal{B}$ be the solution returned by the algorithm $\mathcal{B}_5$. Thus, $\text{val}(\mathcal{G}) = 10|P_B| + 6|Q_B| + 3|T_B| + |\mathcal{E}_B|$ and $\text{val}(\mathcal{B}) = 10|P_B| + 6|Q_B| + 3|T_B| + |\mathcal{E}_B|$.

Let $p_i, 0 \leq i \leq 5$, be the number of 5-cliques of $P_B$ that intersect precisely $i$ vertices of $P_B \cup Q_B \cup T_B$. Let $q_i, 0 \leq i \leq 4$, be the number of 4-cliques of $Q_B$ that intersect precisely $i$ vertices of $P_B \cup Q_B \cup T_B$. Let $t_i, 0 \leq i \leq 3$, be the number of triangles of $T_B$ that intersect precisely $i$ vertices of $P_B \cup Q_B \cup T_B$. Similarly as in the proof of Theorem 2.3, we get $p_0 = q_0 = t_0 = p_1 = p_2 = 0$.

Observe that the number of vertices of $P_B \cup Q_B \cup T_B$ not covered by $P_B \cup Q_B \cup T_B$ is $5|P_B| + 4|Q_B| + 3|T_B| - (3p_3 + 4p_4 + 5p_5 + 2q_1 + 3q_3 + 4q_4 + t_1 + 2t_2 + 3t_3)$. We now define $z := |\mathcal{E}_B| - 5|P_B| - 4|Q_B| - 3|T_B| + 3p_1 + 3p_2 + 4p_3 + 5p_4 + 2q_2 + 3q_3 + 4q_4 + t_1 + 2t_2 + 3t_3$.

Observe that at least $\max(0, z)$ edges of $\mathcal{E}_B$ are disjoint from $P_B \cup Q_B \cup T_B$. Furthermore, every triangle of $T_B$ (resp. 4-clique of $Q_B$, 5-clique of $P_B$) that intersects precisely 1 vertex (resp. 2 vertices, 3 vertices) of $P_B \cup Q_B \cup T_B$ contributes an edge that is disjoint from $P_B \cup Q_B \cup T_B$. Since $\mathcal{E}_B$ is a maximum matching of $G - \{v: v$ is a vertex in $P_B \cup Q_B \cup T_B\}$, we have

$$|\mathcal{E}_B| \geq t_1 + q_2 + p_3 + \max(0, z). \quad (5)$$

Using the facts that $|P_B| = p_1 + p_4 + p_5$, $|Q_B| = q_2 + q_3 + q_4$ and $|T_B| = t_1 + t_2 + t_3$, we can rewrite $z$ obtaining

$$z = |\mathcal{E}_B| - 5|P_B| - 4|Q_B| - 3|T_B| + 4|P_B| + 3|Q_B| + 2|T_B| - p_3 + p_5 - q_2 + q_4 + t_1 + t_3. \quad (6)$$

Now, using (5) we get

$$\text{val}(\mathcal{B}) = 10|P_B| + 6|Q_B| + 3|T_B| + |\mathcal{E}_B| \geq 10|P_B| + 6|Q_B| + 3|T_B| + p_3 + q_2 + t_1 + \max(0, z).$$

Thus

$$\text{val}(\mathcal{B}) \geq 10|P_B| + 6|Q_B| + 3|T_B| + p_3 + q_2 + t_1 + z.$$  

Substituting the value of $z$ given in (6) and discarding some terms we obtain

$$\text{val}(\mathcal{B}) \geq 5|P_B| + 2|Q_B| + 4|P_B| + 3|Q_B| + 2|T_B| + |\mathcal{E}_B|.$$  

(7)

Observe now that each element of $P_B \cup Q_B$ intersects $P_B \cup Q_B$ in at most 5 vertices. Thus, $|P_B| + |Q_B| \geq \frac{1}{5}(|P_B| + |Q_B|)$.

(8)

Indeed, if $|P_B| + |Q_B| < \frac{1}{5}(|P_B| + |Q_B|)$, there would be an element of $P_B \cup Q_B$ that does not intersect any of the elements from $P_B \cup Q_B$, contradicting the fact that the set $P_B \cup Q_B$ was found by algorithm $\mathcal{B}_5$. 


Since HS$(5, t)$ has an approximation ratio $5/2 + \varepsilon$, we have
\[
|\mathcal{P}_B| \geq \left( \frac{2}{5} - \varepsilon' \right) |\mathcal{P}_T|.
\]
Now multiplying inequality (8) by 2 and adding with the inequality above multiplied by 3, we get
\[
5 |\mathcal{P}_B| + 2 |\mathcal{Q}_B| \geq (8/5 - 3\varepsilon') |\mathcal{P}_T| + 2/5 |\mathcal{Q}_T|. \tag{8}
\]
Combining this inequality with inequality (7) we obtain
\[
\text{val}(B) \geq (8/5 - 3\varepsilon') |\mathcal{P}_T| + 2/5 |\mathcal{Q}_T| + 4 |\mathcal{P}_T| + 3 |\mathcal{Q}_T| + 2 |\mathcal{E}_T| + |\mathcal{E}_S|
\]
\[
= \frac{28}{10} (\frac{28}{5} - 3\varepsilon') |\mathcal{P}_T| + 6 |\mathcal{Q}_T| + 1 \left( \frac{17}{5} \right) 6 |\mathcal{Q}_T| + 1 |\mathcal{E}_S|
\]
\[
\geq \left( \frac{28}{50} - \varepsilon' \right) \left( 10 |\mathcal{P}_T| + 6 |\mathcal{Q}_T| + 3 |\mathcal{E}_T| + |\mathcal{E}_S| \right)
\]
\[
= \frac{14}{25} - \varepsilon' \text{val}(T). \quad \blacksquare
\]

We are not sure whether the ratio $(3/2 + \varepsilon)$ (resp. $(25/14 + \varepsilon)$) for the algorithm $\mathcal{B}_4$ (resp. $\mathcal{B}_5$) is tight. We note that for $r \geq 6$, using the same approach it is not possible to show that the ratio of the algorithm $\mathcal{B}_r$ is smaller than $2$ (as we need a better ratio for HS$(r, t)$).

3. The \textbf{apx}-hardness of the $\mathcal{K}_3$-packing problem

In this section we prove that the $\mathcal{K}_3$-packing problem is $\textbf{apx}$-hard on graphs with maximum degree 5. We also show that this problem is $\textbf{apx}$-hard even on irredundant graphs with maximum degree 4. We recall that we defined a graph to be irredundant if each of its edges belongs to some triangle. As we know that the $\mathcal{K}_3$-packing problem has a constant approximation algorithm, we can conclude that it is an $\textbf{apx}$-complete problem.

We show first the result for graphs with maximum degree 5, and then for graphs with maximum degree 4. In both cases we consider the problem of finding the maximum number of vertex-disjoint triangles in a graph, denoted here as VTP, and known to be $\textbf{apx}$-complete [2]. (This problem is equivalent to the $\{K_3\}$-packing problem; it is just more convenient to simplify the counting arguments.)

The second proof is significantly more elaborate than the first: its structure is analogous to the reduction presented by Caprara and Rizzi [2] to show that the VTP problem is $\textbf{apx}$-complete on graphs with maximum degree 4.

\textbf{Theorem 3.1.} The $\mathcal{K}_3$-packing problem is $\textbf{apx}$-hard on graphs with maximum degree 5.

\textbf{Proof.} We show an L-reduction from the VTP problem to the $\mathcal{K}_3$-packing problem. For that, we shall exhibit a pair of functions $(f, g)$, and constants $\alpha$ and $\beta$, in accordance with the definition of L-reduction given in Section 1.

Let $G$ be an irredundant graph with $\Delta(G) = 4$. Define $G' := f(G)$ as the union of two copies, say $G_1$ and $G_2$, of $G$ together with the set of edges
\[
\{u_1u_2: u_1 \in V_{G_1}, u_2 \in V_{G_2}, \text{ and } u_1, u_2 \text{ correspond to the same vertex } u \in V_G\}.
\]

We first show that
\[
\text{opt}_{\mathcal{K}_3}(G') = 3 \text{opt}_{\mathcal{VTP}}(G) + n_G. \tag{9}
\]
Indeed, if $T^*$ is an optimal solution of the VTP problem in $G$, then there is a $\{K_2, K_3\}$-packing of $G'$ consisting of the triangles in $G_1$ and $G_2$ that are copies of triangles in $T^*$, and set of edges $\{u_1u_2: u_1 \in V_{G_1}, u_2 \in V_{G_2}, \text{ and } u_1, u_2 \text{ correspond to the same vertex } u \in V_G\}$. Since the number of vertices of $G$ not covered by $T^*$ is $n_G - 3 \text{opt}_{\mathcal{VTP}}(G)$, we have
\[
\text{opt}_{\mathcal{K}_3}(G') \geq 6 \text{opt}_{\mathcal{VTP}}(G) + n_G - 3 \text{opt}_{\mathcal{VTP}}(G) = 3 \text{opt}_{\mathcal{VTP}}(G) + n_G.
\]
On the other hand, if an optimal solution of the $\mathcal{K}_3$-packing problem in $G'$ has $t'$ triangles and $e'$ edges, since $e' \leq \frac{n_G - 3t'}{2} = n_G - \frac{3}{2}t'$, we have
\[
\text{opt}_{\mathcal{K}_3}(G') = 3t' + e' \leq \frac{3}{2}t' + n_G.
\]
Of course, $t' \leq 2 \text{opt}_{\mathcal{VTP}}(G)$, and thus $\text{opt}_{\mathcal{K}_3}(G') \leq 3 \text{opt}_{\mathcal{VTP}}(G) + n_G$. Hence, statement (9) holds.

Let $T^*$ be an optimal solution of the VTP problem in $G$. Suppose that there exists a triangle $T \in T^*$, such that $T$ has 5 neighbouring vertices in $V_G \setminus V_T$ that are not covered by $T^*$. Since $\Delta(G) = 4$, one pair of them, say $v_1, v_2$, is adjacent to the same vertex, say $x$ from $V_T$; another pair, say $v_3, v_4$ (disjoint from $v_1, v_2$), is adjacent to the same vertex, say $y$ from $V_T$. Note that the third vertex of $V_T$, say $z$, has degree at least 3. Furthermore, since $G$ is irredundant and $\Delta(G) = 4$, we have that $v_1v_2, v_3v_4 \in E_G$. Indeed, since $G$ is irredundant, edge $v_1x$ (resp. $v_2x$) has to be in some triangle. Since $d_G(x) = d_G(y) = \Delta(G) = 4$, the only possible triangle having edge $xv_1$, not using $v_1v_2$, is the triangle $[x, v_1, z]$ (see Fig. 1). But now, the only possible triangle having edge $xv_2$ is the triangle $[x, v_2, v_1]$, and hence, $v_1v_2 \in E_G$. Similarly, $v_3v_4 \in E_G$. 


The WeshowanL-reduction from the SAT problem has more triangles than $T^*$ does, a contradiction. Hence, each triangle from $T^*$ has at most 4 neighbours not covered by $T^*$. Note, furthermore, that since $G$ is irredundant, each vertex not covered by $T^*$ is adjacent to at least one vertex covered by $T^*$. Indeed, suppose that there is a vertex $v$ not covered by $T^*$, and not adjacent to any vertex covered by $T^*$. Since $G$ is irredundant, $v$ is a vertex of a triangle $T$. Observe that none of the vertex of $T$ is covered by $T^*$, and thus, $T^*$ is not an optimal solution of the SAT problem in $G$, a contradiction. It thus follows that the number of vertices in $G$ not covered by $T^*$ is at most 4 opt$_{VTP}(G)$, that is, $n_G - 3 \leq$ opt$_{VTP}(G) \leq 4$ opt$_{VTP}(G)$. Using (9) we have opt$_{X_j}(G') \leq 10$ opt$_{VTP}(G)$. Thus, for $\alpha = 10$ condition (C1) of the definition of L-reduction is satisfied.

Given a $[K_2, K_3]$-packing $A'$ of $G = f(G)$, we define $g(G, A)$ as the largest of the two sets $T_A \cap G_2 = T_A \cap G_2$. Suppose, without loss of generality, that $g(G, A) = T_A \cap G_1$. Let $t_i := |T_A \cap G_1|, t_i' := |T_A \cap G_2|, e_i := |E_A \cap G_1|, e_i' := |E_A \cap G_2|$, and $e'$ be the number of edges in $E_A$ with one endpoint in $G_1$ and the other in $G_2$. Of course, $t_i' \leq$ opt$_{VTP}(G)$. Thus, $1/2 t_i' + 1/2 t_i - 2$ opt$_{VTP}(G) \leq 0$. Since $t_i \leq t_i'$, we have $1/2 t_i' + 1/2 t_i' - 2$ opt$_{VTP}(G) \leq 0$, or equivalently,

$$\text{opt}_{VTP}(G) - t_i \leq 3 \text{opt}_{VTP}(G) + \left(\frac{1}{2} t_i' + 2 e_i + e'_i + e' \right) - (3 t_i' + 3 t_i' + e_i' + e_i' + e').$$

(10)

Now, $3 t_i' + 3 t_i' + 2 e_i + 2 e'_i + 2 e' \leq n_G = 2 n_G$, and hence, $1/2 t_i' + 1/2 t_i' + e_i' + e_i' + e' \leq n_G$. Thus, from (10) we have $\text{opt}_{VTP}(G) - t_i' \leq 3 \text{opt}_{VTP}(G) + n_G - (3 t_i' + 3 t_i' + e_i' + e_i' + e')$. Using (9), we get $\text{opt}_{VTP}(G) - t_i' \leq \text{opt}_{X_j}(G') - \text{val}_{X_j}(G', A)$. Thus, condition (C2) holds with $\beta = 1$.

**Theorem 3.2.** The $K_3$-packing problem is APX-hard on the class of irredundant graphs with maximum degree 4.

**Proof.** We show an L-reduction from the SAT problem we have defined in Section 1. For that, as in the previous proof, we shall exhibit a pair of functions ($f$, $g$), and constants $\alpha$ and $\beta$, according to the definition of L-reduction given in Section 1. Let $\psi = (C, X)$ with $C = \{c_1, c_2, \ldots, c_l\}$ and $X = \{x_1, x_2, \ldots, x_p\}$ be an instance of SAT. Let $m_i$ denote the number of occurrences of $x_i$. We may assume, without loss of generality, that $m_i \geq 2$ (for if $x_i$ appears only in one clause we can set $x_i$ to the value which satisfies that clause). We define $G := f(\psi)$ in the following way.

To each clause $c_j$ we associate a test component $C_j$. The test component of a clause with two literals consists of 4 triangles $[t_1, t_1', r_1^1, r_1^2], [s_1, r_1^1, t_1^1], [s_1, r_1^2, t_1^2]$, (see Fig. 2(a)), whereas the test component associated with a clause with one literal consists of 3 triangles $[t_1, s_1, r_1^1], [s_1, r_1^1, t_1^1], [s_1, r_1^2, t_1^2]$ (see Fig. 2(b)).

To each variable $x_i$ we associate a truth component $X_i$ (see Fig. 2(c)). This component consists of $2m_i$ triangles $T_1, T_2, \ldots, T_{2m_i}$, where $T_{2k-1} = [a_1^k, a_1^k(b_1^k, u_1^k)]$ and $T_{2k} = [b_1^k, u_1^k(v_1^k)], k = 1, \ldots, m_i$ (all upper indices being modulo $m_i$). The parity of $T_k$ is the parity of $k$.

The graph $G$ is obtained by connecting the test and truth components as follows. Let $c_j$ be a clause with two literals and let $x_1, x_2$ be the variables which occur in $c_j$. If $x_1$ occurs positive (resp. negated) in $c_j$, then identify the vertex $t_1^j$ of the test component $C_j$, with a vertex $a_1^k$ (resp. $b_1^k$) of the truth component $X_i$ which has not yet been involved in any identification.
Similarly, let $c_j$ be a clause with one literal, say, $x_i$. If $x_i$ occurs positive (resp. negated) in $c_j$, then identify the vertex $t^1_j$ of $C_j$, with a vertex $a^+_i$ (resp. $b^+_i$) of $X_1$ which has not yet been involved in any identification. Note that $G'$ is irredundant and $\Delta(G') = 4$.

A maximal $\{K_2, K_3\}$-packing $A$ of $G'$ is said to be canonical if, for each truth component, it contains either all even or all odd triangles, and for each test component $C_i$ it contains the triangle $[t^1_i, t^2_i, s^1_i]$, and possibly one of the edges $t^1_i s^1_i$ or $t^2_i s^1_i$.

First, we show that the following statement holds.

Given a non-canonical $\{K_2, K_3\}$-packing $A$ of $G'$, one can find in polynomial time a canonical packing of $G'$ whose value is at least the value of $A$.

We will construct the desired packing $A'$ from $A$ (we start with $A' = A$). Initially, for each test component $C_j$, $1 \leq j \leq l$, we remove from $A'$ the triangles and edges that are in $C_j$ and add $[t^1_j, t^2_j, s^1_j]$ to it. Furthermore, if one of the edges $t^1_j s^1_j, t^2_j s^1_j$ is covered by $A$, then we add to $A'$ the one that is covered by $A$. Observe that for each $C_j$, the value of $A$ restricted to $C_j$ is at most 4. Moreover, if the value of $A$ restricted to $C_j$ is exactly 4, then one of the edges $t^1_j s^1_j, t^2_j s^1_j$ is covered by $A$. Thus, so far the resulting packing $A'$ has a value that is at least the value of $A$.

Moreover, for each $i$, $1 \leq i \leq p$, if the triangles of the truth component $X_i$ that are in $T_A$ are not all of the same parity, we do the following (depending on the number of occurrences of $x_i$).

1. $m_i = 3$.

We may assume, without loss of generality, that $x_i$ appears negated in one clause, say $c_j$, and positive in two clauses (for if $x_i$ appears only negated or only positive, we can set it to the value that satisfies all the clauses in which it appears in). Let $t^j_k, k \in \{1, 2\}$ be the vertex of $C_j$ incident with $X_i$. Then, we remove from $A'$ the triangles and edges that are in $X_i$, and add all even triangles of $X_i$ to $A'$. Furthermore, if $t^1_i s^1_j$ is in $A'$, we remove it. We next show that after those changes the value of $A'$ is at least the value of $A$.

(a) If there is no triangle of $X_i$ that is in $T_A$, then there are at most 6 edges of $X_i$ that are in $E_A$, one from each triangle.

Hence, the value of packing decreases by at most 7. Since the value of the packing increases by 9, we have that the value of $A'$.

(b) If there is exactly one triangle of $X_i$ that is in $T_A$, then there are at most 5 edges of $X_i$ that are in $E_A$, one from each other triangle.

Thus, the value of $A'$ decreases by at most 9, and increases by 9.

(c) If there are exactly two triangles of $X_i$ that are in $T_A$, then there are at most 4 edges of $X_i$ that are in $E_A$ (see the examples in Fig. 3(a) and (b)). Hence, we have that the value of packing $A'$ decreases by at most 9, and increases by 9.

2. $m_i = 2$.

We may assume, without loss of generality, that $x_i$ appears negated in one clause, say $c_j$, and positive in another. Then, we remove from $A'$ the triangles and edges that are in $X_i$, and add two even triangles of $X_i$ to $A'$. Furthermore, if $t^1_i s^1_j$ is in $A'$, we remove it. We next show that those changes yield a packing $A'$ whose value is at least the value of $A$.

(a) If there is no triangle of $X_i$ that is in $T_A$, then there are at most 4 edges of $X_i$ that are in $E_A$, one from each triangle.

Hence, the value of packing $A'$ decreases by at most 5. Since the value of the packing increases by 6, we have that the value of $A'$.

(b) If there is a triangle of $X_i$ that is in $T_A$, then there is only one such triangle, say $T_k$. Furthermore, there are at most 2 edges of $X_i$ that are in $E_A$, since the number of vertices in $X_i - V_{T_k}$ is 5 (see an example in Fig. 3(c)). Hence, the value of $A'$ decreases by at most 6, and increases by 6.

Finally, for each test component $C_j$, if $s^1_j$ is not already an endpoint of an edge in $E_{A'}$, then whenever possible, we add one of the edges $t^1_i s^1_j$ or $t^2_i s^1_j$ to $A'$. That is, if the corresponding clause $c_j$ has two literals, then, if $t^1_i$ is not covered by $A'$, we add $t^1_i s^1_j$ to $A'$; otherwise, if $t^2_i$ is not covered by $A'$, we add $t^2_i s^1_j$ to $A'$. If, however, the clause $c_j$ has one literal, then if $t^1_i$ is not covered by $A'$, we add $t^1_i s^1_j$ to $A'$.
An example of the construction of $A'$ (case $m_i = 3$, and $x_i$ appears negated in only one clause, say $c_i$, and positive in two other clauses). Dotted lines indicate edges of another truth component. (a) Shows a non-canonical $\{K_2, K_3\}$-packing $A$ restricted to $X_i$ and $c_i$ (highlighted edges and triangles are in $A$). In the first step, $[s^1_1, s^1_2, r^1_1]$ and $s^1_2$ are removed from $A'$, and $[r^1_1, r^1_2, 1^1, 1^2]$ are added to $A'$. In the second step, $T_1, T_2, T_3, T_4, T_5$ are added to $A'$. (b) The resulting packing $A'$.  

An example of a canonical packing $A'$ of $G'$ and a corresponding truth assignment for the variables of the SAT problem instance $\psi = (x_1 \lor x_2) \land \neg x_1 \land (x_1 \lor x_3) \land (x_2 \lor \neg x_3)$: $x_1$ and $x_3$ are set to true, $x_2$ is set to false.

An example of the construction of $A'$ is shown in Fig. 4.

Note that the resulting packing $A'$ is a canonical packing of $G'$ whose value is at least the value of $A$. We have thus proved (11).

We observe that a given canonical packing $A'$ of $G'$ corresponds to a truth assignment for the variables in $X$ in the following way. If $A'$ contains all even (resp. odd) triangles of the truth component $X_i$, then $x_i$ is set to true (resp. false). On the other hand, given a truth assignment for the variables in $X$, we can construct a canonical packing $A'$ of $G'$ in the following way. If $x_i$ is true (resp. false), we add all even (resp. odd) triangles of $X_i$ to $A'$. For each test component $c_j$ we add the triangle $[r^1_j, r^2_j, 2^j]$ to $A'$. Moreover, if the corresponding clause $c_j$ has two literals, then if $t^j_1$ is not covered by $A'$, we add $t^1_2$ to $A'$; otherwise, if $t^j_1$ is not covered by $A'$, we add $t^2_1$ to the packing. If, however, the clause $c_j$ has one literal, then if $t^j_1$ is not covered by $A'$, we add $t^1_1$ to $A'$.

Consider now a canonical packing $A'$ and the corresponding truth assignment for the variables in $X$. Let $c_j$ be a clause with two literals, and let $x_1, x_2$ be the variables which occur in $c_j$. Note that $t^j_1$ (for $i = 1, 2$) is not covered by a triangle of $A'$ that belongs to the corresponding truth component $X_i$, if and only if, $x_i$ is set to the value that satisfies $c_j$. Thus, from the construction of the canonical packing we have that the following statements are equivalent: clause $c_j$ is satisfiable; at least one of $t^1_1, t^2_1$ is not covered by a triangle of $A'$ that belongs to the corresponding truth component; exactly one of $t^1_1, t^2_1$ is in $E_{A'}$; the value of $A'$ restricted to $c_j$ is 4. Similar statements hold for a clause with one literal. Thus, the value of $A'$ restricted to $c_j$ is 4 (resp. 3), if and only if, $c_j$ is satisfiable (resp. not satisfiable). Moreover, exactly $m_i$ triangles of each $X_i$ are in $A'$. Thus, the following claim holds (see Fig. 5).

A canonical packing $A'$ of $G'$ with value $\sum_{i=1}^n 3m_i + 4k + 3(l - k)$ corresponds to a truth assignment for the variables in $X$ that satisfies exactly $k$ clauses of $\psi$, and vice versa.  

(12)
Now, given a $\{K_2, K_3\}$-packing $A$ of $G' := f(\varphi)$, we define a truth assignment $g(\varphi, A)$ in the following way. First, find a canonical packing $A'$ of $G'$ with value at least the value of $A$. Set a variable $x_i$ to true (resp. false) if $A'$ contains all even (resp. odd) triangles of the truth component $X_i$.

We next show that

$$\text{opt}_{X_3}(G') = \sum_{i=1}^{p} 3m_i + \text{opt}_{\text{Sat}}(\varphi) + 3l. \tag{13}$$

Indeed, from (12) we have that an optimal solution of $\text{Sat}(\varphi)$ corresponds to an optimal packing $A'$ of $G'$ with the value $\sum_{i=1}^{p} 3m_i + 4\text{opt}_{\text{Sat}}(\varphi) + 3(l - \text{opt}_{\text{Sat}}(\varphi))$. Thus, $\text{opt}_{X_3}(G') \geq \sum_{i=1}^{p} 3m_i + \text{opt}_{\text{Sat}}(\varphi) + 3l$. On the other hand, let $A$ be a $\{K_2, K_3\}$-packing of $G'$. If the corresponding feasible solution $g(\varphi, A)$ of $\text{Sat}(\varphi)$ satisfies $k$ clauses, we have that $k \leq \text{opt}_{\text{Sat}}(\varphi)$. Furthermore, $\text{val}_{X_3}(G', A) \leq \text{val}_{X_3}(G', A')$, and by (12), $\text{val}_{X_3}(G', A') = \sum_{i=1}^{p} 3m_i + k + 3l$. Hence, we have that $\text{opt}_{X_3}(G') \leq \sum_{i=1}^{p} 3m_i + \text{opt}_{\text{Sat}}(\varphi) + 3l$. We have thus proved (13).

Since each clause has at most 2 literals, we have $\sum_{i=1}^{p} m_i \leq 2l$. Furthermore, note that the optimal value of $\text{Sat}$ problem on $\varphi$ is at least $\frac{l}{2}$, since at least half of the clauses can be satisfied by a simple greedy approach. Thus, from (13) we have $\text{opt}_{X_3}(G') \leq 9l + \text{opt}_{\text{Sat}}(\varphi) \leq 19 \text{opt}_{\text{Sat}}(\varphi)$. Hence, taking $\alpha = 19$ we can conclude that condition (C1) of the definition of L-reduction holds.

Finally, suppose that $\text{val}_{\text{Sat}}(\varphi, g(\varphi, A)) = k$, that is, the truth assignment $g(\varphi, A)$ satisfies exactly $k$ clauses of $\varphi$. Hence, from (12) we have $\text{val}_{X_3}(G', A') = \sum_{i=1}^{p} 3m_i + k + 3l$. From this, equality (13), and the fact that $\text{val}_{X_3}(G', A') \geq \text{val}_{X_3}(G', A)$, we have

$$\text{opt}_{\text{Sat}}(\varphi) - \text{val}_{\text{Sat}}(\varphi, g(\varphi, A)) \leq \text{opt}_{X_3}(G') - \text{val}_{X_3}(G', A).$$

Thus, (C2) holds if we take $\beta = 1$. $\blacksquare$

4. Approximation algorithm for the $X_3$-packing problem

Let us denote by $C_3(\rho)$ an algorithm for the $X_3$-packing problem that consists of the algorithm Basic$_3$ together with a Procedure $P_3$ that is a $\rho$-approximation algorithm for the VTP problem. We are interested in the performance ratio of $C_3(\rho)$.

**Theorem 4.1.** Let $P_3$ be a $\rho$-approximation algorithm for the VTP problem which produces for any input graph $G$ a triangle packing that is maximal. Then the algorithm $C_3(\rho)$ is a $(1 + \frac{1}{3}\rho)$-approximation algorithm for the $X_3$-packing problem.

**Proof.** Let $G$ be a graph and $A$ the solution returned by the algorithm $C_3(\rho)$ applied to $G$. Let $\emptyset$ be an optimal solution for the $X_3$-packing problem on $G$ with the largest possible number of triangles in common with $A$. Let $l_i$ (resp. $o_i$), $0 \leq i \leq 3$, be the number of triangles of $A$ (resp. $\emptyset$) that intersect exactly $i$ vertices of $T_0$ (resp. $T_A$).

We show first that $t_0 = 0$. Suppose that $t_0 > 0$ and that $T$ is a triangle of $A$ that intersects no triangle of $\emptyset$. If at most two edges of $E_\emptyset$ are adjacent to $T$, then we can replace these edges with $T$, obtaining a $\{K_2, K_3\}$-packing with value greater than the value of $\emptyset$, a contradiction. Thus, there are 3 edges of $E_\emptyset$ adjacent to $T$. Removing these edges and adding $T$ to $\emptyset$, we get an optimal solution of the $X_3$-packing problem that has more triangles in common with $A$ than $\emptyset$ does, which is again a contradiction. Thus, $t_0 = 0$. Since $P_3$ returns a maximal triangle packing, $o_0$ must be zero.

Now, counting the vertices that are in the intersection of triangles from $A$ and $\emptyset$ we get

$$3t_3 + 2t_2 + t_1 = 3o_3 + 2o_2 + o_1. \tag{14}$$

We next define $e_1$ (resp. $e_0$) as the number of edges in $E_\emptyset$ with at least one (resp. none) of its endpoints in a triangle of $A$. Clearly, $e_1$ is at most the number of vertices $v$ of the triangles in $A$ such that $v$ is not covered by a triangle from $\emptyset$, that is,

$$e_1 \leq 2t_1 + t_2. \tag{15}$$

Let $G' := G - \{v: v$ is a vertex of a triangle in $T_A\}$. Note that a matching of $G'$ can be obtained by taking one edge of each triangle of $\emptyset$ that has exactly one vertex in common with a triangle of $A$, and taking the edges of $E_\emptyset$ that have no vertex in common with any triangle of $A$. Hence, as $E_A$ is a maximum matching of $G'$, we have $|E_A| \geq o_1 + e_0$. From this, and inequality (15), we have

$$|E_\emptyset| = e_1 + e_0 \leq 2t_1 + t_2 + |E_A| - o_1. \tag{16}$$

We now consider the ratio $r$ of the value of $\emptyset$ to the value of $A$, that is, $r := (3|T_\emptyset| + |E_\emptyset|)/(3|T_A| + |E_A|)$. Using (16) and the fact that $|T_\emptyset| = o_3 + o_2 + o_1$, we get

$$r \leq \frac{3(o_3 + o_2 + o_1) + (2t_1 + t_2 + |E_A| - o_1)}{3|T_A| + |E_A|}.$$

Since $|E_A| \geq 0$, and $r \geq 1$, we can remove $|E_A|$ in the last inequality, obtaining

$$r \leq \frac{3(o_3 + o_2 + o_1) + (2t_1 + t_2 - o_1)}{3|T_A|}.$$
Using (14), we have
\[ r \leq \frac{(3t_3 + 2t_2 + t_1) + (o_2 + 2o_1) + (2t_1 + t_2 - o_1)}{3|T_A|} = \frac{3(t_3 + t_2 + t_1) + (o_2 + o_1)}{3|T_A|} = \frac{3|T_A| + (o_2 + o_1)}{3|T_A|}. \]

Since \( o_2 + o_1 \leq |T_0| \), we have \( r \leq 1 + \frac{1}{3} \frac{|T_0|}{|T_A|} \). As \( |T_0| \leq \text{opt}_{VTP}(G) \), and \( P_3 \) is a \( \rho \)-approximation algorithm for the VTP problem,
\[ \frac{|T_0|}{|T_A|} \leq \frac{\text{opt}_{VTP}(G)}{|T_A|} \leq \rho, \quad \text{and hence, } r \leq 1 + \frac{1}{3} \rho. \]

**Corollary 4.2.** There is a \( (\frac{3}{2} + \varepsilon) \)-approximation algorithm for the \( K_3 \)-packing problem.

**Proof.** Hurkens and Schrijver [7] showed that HS\(^3\)(3, \( t \)) is a \( (\frac{3}{2} + \varepsilon) \)-approximation algorithm for the VTP problem (\( \varepsilon \) is inversely proportional to \( t \)). So it suffices to apply Theorem 4.1 with \( P_3 = \text{HS}(3, t) \) and \( \rho = \frac{3}{2} + \varepsilon \).

**Corollary 4.3.** There is a 1.4-approximation algorithm for the \( K_3 \)-packing problem on graphs with maximum degree 4.

**Proof.** It follows from Theorem 4.1 and the result of [8] showing that there is a \( \rho \)-approximation algorithm for the triangle packing problem on graphs with maximum degree 4, where \( \rho \) is slightly less than 1.2.

A more precise analysis of the greedy algorithm \( g_3 \) gives the following result.

**Theorem 4.4.** The algorithm \( g_3 \) is a 3/2-approximation for the \( K_3 \)-packing problem. Furthermore, the ratio 3/2 is tight.

**Proof.** It is similar to the proof of Theorem 2.3. Let \( \Theta \) be an optimal solution and \( B \) be the solution returned by the algorithm \( g_3 \). Let \( t_i \), \( 0 \leq i \leq 3 \), be the number of triangles of \( \Theta \) that intersect precisely \( i \) vertices of \( T_B \). Note that \( t_0 = 0 \).

Now let \( z := |\varepsilon_0| - 3|T_B| + t_1 + 2t_2 + 3t_3 \). Then at least \( \max(0, z) \) edges of \( \varepsilon_0 \) are disjoint from \( T_B \). Furthermore, every triangle of \( \Theta \) that intersects precisely 1 vertex of \( T_B \) contributes an edge that is disjoint from \( T_B \). Since \( \varepsilon_B \) is a maximum matching of \( G - \{v: v \text{ is a vertex in } T_B\} \), we have
\[ |\varepsilon_B| \geq t_1 + \max(0, z). \]  \hspace{1cm} (17)

Using the fact that \( |T_0| = t_1 + t_2 + t_3 \), we can rewrite \( z \) obtaining
\[ z = |\varepsilon_0| - 3|T_B| + 2|T_0| - t_1 + t_3. \]  \hspace{1cm} (18)

Now substituting the value of \( z \) in the inequality \( \text{val}(B) \geq 3|T_B| + t_1 + z \), we get
\[ \text{val}(B) \geq 3|T_B| + t_1 + |\varepsilon_0| - 3|T_B| + 2|T_0| - t_1 + t_3 \geq 2|T_0| + |\varepsilon_0| \geq \frac{2}{3} \text{val}(\Theta). \]

To see that the ratio 3/2 of algorithm \( g_3 \) is tight, consider the following graph \( G \): it consists of 4 triangles \( T_0, T_1, T_2, T_3 \), such that \( T_1, T_2 \) and \( T_3 \) are pairwise vertex-disjoint and each of them “hangs” in a different vertex of \( T_0 \) (\( G \) has 3 vertices of degree 4 and 6 vertices of degree 2).

5. Concluding remarks

The approximation algorithm \( C_3(\rho) \) that we presented for the \( K_3 \)-packing problem makes use of a routine to find an approximate solution for the VTP problem. From our result, it follows that any improvement on the \( (\frac{3}{2} + \varepsilon) \)-approximation ratio for the VTP problem would yield an improvement on the approximation ratio for the \( K_3 \)-packing problem.

Halldórsson [4] presented an algorithm for a version of the minimum 3-set cover problem, with the constraint that the sets found are pairwise disjoint, in addition to forming a cover of the vertices of the input graph. His algorithm is also another approach for the \( K_3 \)-packing problem. Using the results presented in [4], one can deduce that its approximation ratio is 3/2. This algorithm is however not as simple as the greedy algorithm \( g_3 \).

It would be interesting to study the \( F \)-packing problem for other families \( F \).

Acknowledgements

The first author was supported by FAPESP, Proc. 05/53840-0. The second author was supported by FAPESP, Proc. 2006/01817-7. The third author was partially supported by CNPq (Proc. 490333/04-4, 308138/04-0) and PRONEX–FAPESP/CNPq (Proc. 2003/09925-5).
References