Available online at www.sciencedirect.com
ScienceDirect
Discrete Mathematics 308 (2008) 1455–1471
DISCRETE MATHEMATICS
www.elsevier.com/locate/disc

Packing triangles in low degree graphs and indifference graphs☆

Gordana Manić*, Yoshiko Wakabayashi

Departamento de Ciência da Computação, Universidade de São Paulo, Rua do Matão, 1010—CEP 05508-090–São Paulo, Brazil

Received 5 November 2005; received in revised form 28 August 2006; accepted 11 July 2007
Available online 6 September 2007

Abstract

We consider the problems of finding the maximum number of vertex-disjoint triangles (VTP) and edge-disjoint triangles (ETP) in a simple graph. Both problems are NP-hard. The algorithm with the best approximation ratio known so far for these problems has ratio $3/2+\varepsilon$, a result that follows from a more general algorithm for set packing obtained by Hurkens and Schrijver [On the size of systems of sets every *t* of which have an SDR, with an application to the worst-case ratio of heuristics for packing problems, SIAM J. Discrete Math. 2(1) (1989) 68–72]. We present improvements on the approximation ratio for restricted cases of VTP and ETP that are known to be APX-hard: we give an approximation algorithm for VTP on graphs with maximum degree 4 with ratio slightly less than 1.2, and for ETP on graphs with maximum degree 5 with ratio 4/3. We also present an exact linear-time algorithm for VTP on the class of indifference graphs.

Keywords: Packing triangles; Approximation algorithm; Polynomial algorithm; Low degree graph; Indifference graph

1. Introduction

For a given family $\mathscr{F}$ of sets, a collection of pairwise disjoint sets of $\mathscr{F}$ is called a *packing* of $\mathscr{F}$. The *maximum k-set packing* problem, $k \geqslant 2$, is defined as follows: given a family $\mathscr{F}$ of sets of size precisely k, find a largest packing of $\mathscr{F}$. This problem is a fundamental combinatorial problem that underlies a range of practical and theoretical problems. The case $k=2$ is the well-known maximum matching problem. We study two special cases of the maximum 3-set packing problem that are NP-hard.

A cycle of length 3 in a graph $G=(V_G, E_G)$ is called a *triangle*. Let $\mathscr{T}_{\mathrm{V}}(G)$ (resp. $\mathscr{T}_{\mathrm{E}}(G)$) denote the collection of the sets of vertices (resp. edges) of all triangles in G. We address the following problems on simple graphs. *Vertex-disjoint triangle packing* (VTP): given a graph G, find a maximum size packing of $\mathscr{T}_{\mathrm{V}}(G)$, and *edge-disjoint triangle packing* (ETP): given a graph G, find a maximum size packing of $\mathscr{T}_{\mathrm{E}}(G)$. For simplicity, we may also refer to a collection of vertex-disjoint (resp. edge-disjoint) triangles of a graph G as a packing of $\mathscr{T}_{\mathrm{V}}(G)$ (resp. $\mathscr{T}_{\mathrm{E}}(G)$).

The problem VTP arises in scheduling problems and in 3-grouping problem: given a set of people and the affinities between them, divide them into groups of three members each, in the way that the persons in each group are mutually

☆ The first author was partially supported by CAPES, Brazil; the second author was partially supported by ProNEx - FAPESP/CNPq Proc. No. 2003/09925-5 and CNPq Proc. No. 308138/04-0 and 490333/04-4, Brazil.
* Corresponding author.
E-mail addresses: gocam@ime.usp.br (G. Manić), yw@ime.usp.br (Y. Wakabayashi).

0012-365X/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.disc.2007.07.100

compatible and the number of isolated persons is minimum [7]. The problem ETP has applications in computational biology [4].

As both problems are NP-hard [14,11], we wish to find good approximation algorithms or special instances amenable to polynomial algorithms. Given a parameter $\rho \geqslant 1$, a *ρ-approximation algorithm* for a maximization problem Π is a polynomial-time algorithm that, for any instance I of Π produces a solution whose value is at least $(1/\rho)\mathrm{opt}(I)$, where $\mathrm{opt}(I)$ denotes the optimal solution value for I. We also say that ρ is the *approximation ration*. A *polynomial-time approximation scheme* (PTAS) for Π is a family of algorithms $\{A_\varepsilon : \varepsilon \in (0,1)\}$ such that for each ε, A_ε is a $1/(1-\varepsilon)$-approximation algorithm for Π.

Consider the following local search algorithm $\mathrm{HS}(\mathscr{T}, t)$, where $\mathscr{T}$ is $\mathscr{T}_{\mathrm{V}}(G)$ for VTP (resp. $\mathscr{T}_{\mathrm{E}}(G)$ for ETP), and t is a positive integer.

Algorithm $\mathrm{HS}(\mathscr{T}, t)$**.** Given a collection $\mathscr{C}$ of disjoint sets constructed so far, check whether there are $p \leqslant t$ disjoint sets in $\mathscr{T} \backslash \mathscr{C}$ that intersect at most $p-1$ sets that are in $\mathscr{C}$. If this happens, swap the sets to form a larger collection $\mathscr{C}$, and repeat; otherwise, return $\mathscr{C}$ (the solution is said to be *t-optimal).*

A general result of Hurkens and Schrijver [13] on the maximum k-set packing problem implies that the above algorithm is a $(\frac{3}{2}+\varepsilon)$-approximation algorithm for both VTP and ETP (ε is inversely proportional to t). This ratio is tight and is the best approximation ratio known so far for both problems. There are only a few more results concerning maximum triangle packing problems. For the planar case, Baker [2] presented a PTAS for VTP. According to [4], this result can be extended to handle ETP as well. Both problems also admit such a scheme for λ-precision unit disk graphs [12]. The problem VTP is NP-hard when restricted to chordal graphs, while it is polynomially solvable on split graphs and cographs [9]. Recently, Hassin and Rubinstein [10] presented a randomized $(\frac{169}{89}+\varepsilon)$-approximation algorithm for VTP and also for its weighted version.

For a given integer $k \geqslant 3$, we denote by VTP-k (resp. ETP-k), the problem VTP (resp. ETP) on graphs with maximum degree k. In 2002, Caprara and Rizzi [4] proved that VTP-3 and ETP-4 can be solved in polynomial time, whereas VTP-4 (see also [3]) and ETP-5 are APX-hard (that is, they do not admit a PTAS, unless P = NP [1]). Chlebík and Chlebíková [5] showed recently that it is NP-hard to approximate VTP-4 within 95/94. We observe that the ratio $\frac{3}{2}+\varepsilon$ obtained by Hurkens and Schrijver [13] is tight even for the problem VTP-4. We present improvements on the approximation ratios of these APX-hard instances: a $(3-\frac{\sqrt{13}}{2}+\varepsilon)$-approximation algorithm for VTP-4, and a $\frac{4}{3}$-approximation algorithm for ETP-5. We also give an exact linear-time algorithm for VTP on indifference graphs (or, equivalently, unit interval graphs and proper interval graphs). This result is of interest in view of the many applications of such graphs in management, psychology, scheduling (see [8]).

1.1. Basic definitions and notation

A natural reduction for both VTP and ETP consists of deleting the edges that do not belong to any triangle. We, thus, restrict our attention to simple graphs in which every edge belongs to some triangle; these graphs will be called *irredundant*. The terminology we use is standard. One exception is that, when we write $G-U$ (for $U \subseteq V_G$ or $U \subseteq E_G$) we assume that isolated vertices and edges that do not belong to any triangle on the graph obtained by deleting U from G have been removed as well. Graphs G and H *intersect* if $G \cap H$ is a non-empty graph. The *degree* of a triangle T in a graph G, denoted by $d_G(T)$, is the number of triangles in G, different from T, that intersect T. We denote by $\mathscr{T}_G$ the collection of all triangles in G, and by $[u, v, w]$ the triangle with vertices u, v and w. If two triangles T_1 and T_2 of G have only one vertex in common and there is no other triangle in G that intersects both T_1 and T_2, we say that the subgraph $T_1 \cup T_2$ is a *butterfly* in G, and denote by $v_{T_1T_2}$ the only vertex in common to T_1 and T_2. A collection $\mathscr{T}$ of vertex-disjoint triangles in G is *locally optimal* in G if $\{V_T\colon\ T \in \mathscr{T}\}$ is a maximum packing of the family $\{V_T\colon T \in \mathscr{T}_G, T$ intersects a triangle in $\mathscr{T}\}$.

The *intersection graph* of a collection of sets $\mathscr{T}$ is the graph H with $V_H := \mathscr{T}$ and such that $TT' \in E_H \Leftrightarrow T \cap T' \neq \emptyset$. A graph G is an *indifference graph* if there exists a positive number δ and a real-valued function f on V_G such that for all $u, v \in V_G$ $(u \neq v)$, uv is an edge in G whenever $|f(u)-f(v)| < \delta$.

In all figures, each square vertex is a vertex common to two triangles in G whose union is a butterfly. A vertex x that is marked with a circle is *saturated*, that is, no more edges can be incident to x.

2. Algorithm for VTP on graphs with maximum degree 4

In this section we restrict our attention to graphs with maximum degree 4 and describe an approximation algorithm, called VT4$_k$, for VTP on such graphs. This algorithm performs some approximation-preserving reductions to transform the input graph G into another graph G' in which every triangle intersects at most 3 other triangles. Then, on the intersection graph of $\mathscr{T}_{G'}$ it applies the $(3 - \frac{\sqrt{13}}{2} + \frac{13-\sqrt{13}}{52k})$-approximation algorithm of Chlebík and Chlebíková [6], which we denote by MIS3$_k$ (where k is a fixed integer parameter), for the problem of finding a maximum cardinality independent set of vertices on graphs with maximum degree 3. For $k = 4$ the above ratio is slightly less than 1.25; and for $k > 65$ it is slightly less than 1.2. We note that MIS3$_k$ follows essentially the algorithm of Berman and Fujito [3], but the more detailed analysis done by Chlebík and Chlebíková in [6] improves the ratio $(\frac{6}{5} + \frac{1}{5k})$ that was obtained in [3].

A rough sketch of the algorithm VT4$_k$ is as follows. In each iteration, we repeatedly add a set $\mathscr{T} \subseteq \mathscr{T}_G$, $|\mathscr{T}| \leqslant 2$, locally optimal in G to $\mathscr{A}^*$ (the set to be returned by the algorithm) and update G. If G still contains a triangle T with degree greater than 3, the algorithm finds a certain subgraph H that contains T, and applies an appropriate reduction on H in such a way that in the reduced graph the triangles obtained by this reduction have degree at most 3. The notion of butterfly is crucial, because the reduction is based on the number of triangles in H that forms a butterfly with a triangle not in H (which is, as we will prove in Section 2.1, at most 2). We will also prove that for any collection $\mathscr{T}$ locally optimal in G, adding $\mathscr{T}$ to the current solution of our algorithm and deleting from G the vertices of all triangles in $\mathscr{T}$ preserves the approximation ratio of the algorithm VT4$_k$. Furthermore, we will prove that every reduction made on a subgraph H preserves the approximation ratio of the algorithm.

We now give some more details of the algorithm VT4$_k$. In each iteration of the algorithm VT4$_k$, we repeatedly add a set $\mathscr{T} \subseteq \mathscr{T}_G$, $|\mathscr{T}| \leqslant 2$, locally optimal in G to $\mathscr{A}^*$ (and update G) in order to eliminate special instances, that is, instances that have a locally optimal collection with at most two triangles (some of those special instances are shown in Fig. 1). By doing this, as we will see later, we are left with only a few general instances that have similar structure. If G still contains a triangle T with degree greater than 3, the algorithm searches for the subgraph H, defined as a maximal connected irredundant subgraph of G that contains T and does not contain any butterfly. Exploring the structural properties of the graph G (irredundancy and degree boundedness), and using the fact that there is no subcollection $|\mathscr{T}|$, $|\mathscr{T}| \leqslant 2$, locally optimal in G (otherwise, it would be added to the solution in the first step), we are able to prove that such a graph H has a very specific structure. More precisely, the number of triangles T' in $\mathscr{T}_H$ for which there exists a triangle in $\mathscr{T}_G \backslash \mathscr{T}_H$ that forms a butterfly with T' in G is at most 2. Furthermore, H is isomorphic to one of the graphs in Fig. 15. It should be noted that our definition of the subgraph H allows us to find it without having to exhaustively search for subgraphs isomorphic to those of Fig. 15.

We now explain the behaviour of the algorithm in each of the possible cases for the subgraph H. If the subgraph H has two triangles T' and T'' that form a butterfly with a triangle not in H, then as we will prove in Section 2.1, H

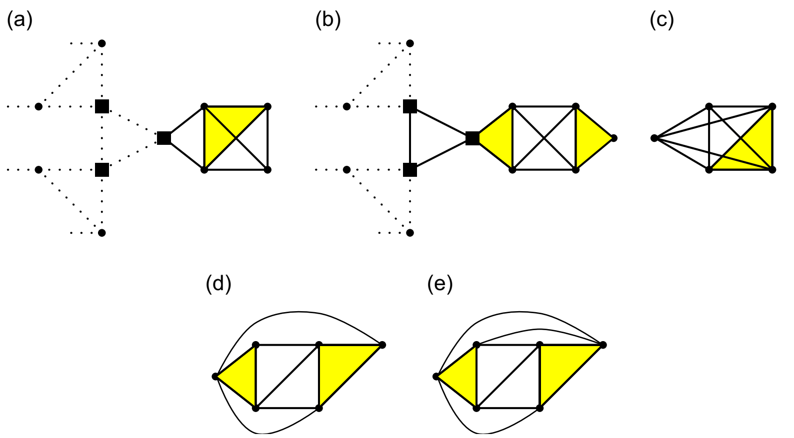

Fig. 1. Examples of instances that have a locally optimal collection $\mathscr{T}$ with at most two triangles (the marked triangles are in $\mathscr{T}$, and each square vertex is a vertex common to two triangles in G whose union is a butterfly). Graphs (c)–(e) are components of G.

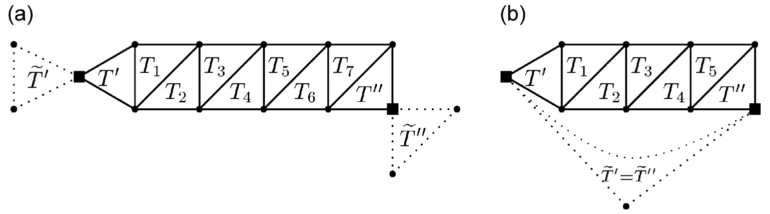

Fig. 2. Examples of a graph H obtained in the algorithm VT4$_k$ (H is the graph with full lines). T' (resp. T'') is a triangle of H that forms a butterfly with a triangle $\widetilde{T}'$ (resp. $\widetilde{T}''$) not in H. Note that possibly $\widetilde{T}' = \widetilde{T}''$. (a) Maximum packing $\{T_1, T_4, T_7\}$ of $\mathscr{T}_{\mathrm{V}}(H)$ is locally optimal in G. (b) Maximum packing $\{T', T_3, T''\}$ of $\mathscr{T}_{\mathrm{V}}(H)$ is locally optimal in G.

is isomorphic to one of the graphs (a), (b) or (c) of Fig. 15. In all those cases the algorithm performs an appropriate reduction. Making use of the specific structure of the graph H, the algorithm is able to decide whether there is a collection $\mathscr{T} \subseteq \mathscr{T}_H$, $|\mathscr{T}| > 2$, locally optimal in G. If such a collection exists, the reduction adds $\mathscr{T}$ to $\mathscr{A}^*$ and updates G. Otherwise, the algorithm reduces H in such a way that in the reduced graph the triangles obtained by the reduction have degree at most 3. The algorithm decides which of these reductions to apply by comparing the cardinality of the maximum packings of the $\mathscr{T}_{\mathrm{V}}(H)$ in the cases when both T' and T'' are in the packing, when only one of them is in the packing, and when neither of them is in the packing. In order to give an intuition of the reductions, we present some examples, shown on Figs. 2 and 3.

Note that for the example in Fig. 2(a), maximum packings of $\mathscr{T}_{\mathrm{V}}(H)$ in all possible cases (when both T', T'' are in the packing, when only one of them is in the packing, and when neither of them is in the packing) have the same cardinality. Observe, furthermore, that these packings are easy to be found. For example, a maximum packing of $\mathscr{T}_{\mathrm{V}}(H)$ in the case T' is in the packing and T'' is not in the packing can be obtained by taking T', the triangle T_3 (which is locally optimal in $H - v_{T''\widetilde{T}''} - V_{T'}$), and the triangle T_6 (which is locally optimal in $H - v_{T''\widetilde{T}''} - V_{T'} - V_{T_3}$). We will prove that in this case a maximum packing of $\mathscr{T}_{\mathrm{V}}(H)$ which contains neither T' nor T'' is locally optimal in G (in the example, this collection is $\{T_1, T_4, T_7\}$). Fig. 2(b) shows a case when $\widetilde{T}' = \widetilde{T}''$, and maximum packings of $\mathscr{T}_{\mathrm{V}}(H)$ when bothT', T'' are in the packing, when only one of them is in the packing, and when neither of them is in the packing are not all of the same cardinality. We will prove that in this case a maximum packing of $\mathscr{T}_{\mathrm{V}}(H)$ which contains both T' and T'' is locally optimal in G (in the example, this collection is $\{T', T_3, T''\}$).

We describe now the algorithm VT4$_k$, but the reader may find useful to read first some comments on Fig. 3 given later. As for the example in Fig. 3, comparing the maximum packings of $\mathscr{T}_{\mathrm{V}}(H)$ in all possible cases (when both T', T'' are in the packing, when only one of them is in the packing, and when neither of them is in the packing), we note the following. Even though the packings in (c) and (e) are of the same size, the solution in (e) is a "better solution" than the solution in (c) (because if we choose $\{T', T_5\}$ to be included in the final solution, we must omit the triangle $\widetilde{T}'$, but this is not the case if we choose $\{T_1, T_4\}$ to be included in the final solution of the algorithm). Similarly, even though the packings in (d) and (e) are of the same size, choosing $\{T_1, T_4\}$ is better than choosing $\{T_1, T''\}$ to be included in the final solution. Hence, it is always good to choose $\{T', T_3, T''\}$ or $\{T_1, T_4\}$ (and include it in the final solution of the algorithm). Yet, we do not know which one is better globally. Thus, we apply the Reduce1(H), that is, we replace all the triangles of H, except T' and T'', with a new triangle T_H. We apply the reduction in order to eliminate the triangles of degree greater than 3, that is, in order to be able to apply the algorithm MIS3$_k$. As we will show later, this reduction preserves the approximation ratio of the algorithm.

If the subgraph H has only one triangle T' such that exists a triangle in $\mathscr{T}_G \backslash \mathscr{T}_H$ that forms a butterfly with T' in G then, as we will prove in Section 2.1, H is isomorphic to the graph in Fig. 15(d). Furthermore, $G[V_H]$ is a component of G. Observe that for the graph $G[V_H]$ it is easy to find an optimal packing: take T', T_3, T_6, T_9, etc. Note that T' is locally optimal in H, T_3 is locally optimal in $H - V_{T'}$, T_6 is locally optimal in $H - V_{T'} - V_{T_3}$, etc.

If, however, H has no triangle T' that forms a butterfly with a triangle in $\mathscr{T}_G \backslash \mathscr{T}_H$, then H is isomorphic to the graph in Fig. 15(e). Moreover, every vertex has degree 4, and thus, $G[V_H]$ is a component of G. Note that if $\widetilde{T}$ is any triangle in H, then $H - V_{\widetilde{T}}$ is isomorphic to the graph with full lines shown in Fig. 15(d). We will show, thus, that an optimal solution of the graph H isomorphic to the graph in Fig. 15(e) can be obtained by taking any triangle $\widetilde{T}$ and an optimal solution of $H - V_{\widetilde{T}}$. An optimal solution of $H - V_{\widetilde{T}}$ can be constructed as for the graph in Fig. 15(d).

We repeat the iterations while there exists a triangle in G with degree greater than 3. When there is no more such triangles, we apply the algorithm MIS3$_k$ on the intersection graph of $\mathscr{T}_G$. Finally, for every application (in the reverse order) of Reduce(H) we do Restore(H), that is, we add the appropriate triangles of H to the final solution of the algorithm (see Fig. 3).

Algorithm VT4$_k$.
Input: An irredundant graph G with maximum degree 4.

1 $\mathscr{A}^* \leftarrow \emptyset$
2 **while** there exists a triangle in G with degree greater than 3
3   **do while** there exists $\mathscr{T} \subseteq \mathscr{T}_G$, $|\mathscr{T}| \leqslant 2$, locally optimal in G
4   **do** Accept($\mathscr{T}$)
5   **if** there exists a triangle $T \in \mathscr{T}_G$ with $d_G(T) > 3$
6   **then** $H \leftarrow$ maximal connected irredundant subgraph of G that
7   contains T and does not contain any butterfly
8         $B_H \leftarrow \{T' \in \mathscr{T}_H :$ exists a triangle in $\mathscr{T}_G \backslash \mathscr{T}_H$ that forms
9   a butterfly with T' in $G\}$
10   **switch** ($|B_H|$){
11   **case 2**: Reduce(H)
12   **case 1**: $\mathrm{Sol}_H \leftarrow$ Commit(H); $\mathscr{A}^* \leftarrow \mathscr{A}^* \cup \mathrm{Sol}_H$
13   **case 0**: take a triangle $\widetilde{T}$ in $\mathscr{T}_H$;
14              $\mathrm{Sol}_H \leftarrow \{\widetilde{T}\} \cup \mathrm{Commit}(H - V_{\widetilde{T}})$; $\mathscr{A}^* \leftarrow \mathscr{A}^* \cup \mathrm{Sol}_H\}$
15 **if** $G \neq \emptyset$ **then** $\mathscr{A}^* \leftarrow \mathscr{A}^* \cup \mathrm{MIS3}_k$(intersection graph of $\mathscr{T}_G$)
16 **for** every application (in the reverse order) of Reduce(H) **do** Restore(H)
17 **return** $\mathscr{A}^*$.

Each of the procedures of algorithm VT4$_k$ is described next in more detail.

(1) Accept($\mathscr{T}$): Add $\mathscr{T}$ to $\mathscr{A}^*$ and delete from G the vertices of all triangles in $\mathscr{T}$. The algorithm executes Accept($\mathscr{T}$) not only in line 4, but also in the Reduce(H). As we will show later, the algorithm executes Accept($\mathscr{T}$) only if the collection $\mathscr{T}$ is locally optimal.
(2) Commit(H): Set $\mathscr{E} := \emptyset$. While $H \neq \emptyset$, find a triangle T locally optimal in H, add T to $\mathscr{E}$ and delete V_T from H. Return $\mathscr{E}$.
(3) Reduce(H): Given a graph H as defined in the algorithm, with $|B_H| = 2$, this procedure reduces H in such a way that in the reduced graph the triangles obtained by the reduction have degree at most 3 (in the case of Reduce1(H) and Reduce2(H)), or adds some triangles of H to $\mathscr{A}^*$ and updates the graph (that is, executes Accept($\mathscr{T}$), for a $\mathscr{T} \subseteq \mathscr{T}_H$). More formally, the procedure Reduce(H) is as follows.
Take $T', T'' \in B_H$ and $\widetilde{T}', \widetilde{T}'' \in \mathscr{T}_G \backslash \mathscr{T}_H$ such that $T' \cup \widetilde{T}'$ and $T'' \cup \widetilde{T}''$ are butterflies in G (possibly $\widetilde{T}' = \widetilde{T}''$). Let

$$\mathrm{Sol}_{T'T''} := \{T', T''\} \cup \mathrm{Commit}(H - V_{T'} - V_{T''}),$$

$$\mathrm{Sol}_{T'\overline{T''}} := \{T'\} \cup \mathrm{Commit}(H - V_{T'} - v_{T''\widetilde{T}''}),$$

$$\mathrm{Sol}_{\overline{T'}T''} := \{T''\} \cup \mathrm{Commit}(H - V_{T''} - v_{T'\widetilde{T}'}),$$

$$\mathrm{Sol}_{\overline{T'}\,\overline{T''}} := \mathrm{Commit}(H - v_{T'\widetilde{T}'} - v_{T''\widetilde{T}''}).$$

As we will see in Section 2.1, $\mathrm{Sol}_{T'T''}$ is a maximum packing of $\mathscr{T}_V(H)$ that contains T' and T'', $\mathrm{Sol}_{T'\overline{T''}}$ is a maximum packing of $\mathscr{T}_V(H)$ that contains T' but not T'', etc.
 (a) If $|\mathrm{Sol}_{T'T''}| = |\mathrm{Sol}_{T'\overline{T''}}| = |\mathrm{Sol}_{\overline{T'}T''}| = |\mathrm{Sol}_{\overline{T'}\,\overline{T''}}|$ then Accept ($\mathrm{Sol}_{\overline{T'}\,\overline{T''}}$), see Fig. 2(a).
 (b) If the equality in (a) is not satisfied and $\widetilde{T}' = \widetilde{T}''$ then Accept ($\mathrm{Sol}_{T'T''}$), see Fig. 2(b).
 (c) If $|\mathrm{Sol}_{T'T''}| - 1 = |\mathrm{Sol}_{T'\overline{T''}}| = |\mathrm{Sol}_{\overline{T'}T''}| = |\mathrm{Sol}_{\overline{T'}\,\overline{T''}}|$ and $\widetilde{T}' \neq \widetilde{T}''$ then apply Reduce1(H):

$$G \leftarrow (G - (E_H \backslash \{E_{T'} \cup E_{T''}\})) \cup T_H,$$

 where $T_H := [v', w, v'']$, w is a new vertex, v' is any vertex of T' different from $v_{T'\widetilde{T}'}$, and v'' is any vertex of T'' different from $v_{T''\widetilde{T}''}$. Thus, Reduce1(H) replaces all triangles of H, except T' and T'', with a new

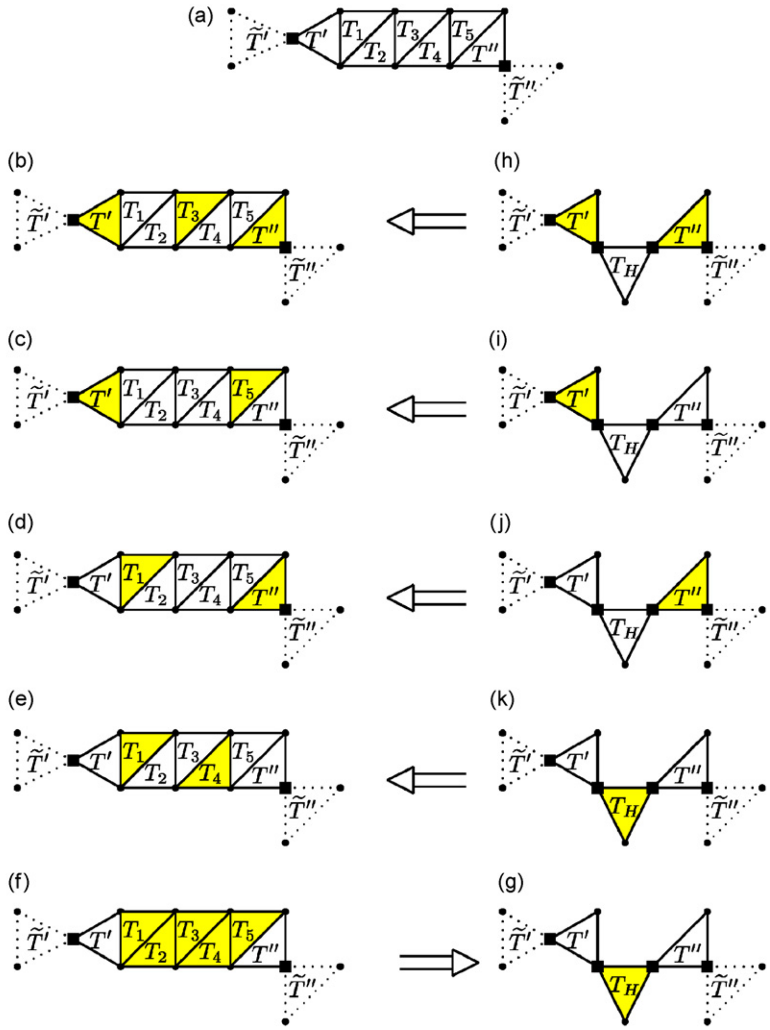

Fig. 3. (a) An example of a graph H obtained in the algorithm $\mathrm{VT4}_k$ (H is the graph with the full lines). $T' \cup \widetilde{T}'$ and $T'' \cup \widetilde{T}''$ are butterflies in G. Depending on whether T', T'' are restricted to be in a maximum packing of $\mathscr{T}_{\mathrm{V}}(H)$ or not, we have one of the cases (b)–(e). (b) $\{T', T_3, T''\}$ is a maximum packing of $\mathscr{T}_{\mathrm{V}}(H)$ that contains T' and T''. (c) $\{T', T_5\}$ is a maximum packing of $\mathscr{T}_{\mathrm{V}}(H)$ that contains T' but not T''. (d) $\{T_1, T''\}$ is a maximum packing of $\mathscr{T}_{\mathrm{V}}(H)$ that contains T'' but not T'. (e) $\{T_1, T_4\}$ is a maximum packing of $\mathscr{T}_{\mathrm{V}}(H)$ that contains neither T' nor T''. (f) and (g) represent $\mathrm{Reduce1}(H)$: it replaces all the triangles of H, except T' and T'', with a new triangle T_H. (h) If T', T'' are in the solution of the algorithm before applying $\mathrm{Restore}(H)$, then the procedure $\mathrm{Restore}(H)$ adds T_3 to the final solution. (i) If T' is in the solution of the algorithm before applying the $\mathrm{Restore}(H)$, but not T'', then this procedure adds T_5 to the final solution. (j) Similar to case (i). (k) If T_H is in the solution of the algorithm before applying the $\mathrm{Restore}(H)$, then the procedure $\mathrm{Restore}(H)$ removes T_H from the solution and adds T_1 and T_4 to it.

triangle T_H that induces a butterfly with T' and with T'' in the reduced graph (see Fig. 3 (f) and (g)). We will prove that T' and T'' are disjoint, that is, $\mathrm{Reduce1}(H)$ is well defined.

(d) If $|\mathrm{Sol}_{T'T''}| = |\mathrm{Sol}_{T'\overline{T}''}| = |\mathrm{Sol}_{\overline{T}'T''}| = |\mathrm{Sol}_{\overline{T}'\overline{T}''}| + 1$ and $\widetilde{T}' \neq \widetilde{T}''$, then apply $\mathrm{Reduce2}(H)$:

$$G \leftarrow (G - E_H) \cup T_H^1 \cup T_H^2,$$

where $T_H^1 := [v_{T'\widetilde{T}'}, w_1, w]$, $T_H^2 := [w, w_2, v_{T''\widetilde{T}''}]$ and w_1, w, w_2 are new vertices. Hence, this reduction replaces all triangles of H with the new triangles T_H^1 and T_H^2, such that T_H^1 induces a butterfly with T_H^2 and with $\widetilde{T}'$; and T_H^2 induces a butterfly with $\widetilde{T}''$ and with T_H^1 on the reduced graph.

(4) $\mathrm{Restore}(H)$: If the reduction applied to H was $\mathrm{Reduce1}(H)$ or $\mathrm{Reduce2}(H)$, then the procedure $\mathrm{Restore}(H)$ adds appropriate triangles of H to $\mathscr{A}^*$.

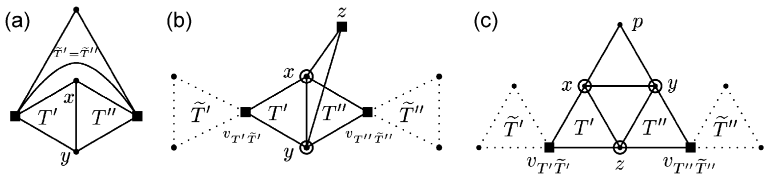

Fig. 4. Triangles in B_H cannot intersect.

(a) If the reduction applied to H was Reduce1(H), then if T_H belongs to $\mathscr{A}^*$ before applying Restore(H), this procedure removes T_H from $\mathscr{A}^*$ and adds to it the set $\mathrm{Sol}_{\overline{T'}\,\overline{T''}}$ (computed in Reduce(H)); if $T', T'' \in \mathscr{A}^*$, then $\mathscr{A}^* \leftarrow \mathscr{A}^* \cup \mathrm{Sol}_{T'T''}$; if $T' \in \mathscr{A}^*$, $T'' \notin \mathscr{A}^*$, then $\mathscr{A}^* \leftarrow \mathscr{A}^* \cup \mathrm{Sol}_{T'\overline{T''}}$; and if $T' \notin \mathscr{A}^*$, $T'' \in \mathscr{A}^*$, then $\mathscr{A}^* \leftarrow \mathscr{A}^* \cup \mathrm{Sol}_{\overline{T'}T''}$ (see Fig. 3).

(b) If, however, the reduction applied to H was Reduce2(H), then if T_H^1 belongs to $\mathscr{A}^*$ before applying Restore(H), this procedure adds $\mathrm{Sol}_{T'\overline{T''}}$ to $\mathscr{A}^*$ and removes T_H^1; if $T_H^2 \in \mathscr{A}^*$, then adds $\mathrm{Sol}_{\overline{T'}T''}$ to $\mathscr{A}^*$ and removes T_H^2; and if $T_H^1, T_H^2 \notin \mathscr{A}^*$, then adds $\mathrm{Sol}_{\overline{T'}\,\overline{T''}}$ to $\mathscr{A}^*$.

2.1. *Analysis and the approximation ratio of* VT4$_k$

We first observe the following fact.

If H is a butterfly-free graph, and T_1, T_2 are triangles of H that have only one vertex z in common, then there is at least one edge of H with one endpoint in $V_{T_1}\backslash\{z\}$ and another in $V_{T_2}\backslash\{z\}$. (1)

This is because the degree bound 4 has already been met at z.

Lemma 1. *For each iteration of the algorithm* VT4$_k$ *for which the condition in line* 5 *is satisfied, if* $|B_H| = 2$, *then the triangles in* B_H *are disjoint.*

Proof. Let G be the graph in line 5 of the algorithm VT4$_k$, H the subgraph of G (as defined in line 6) with $|B_H| = 2$, and $T', T'', \widetilde{T}', \widetilde{T}''$ as defined in Reduce(H).

Suppose that $|V_{T'} \cap V_{T''}| = 2$, say $V_{T'} \cap V_{T''} = \{x, y\}$. If $\widetilde{T}' = \widetilde{T}''$, then $T' \cup \widetilde{T}'$ is not a butterfly in G, a contradiction (see Fig. 4(a)). Thus, $\widetilde{T}' \neq \widetilde{T}''$. From the facts that $v_{T'\widetilde{T}'}$ and $v_{T''\widetilde{T}''}$ are already saturated, $T' \cup \widetilde{T}'$ (resp. $T'' \cup \widetilde{T}''$)is a butterfly in G, and $\Delta(G) = 4$, we conclude that there is at most one more triangle in G that intersects x or y, namely the triangle $[x, y, z]$. Note that now also both x and y are saturated. Since H is butterfly-free, there is no triangle in H other than $[x, y, z]$ which contains z (note that z can be a vertex in the intersection of two triangles that form a butterfly). We have thus showed that H contains T', T'' and at most one more triangle $[x, y, z]$ (see Fig. 4(b)). But this implies that H does not have a triangle T with $d_G(T) > 3$, a contradiction.

Suppose now that T' and T'' have only one vertex in common. Similarly as above, we have $\widetilde{T}' \neq \widetilde{T}''$. If $T' = \{v_{T'\widetilde{T}'}, x, z\}$ and $T'' = \{v_{T''\widetilde{T}''}, z, y\}$, since $v_{T'\widetilde{T}'}$ and $v_{T''\widetilde{T}''}$ are already saturated, using the fact (1), we have that $xy \in E_H$. Now, from the facts that H has a triangle T with $d_G(T) > 3$, z is saturated and $\Delta(G) = 4$, we conclude that there is at most one triangle in G that intersects x or y, namely the triangle $[x, y, p]$ (see Fig. 4(c)). But now, both x and y are saturated and $[x, y, z]$ is locally optimal in G. Thus, the algorithm would have applied Accept($[x, y, z]$) in line 4, and we again obtain a contradiction. □

Corollary 2. Reduce1(H) *is well defined and it does not create new triangles in G, except for* T_H.

Proof. Follows directly from Lemma 1. □

We observe, furthermore, that Reduce2(H) does not create any new triangles in G except for T_H^1 and T_H^2, because $\widetilde{T}' \neq \widetilde{T}''$.

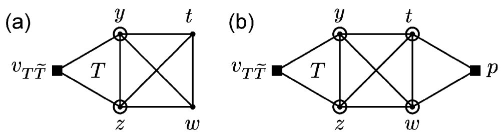

Fig. 5. Cases when $d_H(T) = 4$.

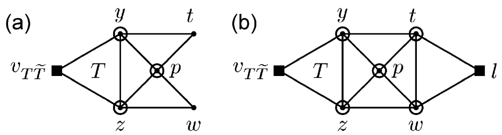

Fig. 6. Cases when $d_H(T) = 3$.

It is easy to see that after the execution of Reduce1(H) or Reduce2(H), the graph obtained still has maximum degree 4, and is irredundant. We will show that all other reductions applied in the algorithm consist of adding a collection $\mathscr{T}$ locally optimal in G to $\mathscr{A}^*$, and deleting from G the vertices of all triangles in $\mathscr{T}$. Thus, the structural properties of the input graph (maximum degree 4 and irredundancy) are maintained in each iteration.

Moreover, we observe that

$$\begin{aligned}&\text{For any iteration of the algorithm for which there exists a triangle}\\&\text{in } G \text{ with degree greater than 3, where } G \text{ is the graph in line 5,}\\&\text{there is no subcollection } \mathscr{T},\ |\mathscr{T}| \leqslant 2, \text{ locally optimal in } G.\end{aligned} \tag{2}$$

Otherwise, of course, the algorithm would have applied Accept$(\mathscr{T})$ in line 4.

Lemma 3. *For each iteration of the algorithm* VT4$_k$, *we have that* $d_G(x) \geqslant 3$ *for all* $x \in V_H$, *where H is the graph in line* 6.

Proof. Suppose that H has a vertex x with $d_G(x) = 2$, and let T be the triangle in H that contains x. If just one triangle of G is adjacent to T, then T would be locally optimal in G, a contradiction with (2). Thus, $d_G(T) \geqslant 2$. If there are two vertex-disjoint triangles T_1 and T_2 that intersect T, then $T \cup T_1$ and $T \cup T_2$ are butterflies in G, that is, $H = T$, which is impossible (since H has a triangle with degree in G greater than 3). Thus, the triangles that have a vertex in common with T intersect pairwise, that is, T is locally optimal in G, which contradicts (2). $\square$

Theorem 4. *For each iteration of the algorithm* VT4$_k$, *we have that* $|B_H| \leqslant 2$.

Proof. Suppose that $B_H \neq \emptyset$, and let T be any triangle in B_H. By the definition of B_H, triangle T forms a butterfly with a triangle $\widetilde{T} \notin \mathscr{T}_H$.

If $d_H(T) = 4$, then using the facts that $\Delta(G) = 4$ and $T \cup \widetilde{T}$ is a butterfly in G, we conclude that the subgraph of H induced by the triangle T and the triangles of H that intersect T is isomorphic to the graph in Fig. 5(a). Note that y and z are saturated. Furthermore, since $\Delta(G) = 4$, the vertex t (resp. w) of the graph shown in Fig. 5(a) is not a vertex in the intersection of two triangles that form a butterfly in G. Hence, if H is isomorphic to the graph in Fig. 5(a), then $[y, t, w]$ would be locally optimal in G, which contradicts (2). Thus, since $v_{T\widetilde{T}}$, y and z are already saturated, there is a vertex p adjacent to t and w. Note that now t and w are also saturated. From Lemma 3 it follows that $d_G(p) \geqslant 3$, that is, the triangle $[t, p, w]$ is also in B_H, and thus, $|B_H| = 2$. Observe that H is isomorphic to the graph in Fig. 5(b).

If, however, $d_H(T) = 3$, then since $\Delta(G) = 4$ and $T \cup \widetilde{T}$ is a butterfly in G, the subgraph of H induced by the triangle T and the triangles of H that intersect T is isomorphic to the graph in Fig. 6(a) (note that y, p and z are saturated). If H is isomorphic to the graph in Fig. 6(a), then $[y, p, z]$ is locally optimal in G, a contradiction. Thus, $tw \in E_H$. If $d_G(t) = d_G(w) = 3$, then $[t, p, w]$ is locally optimal in G, again a contradiction. Hence, there is a vertex l adjacent to t and w. Note that now t and w are also saturated. From Lemma 3 it follows that $d_G(l) \geqslant 3$, that is, the triangle $[t, l, w]$ also belongs to B_H, and thus $|B_H| = 2$. Observe that H is isomorphic to the graph in Fig. 6(b).

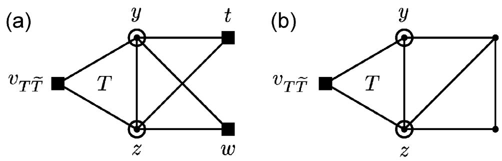

Fig. 7. If $d_H(T) = 2$, the subgraph of H induced by T and the triangles of H that intersect T is not isomorphic to the graph in (a), but to the graph in (b). In (b), the vertex y is saturated because $d_H(T) = 2$.

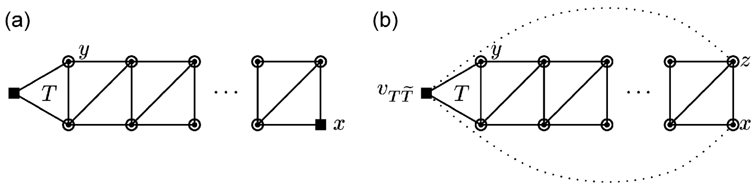

Fig. 8. Possible configurations when $d_H(T) = 2$. In (a), $|B_H| = 2$, and the graph H has at least seven vertices. In (b), full lines indicate edges in H. The graph (b) has at least nine vertices, $G[V_H]$ is a component of G, xy is not an edge of G, and $|B_H| = 1$.

If $d_H(T) = 2$, and the subgraph of H induced by T and the triangles of H that intersect T is isomorphic to the graph in Fig. 7(a), then clearly, $tw \notin E_H$. Furthermore, $T \cup \widetilde{T}$ is a butterfly in G, and $v_{T\widetilde{T}}$, z, y are saturated. Hence, from Lemma 3 we have that $[y, t, z]$ (resp. $[y, w, z]$) forms a butterfly with a triangle not in H, with the vertex t (resp. w) being in the intersection of the triangles that form such butterfly. Thus, from the definition of H, it follows that in this case H is isomorphic to the graph in Fig. 7(a). But then, H does not have a triangle with degree greater than 3 in G, a contradiction. Hence, the subgraph of H induced by T and the triangles of H that intersect T is isomorphic to the graph in Fig. 7(b).

Now, since $T \cup \widetilde{T}$ is a butterfly in G, $\Delta(G) = 4$, G is irredundant and H is butterfly-free, we have the following two possibilities.

- H is isomorphic to the graph in Fig. 8(a). Note that H can have odd or even number of triangles (the configuration of the graph H with even number of triangles is similar to that of Fig. 8(a)). Since H has a triangle with the degree in G greater than 3, this graph H has at least seven vertices. Furthermore, by Lemma 3, $d_G(x) \geqslant 3$, that is, $|B_H| = 2$ for that graph.
- H is isomorphic to the graph in Fig. 8(b) (full lines indicate edges in H). Note that H can have odd or even number of triangles (the configuration of the graph H with even number of triangles is similar to that of Fig. 8(b)). Observe, furthermore, that $xy \notin E_G$ and T forms a butterfly with the triangle $[v_{T\widetilde{T}}, z, x]$. We next show that

$$G[V_H] \text{ is a component of } G. \tag{3}$$

Suppose that (3) does not hold. Since all the vertices of graph in Fig. 8(b) have degree 4 in G, except for x and y, then there is a vertex $w \notin V_H$ such that, without loss of generality, $xw \in E_G$. From degree boundedness and the fact that $xy \notin E_G$, we conclude that xw is not an edge of any triangle, a contradiction. Hence, the statement (3) holds, and we have that $|V_H| \geqslant 9$ (otherwise, H would have $\mathscr{T}$, $|\mathscr{T}| \leqslant 2$ locally optimal in G). Note, furthermore, that $|B_H| = 1$.

We proved, thus, that $|B_H| \leqslant 2$ in all possible cases. □

Lemma 5. *For each iteration of the algorithm* VT4$_k$, *if* $B_H = \emptyset$, *then* $d_H(x) \geqslant 3$ *for all* $x \in V_H$, *and* H *is a component of* G.

Proof. Suppose that there exists $x \in V_H$ with $d_H(x) = 2$. Then Lemma 3 implies that there is an edge xw not in E_H. From the irredundancy of G, we have that xw is the edge of a triangle $\widetilde{T} \in \mathscr{T}_G \backslash \mathscr{T}_H$. Since $B_H = \emptyset$, then $\widetilde{T}$ does not induce a butterfly with a triangle of H. Furthermore, x is in V_H and thus, by the definition of H we have that $\widetilde{T} \in \mathscr{T}_H$, a contradiction. Similarly we have that H is a component of G. □

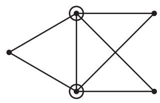

Fig. 9. Case where every pair of triangles in H has two vertices in common is not possible.

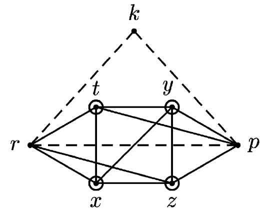

Fig. 10. Graph H when rz, pt is in E_H, and pr is not in E_H. Dashed lines can not be edges of H.

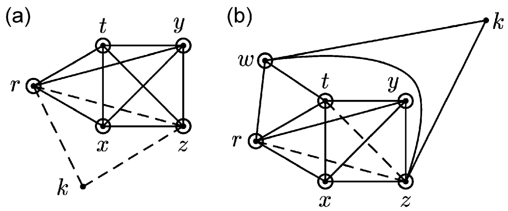

Fig. 11. Cases when rz is not in E_H, and ry is in E_H.

Theorem 6. *For each iteration of the algorithm* $\mathrm{VT4}_k$, *if* $|B_H| \leqslant 1$, *then* $G[V_H]$ *is a component of* G *and* Sol_H *is an optimal solution in that component* (*where* Sol_H *is the set as defined in lines* 12 *and* 14 *of the algorithm* $\mathrm{VT4}_k$).

Proof. As we saw in the proof of Theorem 4, if $|B_H| = 1$, then H is isomorphic to the graph in Fig. 8(b) (with the full lines being the edges of H), and $G[V_H]$ is a component of G. Note that in this case $\mathrm{Commit}(H)$ is a packing of $\mathscr{T}_{\mathrm{V}}(H)$ that covers all the vertices of H, except at most two of them, and thus, $\mathrm{Commit}(H)$ is an optimal solution of the component $G[V_H]$.

Suppose now that $B_H = \emptyset$. From Lemma 5, it follows that H is a component of G, and thus, $|V_H| \geqslant 9$ (for otherwise, G would have $\mathscr{T}$, $|\mathscr{T}| \leqslant 2$ locally optimal).

If every pair of triangles in H has two vertices in common, then H is isomorphic to the graph in Fig. 9. Hence, H does not have a triangle T with $d_G(T) > 3$, a contradiction.

Let then $T_1 = [x, y, z]$ and $T_2 = [x, t, r]$ be triangles of H with only one vertex in common. Since H does not have a butterfly, using the fact (1), we may assume, without loss of generality, that $yt \in E_H$. Now we have the following possibilities.

(i) $rz \in E_H$. Since $|V_H| > 5$, using the degree boundedness and irredundancy, we have that there exists a vertex p adjacent to, without loss of generality, y and z. From Lemma 5, we have that $d_H(p) \geqslant 3$. Thus, as H is butterfly-free and irredundant, and x, y, z are already saturated, it follows that $pt \in E_H$ or $pr \in E_H$. If both pt and pr are edges of H, then all vertices of H are saturated, and $|V_H| \leqslant 8$, a contradiction with the fact that $|V_H| \geqslant 9$. If, however, $pt \in E_H$, $pr \notin E_H$ (see Fig. 10) then there is no new vertex k adjacent to r or p. Indeed, suppose that there is a new vertex k such that, without loss of generality, $rk \in E_G$. From the degree boundedness and the fact that $pr \notin E_H$, it follows that rk is not an edge of any triangle, a contradiction. Thus again, $|V_H| \leqslant 8$, a contradiction. Similarly if $pt \notin E_H$, $pr \in E_H$.

(ii) $rz \notin E_H$, $ry \in E_H$. If $tz \in E_H$, then of course, x, y, t are saturated. Since $rz \notin E_H$, similarly as above, we conclude that there is no vertex k adjacent to r or z. Thus, H is isomorphic to the graph in Fig. 11(a) and $|V_H| \leqslant 8$,

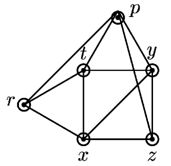

Fig. 12. Case when rz, ry, tz are not edges of H and there exists a vertex p adjacent to t and y in H.

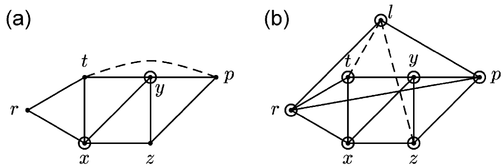

Fig. 13. Cases when rz, ry, tz are not edges of H and there is no vertex adjacent to t and y in H.

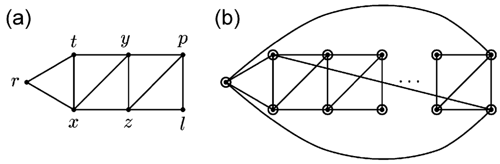

Fig. 14. A graph H that has the graph in (a) as a subgraph and satisfies $B_H = \emptyset$ is isomorphic to the graph in (b).

a contradiction. If, however, $tz \notin E_H$, then from (2) we have that $|V_H| > 5$. Thus, since $tz, rz \notin E_H$, the only way to expand the graph is by a new vertex w adjacent to t and r. Now, from Lemma 5, we have that $d_H(z) \geqslant 3$ and $d_H(w) \geqslant 3$. Since x, y, r, t are already saturated, the only possible configuration of H is shown in Fig. 11(b). But then, H has a butterfly $[r, t, w] \cup [w, k, z]$, a contradiction.

(iii) $rz, ry \notin E_H, tz \in E_H$. Note that this case is equivalent to the case (ii) when $tz \notin E_H$.

(iv) $rz, ry, tz \notin E_H$.

(1) There exists a vertex p adjacent to t and y in H. Since x, y, t are saturated, from Lemma 5 and the fact that H is butterfly-free, we have that $rp, pz \in E_H$, and then again $|V_H| \leqslant 8$, a contradiction (see Fig. 12).

(2) There is no vertex adjacent to t and y in H. Since $|V_H| > 5$, there is a vertex p adjacent to, without loss of generality, y and z. Since case (i) does not occur, we have that $tp \notin E_H$. Indeed, if $tp \in E_H$, we would have a graph (the graph in Fig. 13(a) induced by the vertices x, y, z, t, p) isomorphic to the graph in Fig. 10 induced by the vertices x, y, z, t, r (which we showed to be impossible). Suppose now that there is no vertex that is adjacent to both p and z, or adjacent to both r and t. Then, from Lemma 5 and $tp, rz \notin E_H$ follows that $pr \in E_H$. From the irredundancy of H and the fact that $tp, rz \notin E_H$ we have that there is a triangle $[p, r, l]$ in H that contains the edge pr. Now, our hypothesis (that there is no vertex that is adjacent to both p and z, or adjacent to both r and t) implies that $lt, lz \notin E_H$, and thus, the graph H has a butterfly $[r, p, l] \cup [r, t, x]$ (see Fig. 13(b)), a contradiction. We conclude, thus, that there is a vertex l adjacent to, without loss of generality, p and z (Fig. 14(a)). Note that a graph H that has the graph in Fig. 14(a) as a subgraph and satisfies $B_H = \emptyset$ is isomorphic to the graph in Fig. 14(b). Observe that H can have odd or even number of triangles (the configuration of the graph H with even number of triangles is similar to that of Fig. 14(b)). Since all the vertices in this graph are saturated, H is a component of G, and thus $|V_H| \geqslant 9$ (for otherwise, H would have $\mathscr{T}$, $|\mathscr{T}| \leqslant 2$, locally optimal in G). Observe that if $\widetilde{T}$ is a triangle in H, then $H - V_{\widetilde{T}}$ is isomorphic to the graph with full lines shown in Fig. 8(b) (but now with at least six vertices). Thus, Commit$(H - V_{\widetilde{T}})$ is a vertex-disjoint packing of triangles in $H - V_{\widetilde{T}}$ that covers all vertices of $H - V_{\widetilde{T}}$, except at most two. Since $|V_H| = |V_{H-V_{\widetilde{T}}}| + 3$, we have that $\widetilde{T} \cup \text{Commit}(H - V_{\widetilde{T}})$ is an optimal solution in H. $\square$

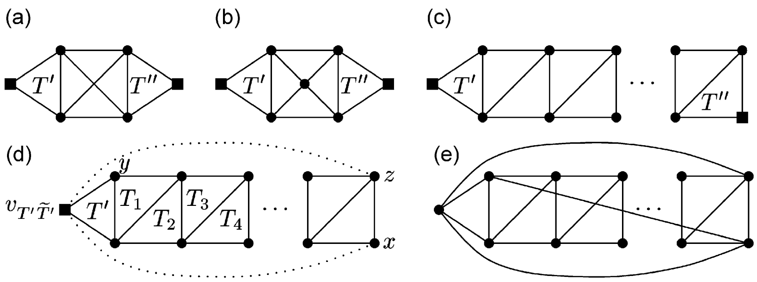

Fig. 15. Possible configurations of graph H. Each square vertex is a vertex common to two triangles in G whose union is a butterfly. The graph (c) has at least seven vertices. The graphs (d) and (e) have at least nine vertices, and $G[V_H]$ is a component of G (in (d) xy is not an edge of G, and dotted lines indicate edges that are in E_G, but not in E_H). Note that in (d) T' induces a butterfly with the triangle $[v_{T'\widetilde{T}'}, x, z]$. In (c)–(e) H can have odd or even number of triangles (the configuration of H with even number of triangles is similar to that of graphs shown on this figure).

Corollary 7. *For each iteration of the algorithm* $\mathrm{VT4}_k$, *the graph H is isomorphic to one of the graphs in Fig.* 15.

Proof. Follows from the proofs of Theorems 4 and 6. □

Theorem 8. *If $\mathscr{T}$ is locally optimal in G, then* $\mathrm{Accept}(\mathscr{T})$ *preserves the approximation ratio of the algorithm* $\mathrm{VT4}_k$.

Proof. We note, initially, that there exists a maximum packing of $\mathscr{T}_\mathrm{V}(G)$ which contains $\mathscr{T}$. Indeed, if a maximum packing of $\mathscr{T}_\mathrm{V}(G)$ does not contain $\mathscr{T}$, we can replace the triangles that intersect a triangle in $\mathscr{T}$, with $\mathscr{T}$. Hence, for $G' := G - \bigcup_{T\in\mathscr{T}} V_T$ we have $\mathrm{opt}(G') \geqslant \mathrm{opt}(G) - |\mathscr{T}|$. Thus, if a packing $\mathscr{A}$ of $\mathscr{T}_\mathrm{V}(G')$ satisfies $\mathrm{opt}(G') \leqslant \rho|\mathscr{A}|$ (with $\rho \geqslant 1$), then $\mathrm{opt}(G) \leqslant \mathrm{opt}(G') + |\mathscr{T}| \leqslant \rho|\mathscr{A}| + |\mathscr{T}| \leqslant \rho|\mathscr{A} \cup \mathscr{T}|$. □

Corollary 9. *If $\mathscr{T}$ is locally optimal in a graph G and $\mathscr{A}$ is a maximum vertex-disjoint packing of triangles in $G - \bigcup_{T\in\mathscr{T}} V_T$, then $\mathscr{A} \cup \mathscr{T}$ is a maximum packing of $\mathscr{T}_\mathrm{V}(G)$.*

Proof. Follows directly from the proof of Theorem 8 (take $\rho = 1$). □

Theorem 10. $\mathrm{Reduce}(H)$ *preserves the approximation ratio of the algorithm* $\mathrm{VT4}_k$.

Proof. Consider any iteration of the algorithm for which there exists a triangle in G with degree greater than 3 (where G is the graph in line 5), and $|B_H| = 2$. From Corollary 7 it follows that H is one of the graphs in Fig. 15(a), (b) or (c). Note that

$$\text{if } K \text{ is any of the graphs } H - V_{T'} - V_{T''},\ H - V_{T'} - v_{T''\widetilde{T}''},\ H - V_{T''} - v_{T'\widetilde{T}'},\ H - v_{T'\widetilde{T}'} - v_{T''\widetilde{T}''}, \text{ then } \mathrm{Commit}(K) \text{ is a maximum packing of } \mathscr{T}_\mathrm{V}(K). \tag{4}$$

Therefore, $\mathrm{Sol}_{T'T''}$ is a maximum packing of $\mathscr{T}_\mathrm{V}(H)$ that contains T' and T'', $\mathrm{Sol}_{T'\overline{T''}}$ is a maximum packing of $\mathscr{T}_\mathrm{V}(H)$ that contains T' but not T'', etc. Besides, for H isomorphic to one of the graphs in Fig. 15(a), (b) or (c), the equalities listed in the definition of $\mathrm{Reduce}(H)$ cover all possible relations between the cardinalities of the sets $\mathrm{Sol}_{T'T''}$, $\mathrm{Sol}_{T'\overline{T''}}$, $\mathrm{Sol}_{\overline{T'}T''}$ and $\mathrm{Sol}_{\overline{T'}\,\overline{T''}}$.

Suppose that the equality in 3(a) is satisfied. From the definition of graph H and the fact that $T' \cup \widetilde{T}'$ (resp. $T'' \cup \widetilde{T}''$) is a butterfly in G, we have that the triangles in H that have a vertex in common with T' (resp. T'') intersect pairwise (the proof is similar to the proof of Lemma 3). Moreover, we saw that T' and T'' are disjoint, and consequently, $\{T', T''\}$ is locally optimal in H. Thus, from statement (4) (take $K = H - V_{T'} - V_{T''}$) and Corollary 9, it follows that $\mathrm{Sol}_{T'T''}$ is a maximum packing of $\mathscr{T}_\mathrm{V}(H)$. Since $|\mathrm{Sol}_{T'T''}| = |\mathrm{Sol}_{\overline{T'}\,\overline{T''}}|$, $\mathrm{Sol}_{\overline{T'}\,\overline{T''}}$ is also a maximum packing of $\mathscr{T}_\mathrm{V}(H)$. Note that $\{T \in \mathscr{T}_G : T \text{ intersects a triangle in } \mathrm{Sol}_{\overline{T'}\,\overline{T''}}\} \subseteq \mathscr{T}_H$. Furthermore, $\mathscr{T}_H \subseteq \{T \in \mathscr{T}_G : T \text{ intersects a triangle in } \mathrm{Sol}_{\overline{T'}\,\overline{T''}}\}$, for otherwise, $\mathrm{Sol}_{\overline{T'}\,\overline{T''}}$ would not be a maximum packing of $\mathscr{T}_\mathrm{V}(H)$. Thus, $\mathrm{Sol}_{\overline{T'}\,\overline{T''}}$ is locally optimal in G and by Theorem 8, $\mathrm{Accept}(\mathrm{Sol}_{\overline{T'}\,\overline{T''}})$ preserves the approximation ratio of the algorithm.

Suppose now that the equality in 3(a) is not satisfied and $\widetilde{T}'=\widetilde{T}''$. Let $\mathscr{A}$ be a maximum packing of $\mathscr{T}_{\mathrm{V}}(H\cup\widetilde{T}')$. If $\widetilde{T}'\in\mathscr{A}$, then of course, $|\mathscr{A}|=|\mathrm{Sol}_{\overline{T}'\overline{T}''}|+1$; and if $\widetilde{T}'\notin\mathscr{A}$, then $|\mathscr{A}|=|\mathrm{Sol}_{T'T''}|$ (since $|\mathrm{Sol}_{T'\overline{T}''}|$, $|\mathrm{Sol}_{\overline{T}'T''}|$, $|\mathrm{Sol}_{\overline{T}'\overline{T}''}|\leqslant|\mathrm{Sol}_{T'T''}|$).The equality in 3(a) is not satisfied, that is, one of the equalities in 3(c) or (d) is satisfied, and hence, $|\mathrm{Sol}_{T'T''}|=|\mathrm{Sol}_{\overline{T}'\overline{T}''}|+1$. Therefore, $\mathrm{Sol}_{T'T''}$ is a maximum packing of $\mathscr{T}_{\mathrm{V}}(H\cup\widetilde{T}')$. Now, similarly as above, we conclude that $\{T\in\mathscr{T}_G : T \text{ intersects a triangle in } \mathrm{Sol}_{T'T''}\}=\mathscr{T}_H\cup\widetilde{T}'$. Hence, $\mathrm{Sol}_{T'T''}$ is locally optimal in G, and Theorem 8 implies that $\mathrm{Accept}(\mathrm{Sol}_{T'T''})$ preserves the approximation ratio of the algorithm.

If one of the equalities in 3(c) or (d) is satisfied and $\widetilde{T}'\neq\widetilde{T}''$, we define G_A to be the graph G before $\mathrm{Reduce1}(H)$ (resp. $\mathrm{Reduce2}(H)$) is applied, and G_P to be the graph G immediately after the application of $\mathrm{Reduce1}(H)$ (resp. $\mathrm{Reduce2}(H)$). Furthermore, let $\mathscr{A}_P$ be a maximal vertex-disjoint packing of triangles in G_P.

Suppose that the equality in 3(c) is satisfied and $\widetilde{T}'\neq\widetilde{T}''$. We show initially that

$$\mathrm{opt}(G_P)\geqslant\mathrm{opt}(G_A)-|\mathrm{Sol}_{\overline{T}'\overline{T}''}|+1. \tag{5}$$

Let $\mathscr{T}_A^*$ be a set of triangles that is a maximum packing of $\mathscr{T}_{\mathrm{V}}(G_A)$. If $T',T''\in\mathscr{T}_A^*$, then $|\mathscr{T}_A^*\cap\mathscr{T}_H|=|\mathrm{Sol}_{T'T''}|$, since $\mathrm{Sol}_{T'T''}$ is a maximum packing of $\mathscr{T}_{\mathrm{V}}(H)$ that contains T' and T''. Hence, $\mathscr{T}_A^*\backslash(\mathscr{T}_H\backslash\{T',T''\})$ is a packing of $\mathscr{T}_{\mathrm{V}}(G_P)$ whose size is $\mathrm{opt}(G_A)-|\mathrm{Sol}_{T'T''}|+2=\mathrm{opt}(G_A)-|\mathrm{Sol}_{\overline{T}'\overline{T}''}|+1$. If $T',T''\notin\mathscr{T}_A^*$, then $|\mathscr{T}_A^*\cap\mathscr{T}_H|=|\mathrm{Sol}_{\overline{T}'\overline{T}''}|$,and thus, $(\mathscr{T}_A^*\backslash\mathscr{T}_H)\cup\{T_H\}$ is a packing of $\mathscr{T}_{\mathrm{V}}(G_P)$ of cardinality $\mathrm{opt}(G_A)-|\mathrm{Sol}_{\overline{T}'\overline{T}''}|+1$. If, however, $T'\in\mathscr{T}_A^*$ and $T''\notin\mathscr{T}_A^*$, then $\mathscr{T}_A^*\backslash(\mathscr{T}_H\backslash\{T'\})$ is a packing of $\mathscr{T}_{\mathrm{V}}(G_P)$ of size $\mathrm{opt}(G_A)-|\mathrm{Sol}_{T'\overline{T}''}|+1=\mathrm{opt}(G_A)-|\mathrm{Sol}_{\overline{T}'\overline{T}''}|+1$. Similarly if $T'\notin\mathscr{T}_A^*$ and $T''\in\mathscr{T}_A^*$. Hence, the statement (5) holds.

Note that since $\mathscr{A}_P$ is maximal, at least one of T',T'',T_H is in $\mathscr{A}_P$. We define now the set $\mathscr{A}_A$ (in accordance with $\mathrm{Restore}(H)$):

$$\mathscr{A}_A:=\begin{cases}\mathscr{A}_P\backslash\{T_H\}\cup\mathrm{Sol}_{\overline{T}'\overline{T}''} & \text{if } T_H\in\mathscr{A}_P,\\ \mathscr{A}_P\cup\mathrm{Sol}_{T'T''} & \text{if } T',T''\in\mathscr{A}_P,\\ \mathscr{A}_P\cup\mathrm{Sol}_{T'\overline{T}''} & \text{if } T'\in\mathscr{A}_P,\ T''\notin\mathscr{A}_P,\\ \mathscr{A}_P\cup\mathrm{Sol}_{\overline{T}'T''} & \text{if } T'\notin\mathscr{A}_P,\ T''\in\mathscr{A}_P.\end{cases}$$

We show next that

$$|\mathscr{A}_A|=|\mathscr{A}_P|+|\mathrm{Sol}_{\overline{T}'\overline{T}''}|-1 \tag{6}$$

and

$$\mathscr{A}_A \text{ is a maximal packing in } \mathscr{T}_{\mathrm{V}}(G_A). \tag{7}$$

If $T_H\in\mathscr{A}_P$, then of course, $|\mathscr{A}_A|=|\mathscr{A}_P|+|\mathrm{Sol}_{\overline{T}'\overline{T}''}|-1$. If $T',T''\in\mathscr{A}_P$, then $|\mathscr{A}_A|=|\mathscr{A}_P|+|\mathrm{Sol}_{T'T''}|-2=|\mathscr{A}_P|+|\mathrm{Sol}_{\overline{T}'\overline{T}''}|-1$. If, however, $T'\in\mathscr{A}_P$, $T''\notin\mathscr{A}_P$, then $|\mathscr{A}_A|=|\mathscr{A}_P|+|\mathrm{Sol}_{T'\overline{T}''}|-1=|\mathscr{A}_P|+|\mathrm{Sol}_{\overline{T}'\overline{T}''}|-1$. Similarly if $T'\notin\mathscr{A}_P$, $T''\in\mathscr{A}_P$. Hence, claim (6) holds. From (4) and the maximality of $\mathscr{A}_P$, it follows that $\mathscr{A}_A$ is a maximal packing of $\mathscr{T}_{\mathrm{V}}(G_A)$ in all cases. Thus, we conclude that (7) holds.

If $\mathrm{opt}(G_P)\leqslant\rho|\mathscr{A}_P|$ (for some $\rho\geqslant 1$), then from (5) and (6) we have $\mathrm{opt}(G_A)\leqslant\mathrm{opt}(G_P)+|\mathrm{Sol}_{\overline{T}'\overline{T}''}|-1\leqslant\rho|\mathscr{A}_P|+|\mathrm{Sol}_{\overline{T}'\overline{T}''}|-1=\rho(|\mathscr{A}_A|-|\mathrm{Sol}_{\overline{T}'\overline{T}''}|+1)+|\mathrm{Sol}_{\overline{T}'\overline{T}''}|-1$. Note that $|\mathrm{Sol}_{\overline{T}'\overline{T}''}|\geqslant 1$,and thus $\mathrm{opt}(G_A)\leqslant\rho|\mathscr{A}_A|$.

We remark that the algorithm $\mathrm{MIS3}_k$ returns a maximal independent set of vertices, and hence, immediately after line 15, $\mathscr{A}^*$ is a maximal packing of $\mathscr{T}_{\mathrm{V}}(G)$. Observe that the restorations of subgraphs are done in the reverse order of their reductions. Hence, using (7), one can deduce by induction that for each $\mathrm{Reduce1}(H)$ applied in the algorithm, if G_P is the graph as defined previously, then $\mathscr{A}_P$ is a maximal packing of $\mathscr{T}_{\mathrm{V}}(G_P)$. It follows, thus, that $\mathrm{Restore}(H)$ is well defined, that is, at least one of T',T'',T_H belongs to $\mathscr{A}^*$ before applying $\mathrm{Restore}(H)$. Furthermore, as we saw, $\mathscr{A}_A$ is a packing of $\mathscr{T}_{\mathrm{V}}(G_A)$, and $\mathrm{opt}(G_P)\leqslant\rho|\mathscr{A}_P|$ implies $\mathrm{opt}(G_A)\leqslant\rho|\mathscr{A}_A|$. We conclude thus that each $\mathrm{Reduce1}(H)$ (and corresponding $\mathrm{Restore}(H)$) preserves the approximation ratio of the algorithm.

The proof of the fact that $\mathrm{Reduce2}(H)$ (and corresponding $\mathrm{Restore}(H)$) preserves the approximation ratio is analogous. □

Theorem 11. *The algorithm* $\mathrm{VT4}_k$ *is a* $(3-\frac{\sqrt{13}}{2}+\frac{13-\sqrt{13}}{52k})$*-approximation algorithm for* VTP-4.

Proof. According to Theorems 6, 8 and 10, all reductions applied in Algorithm $\mathrm{VT4}_k$ preserve the approximation ratio. Thus, the approximation ratio of $\mathrm{VT4}_k$ is that of $\mathrm{MIS3}_k$. □

It should be noted that Algorithm $\mathrm{VT4}_k$ does not search exhaustively for the graphs shown in Fig. 15; hence if no call is made to $\mathrm{MIS3}_k$, the algorithm $\mathrm{VT4}_k$ finds an optimal solution and can be implemented to run in no more than $\mathrm{O}(n^3)$ time, where n is the number of vertices of the input graph. The time complexity of $\mathrm{VT4}_k$ is dominated by that of $\mathrm{MIS3}_k$, which is $\mathrm{O}(n^{\mathrm{O}(k)})$. Thus, any improvement on the approximation ratio and/or time complexity of algorithms for the maximum independent set problem on graphs with maximum degree 3 will lead to an improvement on the quality of $\mathrm{VT4}_k$.

3. Algorithm for ETP on graphs with maximum degree 5

In this section we restrict our attention to graphs with maximum degree 5 and describe an approximation algorithm called ET5 for ETP on such graphs. For that, consider the graph shown in Fig. 16, called *Hajós graph*, which will be denoted by $H := H[T_1, T_2, T_3]$. This graph consists of a circuit of length 6 together with three chords that induce a triangle; the other three triangles containing one of these chords are denoted by T_1, T_2 and T_3.

Algorithm ET5**.**
Input: A graph G with maximum degree 5.
$\mathscr{A}^* \leftarrow \emptyset$
while G contains a Hajós graph $H = H[T_1, T_2, T_3]$
  do $\{\mathscr{A}^* \leftarrow \mathscr{A}^* \cup \{T_1, T_2, T_3\}, G \leftarrow G - E_H\}$
return $\mathscr{A}^* \cup \{T : E_T \in \mathrm{HS}(\mathscr{T}_{\mathrm{E}}(G), 3)\}$.

3.1. *Analysis of the performance ratio of* ET5

To obtain the approximation ratio of ET5, we need the following lemmas.

Lemma 12. *If G is a graph that contains a Hajós graph H, then the number of triangles in any maximum packing of* $\mathscr{T}_{\mathrm{E}}(G)$ *that share an edge with H is at most* 4.

The proof of the above lemma can be easily obtained by inspection (using the degree boundedness).

Lemma 13. *The algorithm* $\mathrm{HS}(\mathscr{T}_{\mathrm{E}}(G), 3)$ *is a* $\frac{4}{3}$*-approximation algorithm for the problem* ETP-5 *on a graph G that does not contain a Hajós graph.*

Proof. The proof is by induction on the number of triangles in G. For $|\mathscr{T}_G| = 0$ and $|\mathscr{T}_G| = 1$, the proof is immediate. Let G be a graph with $|\mathscr{T}_G| > 1$, $\mathscr{T}^*$ a maximum packing of $\mathscr{T}_{\mathrm{E}}(G)$, and $\mathscr{A}^* := \{T : E_T \in \mathrm{HS}(\mathscr{T}_{\mathrm{E}}(G), 3)\}$. We show next that there exists a collection of triangles $\mathscr{A}$ such that

$$\mathscr{A} \subseteq \mathscr{A}^*,\ |\mathscr{A}| \leqslant 3 \text{ and } |\mathscr{T}|/|\mathscr{A}| \leqslant \tfrac{4}{3}, \text{ where } \mathscr{T} \text{ is the set of triangles from } \mathscr{T}^* \text{ that share an edge with a triangle in } \mathscr{A}. \tag{8}$$

For that, we analyse four cases.

(i) If there exists a triangle $T \in \mathscr{A}^* \cap \mathscr{T}^*$, then we take $\mathscr{A} := \{T\}$.
(ii) If case (i) is not satisfied and $\mathscr{A}^*$ has distinct triangles T_1, T_2, T_3 such that $|V_{T_1} \cap V_{T_2}| = |V_{T_2} \cap V_{T_3}| = 1$, then we take $\mathscr{A} := \{T_1, T_2, T_3\}$ (see Fig. 17). Using the fact that G does not contain a Hajós graph and that the maximum

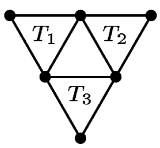

Fig. 16. Hajós graph denoted by $H[T_1, T_2, T_3]$.

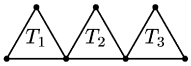

Fig. 17. Triangles T_1, T_2 and T_3 from the case (ii).

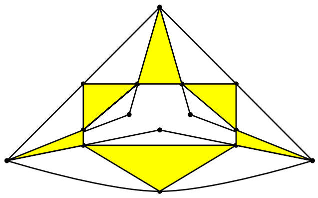

Fig. 18. The algorithm $\mathrm{HS}(\mathscr{T}_{\mathrm{E}}(G), 3)$ can possibly return the set of the six marked triangles.

degree of G is 5, it is easy to see that the number of triangles in $\mathscr{T}^*$ that share an edge with a triangle in $\mathscr{A}$ is at most 4.

(iii) If cases (i) and (ii) are not satisfied, and $\mathscr{A}^*$ has distinct triangles T_1 and T_2 such that $|V_{T_1} \cap V_{T_2}| = 1$, then we take $\mathscr{A} := \{T_1, T_2\}$. Suppose that $|\mathscr{T}| \geqslant 3$. If no triangle in $\mathscr{A}^*$, other than T_1 and T_2, shares an edge with a triangle in $\mathscr{T}$, then T_1 and T_2 could be replaced by $\mathscr{T}$, a contradiction. Thus, there is a triangle T_3 ($T_3 \neq T_1, T_2$) in $\mathscr{A}^*$ that shares an edge with a triangle in $\mathscr{T}$. Since G does not contain a Hajós graph, we have that T_1, T_2 and T_3 are as in case (ii), which is again a contradiction. We conclude, thus, that $|\mathscr{T}| \leqslant 2$.

(iv) If cases (i)–(iii) are not satisfied, then let T_1 be any triangle in $\mathscr{A}^*$ and $\mathscr{A} := \{T_1\}$. Suppose $|\mathscr{T}| \geqslant 2$. If the only triangle in $\mathscr{A}^*$ that shares an edge with a triangle in $\mathscr{T}$ is T_1, then T_1 could be replaced by $\mathscr{T}$, a contradiction. Thus, there is a triangleT_2 ($T_2 \neq T_1$) in $\mathscr{A}^*$ that shares an edge with a triangle in $\mathscr{T}$, that is, we have the case (iii), a contradiction. We conclude that $|\mathscr{T}| \leqslant 1$.

In all cases above we conclude that claim (8) holds. Let $\mathscr{A}$ be a collection that satisfies (8) and $G' := G - \bigcup_{T \in \mathscr{A}} E_T$. Observe that $\mathscr{A}^* \backslash \mathscr{A}$ is 3-optimal in G' (for otherwise, $\mathscr{A}^*$ would not be 3-optimal in G). Thus, the induction hypothesis implies that $\mathrm{opt}(G') \leqslant \frac{4}{3}|\mathscr{A}^* \backslash \mathscr{A}|$. Hence,

$$\mathrm{opt}(G) \leqslant \mathrm{opt}(G') + |\mathscr{T}| \leqslant \tfrac{4}{3}|\mathscr{A}^* \backslash \mathscr{A}| + |\mathscr{T}| = \tfrac{4}{3}|\mathscr{A}^*| - \tfrac{4}{3}|\mathscr{A}| + |\mathscr{T}|.$$

Since $|\mathscr{T}|/|\mathscr{A}| \leqslant \frac{4}{3}$, from the above inequalities we conclude that $\mathrm{opt}(G) \leqslant \frac{4}{3}|\mathscr{A}^*|$. $\square$

Theorem 14. *The algorithm* ET5 *is a* $\frac{4}{3}$*-approximation algorithm for the problem* ETP-5.

Proof. Follows directly from Lemmas 12 and Lemma 13. $\square$

It is not difficult to show that the ratio $\frac{4}{3}$ is tight. As for the time complexity of the algorithm ET5, we note that it can be implemented to run in $\mathrm{O}(n^3)$ time, where n is the number of vertices of the input graph.

A natural question that arises is whether the local search algorithm of Hurkens and Schrijver performs better on graphs with maximum degree 5. We note that the tight instance of the k-set packing problem presented by Hurkens and Schrijver [13] leads to a graph G for which the value of the $\mathrm{opt}_{\mathrm{ETP}}(G)$ to the value of the solution obtained by the algorithm $\mathrm{HS}(\mathscr{T}_{\mathrm{E}}(G), t)$ approaches $3/2$ (as t increases), but $\Delta(G) = 6$. However, we found some examples of irredundant graphs with maximum degree 5 for which the local search algorithm achieves the ratio $3/2$ for $t = 3$. In Fig. 18 we show a graph G with $\Delta(G) = 5$, for which $\mathrm{opt}_{\mathrm{ETP}}(G) = 9$ and $\mathrm{HS}(\mathscr{T}_{\mathrm{E}}(G), 3)$ can possibly output a solution with six triangles. For this example ET5 finds an optimal solution. We do not have examples for $t > 3$, but we note that in this case, the running time of the algorithm $\mathrm{HS}(\mathscr{T}_{\mathrm{E}}(G), t)$ is worse than that of ET5. Possibly, for $t > 3$, the performance ratio of $\mathrm{HS}(\mathscr{T}_{\mathrm{E}}(G), t)$ on graph with $\Delta(G) = 5$ is better than $3/2 + \varepsilon$. It would be interesting to decide whether this is the case.

4. The problem VTP on indifference graphs

It is well known that the class of the indifference graphs is contained in the class of interval graphs (see [8]). Formally, an interval graph is the intersection graph of a finite set of intervals in $\mathbb{R}$. When the intervals have the same length, we have a *unit interval graph*. When no interval contains another, we have a *proper interval graph*. Roberts [16] showed that the classes of indifference, proper interval and unit interval graphs are all equivalent.

For the next result we use the following characterization obtained by Looges and Olariu [15]: a graph G is an indifference graph if, and only if, there exists a linear order $<$ (which we call *canonical*) on V_G such that, for every choice of vertices u, v, w we have that

$$\text{if } u < v < w \text{ and } uw \in E_G \quad \text{then } uv, vw \in E_G. \tag{9}$$

Algorithm VTindifference**.**
Input: An indifference graph G.

Find a canonical order $v_1 < v_2 < \cdots < v_n$ on V_G
$\mathscr{A}^* \leftarrow \emptyset$
for $i \leftarrow 1$ to $|V_G| - 2$
  do if $v_i v_{i+2} \in E_G$ **then** $\{T \leftarrow [v_i, v_{i+1}, v_{i+2}],\ \mathscr{A}^* \leftarrow \mathscr{A}^* \cup T,\ G \leftarrow G - V_T\}$
return $\mathscr{A}^*$.

Theorem 15. *The algorithm* VTindifference *applied to an indifference graph G returns a maximum packing of* $\mathscr{T}_{\mathrm{V}}(G)$ *in linear time*.

Proof. Suppose that, for an indifference graph G, the collection $\mathscr{A}^*$ returned by the algorithm is not an optimal solution. Let $\mathscr{T}^*$ be a maximum packing of $\mathscr{T}_{\mathrm{V}}(G)$ that has the maximum number of triangles in common with $\mathscr{A}^*$, and i the smallest number such that $[v_i, v_{i+1}, v_{i+2}] \in \mathscr{A}^* \backslash \mathscr{T}^*$. We denote the triangle $[v_i, v_{i+1}, v_{i+2}]$ by T and show that

$$\text{if } T' \in \mathscr{T}^*, T' \cap T \neq \emptyset \text{ and } v_j \in V_{T'} \backslash V_T \text{ then } j > i + 2. \tag{10}$$

Suppose, by contradiction, that $j \leqslant i + 2$. Since $v_j \notin V_T$, we have $j < i$.

If v_j is a vertex of a triangle T'' in $\mathscr{A}^*$, from the definition of the number i and the fact that $j < i$, we have $T'' \in \mathscr{T}^* \cap \mathscr{A}^*$. Furthermore, $T \in \mathscr{A}^*$, $T' \cap T \neq \emptyset$ imply that $T' \notin \mathscr{A}^*$, and thus $T' \neq T''$. Note that v_j is covered by T' and T'', both in $\mathscr{T}^*$, which is impossible. It follows, thus, that v_j is not covered by a triangle in $\mathscr{A}^*$. We analyse two possible cases.

(i) $v_j v_{i+1} \in E_G$ or $v_j v_{i+2} \in E_G$. From $j < i$ and (9), it follows that $v_j v_{j+2} \in E_G$. Since v_j is not covered by a triangle in $\mathscr{A}^*$, the algorithm would include $[v_j, v_{j+1}, v_{j+2}]$ in the solution, a contradiction.
(ii) $v_j v_{i+1}, v_j v_{i+2} \notin E_G$. Then of course $v_i, v_j \in V_{T'}$. Let v_k be the third vertex of T'. Note that $v_k \notin V_T$.If $k \leqslant i + 2$ then, analogously as for v_j, it follows that $k < i$ and v_k is not covered by a triangle in $\mathscr{A}^*$. For $l := \min\{j, k\}$, the facts that $j, k < i$, $v_l v_i \in E_G$ and (9) imply $v_l v_{l+2} \in E_G$. But then, the algorithm would include $[v_l, v_{l+1}, v_{l+2}]$ in the solution, a contradiction. If $k > i + 2$ then $j < i$, $v_j v_k \in E_G$ and (9) imply that $v_j v_{j+2} \in E_G$, which is again a contradiction.

We have thus proved (10).

Now, if only one triangle T_1 from $\mathscr{T}^*$ intersects T, by replacing T_1 with T, we obtain a maximum packing of $\mathscr{T}_{\mathrm{V}}(G)$ that has more triangles in common with $\mathscr{A}^*$ than $\mathscr{T}^*$ does, a contradiction.

If exactly two triangles, T_1 and T_2, from $\mathscr{T}^*$ intersect T, then since T_1 and T_2 are vertex-disjoint, there are at least three distinct vertices, say v_k, v_l and v_p in $(V_{T_1} \cup V_{T_2}) \backslash V_T$. From (10) we have that $k, l, p > i + 2$. Suppose, without loss of generality, that $p = \max\{k, l, p\}$. Since v_p is adjacent to at least one of the v_i, v_{i+1}, v_{i+2}, and $k, l, p > i + 2$, (9) implies that $[v_k, v_l, v_p]$ is a triangle in G. By replacing T_1 and T_2 with T and $[v_k, v_l, v_p]$, we obtain an optimal solution which has more triangles in common with $\mathscr{A}^*$ than $\mathscr{T}^*$ does, a contradiction.

If three triangles, T_1, T_2, and T_3 from $\mathscr{T}^*$ intersect T, then there are six distinct vertices in $(V_{T_1} \cup V_{T_2} \cup V_{T_3}) \backslash V_T$. By (10), the indices of all those vertices are greater than $i + 2$. Furthermore, similarly as above, using (9) we obtain that they induce a complete subgraph in G. Thus, we can replace T_1, T_2 and T_3 with T and two other vertex-disjoint triangles, whose vertices are in $(V_{T_1} \cup V_{T_2} \cup V_{T_3}) \backslash V_T$. This is again a contradiction.

Since the canonical order can be computed in linear time [15], it follows that the algorithm is linear. □

We observe that there exists an interval graph that is not an indifference graph, for which our greedy algorithm fails to return a maximum vertex-disjoint triangle packing. However, algorithm VTindifference can be generalized in a way that, given an indifference graph G and a fixed integer $r \geqslant 2$, it returns a maximum number of vertex-disjoint complete subgraphs on r vertices, in G.

5. Conclusions and open problems

We improved the approximation ratio for the vertex and edge-disjoint triangle packing problems on graphs with bounded degree, instances known to be APX-hard [4]. We observe that the instance exhibited by Hurkens and Schrijver [13] to show that the ratio $3/2 + \varepsilon$ of their algorithm is tight yields an irredundant graph with maximum degree 4. We also note that another example can be constructed by considering the graph, say G, given by Yu and Goldschmidt [17], showing the tightness of local search for the problem of maximum independent set in k-claw free graphs (using G we can construct another graph G' for which the ratio $3/2 + \varepsilon$ is attained). These examples show that our algorithm $\mathrm{VT4}_k$ in fact performs better than the simple local search algorithm on graphs with maximum degree 4.

Any improvement to the $(3/2+\varepsilon)$-approximation ratio for the general case of VTP or ETP would be most interesting. Another open problem is to improve the lower bound for the approximation ratio of VTP or ETP. However, before tackling the general case, we may address the triangle packing problems on some special classes of graphs. For example, the class of the interval graphs. On this class the problem is interesting from both practical and theoretical point of view; this class contains the indifference graph class (on which, as we proved, VTP is polynomially solvable), while it is a subclass of chordal graphs (on which VTP problem is NP-hard [9]).

Acknowledgements

We thank the referees for the suggestions and remarks that improved the presentation of this paper.

References

[1] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi, Complexity and Approximation (Combinatorial Optimization Problems and their Approximability Properties), Springer, Berlin, 1999.
[2] B.S. Baker, Approximation algorithms for NP-complete problems on planar graphs, J. Assoc. Comput. Mach. 41 (1) (1994) 153–180.
[3] P. Berman, T. Fujito, On approximation properties of the independent set problem for low degree graphs, Theory Comput. Syst. 32 (2) (1999) 115–132.
[4] A. Caprara, R. Rizzi, Packing triangles in bounded degree graphs, Inform. Process. Lett. 84 (4) (2002) 175–180.
[5] M. Chlebík, J. Chlebíková, Inapproximability results for bounded variants of optimization problems, in: Lecture Notes in Computer Science, vol. 2751, Springer, Berlin, 2003, pp. 27–38.
[6] M. Chlebík, J. Chlebíková, On approximability of the independent set problem for low degree graph, in: Lecture Notes in Computer Science, vol. 3104, Springer, Berlin, 2004, pp. 47–56.
[7] G. Cornuéjols, D. Hartvigsen, W. Pulleyblank, Packing subgraphs in a graph, Oper. Res. Lett. 1(4) (1981/1982) 139–143.
[8] M.C. Golumbic, Algorithmic graph theory and perfect graphs, Annals of Discrete Mathematics, vol. 57, second ed., Elsevier Science, Amsterdam, 2004.
[9] V. Guruswami, C. Pandu Rangan, M.S. Chang, G.J. Chang, C.K. Wong, The Kr-packing problem, Computing 66 (1) (2001) 79–89.
[10] R. Hassin, S. Rubinstein, An approximation algorithm for maximum triangle packing, Discrete Appl. Math. 154 (6) (2006) 971–979.
[11] I. Holyer, The NP-completeness of some edge-partition problems, SIAM J. Comput. 10 (4) (1981) 713–717.
[12] H.B. Hunt III, M.V. Marathe, V. Radhakrishnan, S.S. Ravi, D.J. Rosenkrantz, R.E. Stearns, NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs, J. Algorithms 26 (2) (1998) 238–274.
[13] C.A.J. Hurkens, A. Schrijver, On the size of systems of sets every t of which have an SDR with an application to the worst-case ratio of heuristics for packing problems, SIAM J. Discrete Math. 2 (1) (1989) 68–72.
[14] R.M. Karp, On the computational complexity of combinatorial problems, Networks 5 (1) (1975) 45–68.
[15] P.J. Looges, S. Olariu, Optimal greedy algorithms for indifference graphs, Comput. Math. Appl. 25 (7) (1993) 15–25.
[16] F.S. Roberts, Indifference graphs, Proof Techniques in Graph Theory, Academic Press, New York, 1969, pp. 139–146.
[17] G. Yu, O. Goldschmidt, Local optimality and its application on independent sets for k-claw free graphs, J. Comb. Optim. 1 (2) (1997) 151–164.