Packing subgraphs in a graph

Gordana Manić, Yoshiko Wakabayashi

[1]Departamento de Ciência da Computação do IME–USP, São Paulo – SP

gocam@ime.usp.br, yw@ime.usp.br

Abstract. *For a fixed family $\mathcal{F}$ of graphs, an $\mathcal{F}$-packing of a graph G is a set of pairwise vertex-disjoint (or edge-disjoint) subgraphs of G, each isomorphic to an element of $\mathcal{F}$. We focus on the algorithmic aspects of the problem of finding an $\mathcal{F}$-packing that maximizes the number of covered edges. We present results for $\mathcal{F} = \{K_3\}$ and $\mathcal{F} = \{K_2, K_3\}$. For $\mathcal{F} = \{K_3\}$, we present approximation algorithms for bounded degree graphs, improving the ratio known for the general case. When $\mathcal{F} = \{K_2, K_3\}$, we study the vertex-disjoint version. We prove that this problem is APX-hard even on graphs with maximum degree 4. Furthermore, we present a $(3/2+\varepsilon)$-approximation algorithm for arbitrary graphs, and a 1.4-approximation algorithm for graphs with maximum degree 4.*

1. Introduction

The maximum matching problem is a classical combinatorial optimization problem, known to be polynomially solvable. A natural generalization of this problem is that of finding an $\mathcal{F}$-*packing* of a given graph G, where $\mathcal{F}$ is a fixed family of graphs. More precisely, this problem is the following. Given a graph G, find a set of pairwise vertex-disjoint (or edge-disjoint) subgraphs of G, each isomorphic to an element of $\mathcal{F}$, that covers a maximum number of edges of G. We consider two variants of this problem: the vertex-disjoint and the edge-disjoint cases. Apart from its theoretical interest, this problem is also important from practical point of view, as it arises naturally in applications such as scheduling.

We point out that there is another variant of the $\mathcal{F}$-packing problem, in which the objective is to maximize the number of *vertices* that are covered. This variant is NP-hard, even when $\mathcal{F}$ consists of a single graph that has a component with at least three vertices [Hell and Kirkpatrick 1983]; and also when $\mathcal{F}$ contains only complete graphs with at least three vertices [Hell and Kirkpatrick 1984]. On the other hand, this (vertex version) problem is polynomially solvable for some non-trivial families $\mathcal{F}$, and many important results in matching theory can be generalized to those cases. For example when $\mathcal{F} = \{K_2, \ldots, K_r\}$, $r > 2$, these authors showed in 1984 that this problem is in P.

Approximation algorithms for NP-hard $\mathcal{F}$-packing problems have been investigated for only a few families $\mathcal{F}$. Among these families, a distinguished one is $\mathcal{F} = \{K_3\}$, the classical triangle packing problem. This, apparently simple case, still lacks more satisfactory results, and is one of the subjects of our study. Another case we investigate here is when $\mathcal{F} = \{K_2, K_3\}$. For all of them the objective is to maximize the number of edges that are covered.

To distinguish the different problems we shall investigate here, we define each of them and give a mnemonic name. We denote by VTP (resp. ETP) the *Vertex-Disjoint Triangle Packing* (resp. *Edge-Disjoint Triangle Packing*) problem: given a graph G, find a collection of vertex-disjoint (resp. edge-disjoint) triangles in G that has a maximum cardinality. The problem VTP arises in scheduling, while ETP has applications in computational biology [Caprara and Rizzi 2002]. Both problems are well known to be NP-hard (see [Karp 1975] and [Holyer 1981]). Note that both problems are $\{K_3\}$-packing problems (maximizing a collection of triangles in the packing is equivalent to finding a collection of triangles that cover a maximum number of edges of G).

For the problems VTP and ETP one of the most relevant results was obtained by [Hurkens and Schrijver 1989]. They designed a simple local search algorithm for the maximum k-set packing problem which gives a $(\frac{3}{2} + \varepsilon)$-approximation algorithm for both VTP and ETP. 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 packings. Both problems admit a polynomial-time approximation scheme on planar graphs [Baker 1994] and λ-precision unit disk graphs (a result proved by Hunt et al. in 1998). The problem VTP is NP-complete when restricted to chordal graphs, while it is polynomially solvable on split graphs and cographs [Guruswami et al. 2001].

The remaining of this paper is organized as follows. In Sections 2 and 3 we present results for the problems VTP and ETP. We show a $(3-\frac{\sqrt{13}}{2}+\varepsilon)$-approximation algorithm for VTP on graphs with maximum degree 4, and a $\frac{4}{3}$-approximation algorithm for ETP on graphs with maximum degree 5. We also give an exact linear-time algorithm for VTP on indifference graphs (or, equivalently, proper interval graphs). In Section 4 we show results for the $\{K_2, K_3\}$-packing problem. We prove that the vertex-disjoint version of this problem is APX-hard even on graphs with maximum degree 4. Furthermore, we present a $(3/2+\varepsilon)$-approximation algorithm for arbitrary graphs, and a 1.4-approximation algorithm for graphs with maximum degree 4. Finally, in Section 5 we mention some ideas on how to extend the results for the $\{K_2, K_3\}$-packing problem to obtain results for the $\{K_2, \ldots, K_r\}$-packing problem.

1.1. Notation and some basic results

All graphs considered here are simple. Furthermore, for the problems VTP and ETP we assume that the input graph is such that each of its edges belongs to some triangle; these graphs will be called *irredundant*. We denote by VTP-k (resp. ETP-k), the problem VTP (resp. ETP) on graphs with maximum degree k. We recall that both VTP-3 and ETP-4 can be solved in polynomial time, whereas VTP-4 and ETP-5 are APX-hard [Caprara and Rizzi 2002].

For a given collection $\mathcal{T}$ of sets, any subcollection of pairwise disjoint sets is called a *packing* of $\mathcal{T}$. The *maximum k-set packing* problem (where k is a positive integer) is defined as follows: given a collection $\mathcal{T}$ of sets of size k over a certain domain, find a largest packing of $\mathcal{T}$. Let $\mathcal{T}_V(G)$ (resp. $\mathcal{T}_E(G)$) denote the collection of the sets of vertices (resp. edges) of all triangles in G. When referring to $\mathcal{T}_V(G)$ (resp. $\mathcal{T}_E(G)$) we adopt the convention that the packing refers to a subcollection of pairwise vertex-dsjoint (resp. edge-disjoint) sets. Then VTP (resp. ETP) can be stated as follows: given a graph G, find a maximum size packing of $\mathcal{T}_V(G)$ (resp. $\mathcal{T}_E(G)$). We also refer to a collection of vertex-disjoint (resp. edge-disjoint) triangles of a graph G as a packing of $\mathcal{T}_V(G)$ (resp. $\mathcal{T}_E(G)$).

The algorithm of [Hurkens and Schrijver 1989] for the maximum k-set packing problem of a given collection $\mathcal{T}$ will be denoted as $\mathrm{HS}(\mathcal{T}, k)$. It is a local search greedy heuristic that, for a fixed positive integer t, starting with any collection of k-sets, while possible, it replaces at most $p-1$ k-sets in the current collection with a set of $p \leq t$ disjoint k-sets that are not in the current collection, and updates the current collection. Its approximation ratio is $k/2+\varepsilon$, where ε depends on t.

2. Vertex-disjoint triangle packing (VTP)

In this section we focus on the vertex-disjoint triangle packing problem. 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, $d_G(T)$, is the number of triangles in G, different from T, that intersect T. We denote by $\mathcal{T}_G$ 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 $\mathcal{T}$ of vertex-disjoint triangles in G is *locally optimal* in G if $\{V_T\colon\ T \in \mathcal{T}\}$ is a maximum packing of the family $\{V_T\colon\ T \in \mathcal{T}_G,\ T \text{ intersects a triangle in } \mathcal{T}\}$.

2.1. The problem VTP on graphs with maximum degree 4

We describe in this section an algorithm, called $\mathrm{VT4}_p$, for VTP on graphs with maximum degree 4. 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 $\mathcal{T}_{G'}$ it applies the $(3-\frac{\sqrt{13}}{2}+\frac{13-\sqrt{13}}{52p})$-approximation algorithm of [Chlebík and Chlebíková 2004], which we denote by $\mathrm{MIS3}_p$ (where p is a fixed integer parameter), for the problem of finding a maximum cardinality independent set of vertices on graphs with maximum degree 3. We note that for $p=4$ the above ratio is slightly less than 1.25; and for $p>65$ it is slightly less than 1.2.

In each iteration of the algorithm $\mathrm{VT4}_p$, a set $\mathcal{T} \subseteq \mathcal{T}_G$, $|\mathcal{T}| \leq 2$, locally optimal in G is repeatedly added to $\mathcal{A}^*$ (the set to be returned by the algorithm) and G is updated. If G contains a triangle T with degree greater than 3, the algorithm finds a certain subgraph

H that contains T and applies an appropriate reduction (in a way that in the reduced graph the triangles obtained by this reduction have degree at most 3). The reduction is based on the number of triangles in H that forms a butterfly with a triangle not in H.

Algorithm $\mathrm{VT4}_p$
Input: A graph G with maximum degree 4.
Output: A set of vertex-disjoint triangles in G.

$\mathcal{A}^* \leftarrow \emptyset$
while exists a triangle in G with degree greater than 3
  while exists $\mathcal{T} \subseteq \mathcal{T}_G$, $|\mathcal{T}| \leq 2$, locally optimal in G $Accept(\mathcal{T})$
  if exists a triangle $T \in \mathcal{T}_G$ with $d_G(T) > 3$
    then $H \leftarrow$ maximal connected irredundant subgraph of G that
        contains T and does not contain any butterfly
      $B_H \leftarrow \{T' \in \mathcal{T}_H \colon \exists$ a triangle in $\mathcal{T}_G \backslash \mathcal{T}_H$ that forms a butterfly with T' in $G\}$
      if $|B_H| = 2$ **then** apply $Reduce(H)$
        else if $|B_H| = 0$
          then { take a triangle $\widetilde{T}$ in $\mathcal{T}_H$, $\mathrm{Sol}_H \leftarrow \widetilde{T} \cup Commit(H - V_{\widetilde{T}})$ }
        if $|B_H| = 1$ **then** $\mathrm{Sol}_H \leftarrow Commit(H)$
        $\mathcal{A}^* \leftarrow \mathcal{A}^* \cup \mathrm{Sol}_H$
if $G \neq \emptyset$ **then** $\mathcal{A}^* \leftarrow \mathcal{A}^* \cup \mathrm{MIS3}_p$(intersection graph of $\mathcal{T}_G$)
for every application of $Reduce(H)$ **do** $Restore(H)$
return $\mathcal{A}^*$

Each of the procedures is described next in more detail.

1. $Accept(\mathcal{T})$: Add $\mathcal{T}$ to $\mathcal{A}^*$ and delete from G the vertices of all triangles in $\mathcal{T}$.
2. $Commit(H)$: Set $\mathcal{E} := \emptyset$. While $H \neq \emptyset$, find a triangle T locally optimal in H, add T to $\mathcal{E}$ and delete V_T from H. Return $\mathcal{E}$.
3. $Reduce(H)$: Take $T', T'' \in B_H$ and $\widetilde{T}', \widetilde{T}'' \in \mathcal{T}_G \backslash \mathcal{T}_H$ such that $T' \cup \widetilde{T}'$ and $T'' \cup \widetilde{T}''$ are butterflies in G (possibly $\widetilde{T}' = \widetilde{T}''$). Let

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

   (a) If $|\mathrm{Sol}_{T'T''}| = |\mathrm{Sol}_{T'\overline{T}''}| = |\mathrm{Sol}_{\overline{T}'T''}| = |\mathrm{Sol}_{\overline{T}'\overline{T}''}|$, then $Accept\ (\mathrm{Sol}_{\overline{T}'\overline{T}''})$.
   (b) If the equalities in (a) are not satisfied and $\widetilde{T}' = \widetilde{T}''$, then $Accept\ (\mathrm{Sol}_{T'T''})$.
   (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 $Reduce\,1(H)$:

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

   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, $Reduce\,1(H)$ replaces all triangles of H, except T' and T'', with a new triangle T_H.

(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 $Reduce\,2(H)$:

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

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.

4. $Restore(H)$:
   (a) If the reduction applied to H was $Reduce\,1(H)$, then if T_H belongs to $\mathcal{A}^*$ before applying $Restore(H)$, this procedure removes T_H from $\mathcal{A}^*$ and adds to it the set $\mathrm{Sol}_{\overline{T}'\overline{T}''}$ (computed in the procedure $Reduce(H)$); if $T', T'' \in \mathcal{A}^*$, then $\mathcal{A}^* \leftarrow \mathcal{A}^* \cup \mathrm{Sol}_{T'T''}$; if $T' \in \mathcal{A}^*$, $T'' \notin \mathcal{A}^*$, then $\mathcal{A}^* \leftarrow \mathcal{A}^* \cup \mathrm{Sol}_{T'\overline{T}''}$; and if $T' \notin \mathcal{A}^*$, $T'' \in \mathcal{A}^*$, then $\mathcal{A}^* \leftarrow \mathcal{A}^* \cup \mathrm{Sol}_{\overline{T}'T''}$.
   (b) If, however, the reduction applied to H was $Reduce\,2(H)$, then if T_H^1 belongs to $\mathcal{A}^*$ before applying $Restore(H)$, this procedure adds $\mathrm{Sol}_{T'\overline{T}''}$ to $\mathcal{A}^*$ and removes T_H^1; if $T_H^2 \in \mathcal{A}^*$, then adds $\mathrm{Sol}_{\overline{T}'T''}$ to $\mathcal{A}^*$ and removes T_H^2; and if $T_H^1, T_H^2 \notin \mathcal{A}^*$, then adds $\mathrm{Sol}_{\overline{T}'\overline{T}''}$ to $\mathcal{A}^*$.

Making use of the structural properties of the input graph, maximum degree 4 and irredundancy (maintained in each iteration), we can prove that the graph H defined in the algorithm is isomorphic to one of the graphs in Figure 1. Thus, for each iteration of $\mathrm{VT4}_p$, the cardinality of B_H in line 8 is less than 3. If $|B_H| \leq 1$, then $G[V_H]$ is a component of G and Sol_H is an optimal solution in that component. We can also prove that $Reduce\,1$, $Reduce\,2$ (and the corresponding restoration) and $Accept$ are all approximation-preserving reductions, and thus the approximation ratio of $\mathrm{VT4}_p$ is that of $\mathrm{MIS3}_p$.

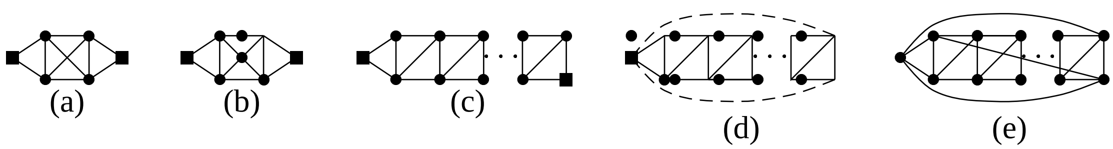


Figure 1. Possible configurations of graph H. Each square vertex is a vertex common to two triangles in G whose union is a butterfly. The graph (c) has at least 7 vertices.The graphs (d) and (e) have at least 9 vertices, and $G[V_H]$ is a component of G (in (d) dashed lines indicate edges not in E_H).

Theorem 2.1 *The algorithm* $\mathrm{VT4}_p$ *is a* $(3 - \frac{\sqrt{13}}{2} + \frac{13-\sqrt{13}}{52p})$*-approximation algorithm for* VTP-4*. It has time complexity* $O(n^{O(p)})$*, where* n *is the order of the input graph.*

2.2. The problem VTP on indifference graphs

For the next result we use the following characterization obtained by Looges and Olariu in 1993: 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 if $u < v < w$ and $uw \in E_G$, then $uv, vw \in E_G$.

Algorithm VTindifference
Input: An indifference graph G of order n.
Output: A maximum set of vertex-disjoint triangles in G.

```
Find a canonical order v_1 < v_2 < ··· < v_n on V_G
A* ← ∅
for i ← 1 to n − 2
   do if v_i v_{i+2} ∈ E_G then { T ← [v_i, v_{i+1}, v_{i+2}],  A* ← A* ∪ T,  G ← G − V_T }
return A*
```

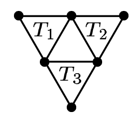
It is not difficult to prove that the algorithm above solves VTP on indifference graphs. Since the canonical order can be computed in linear time [Looges and Olariu 1993], it follows that the algorithm is linear.

3. Edge-disjoint triangle packing (ETP)

We restrict now our attention to graphs with maximum degree 5 and describe an approximation algorithm, called ET5, for the problem ETP on such graphs.

Algorithm ET5
Input: A graph G with maximum degree 5.
Output: A set of edge-disjoint triangles in G.

```
A* ← ∅
while G contains a Hajos graph H = H[T_1, T_2, T_3]  (see the figure)
    do {A* ← A* ∪ {T_1, T_2, T_3},  G ← G − E_H}
return A* ∪ {T :  E_T ∈ HS(𝒯_E(G), 3)}
```



Lemma 3.1 *The algorithm* $\mathrm{HS}(\mathcal{T}_E(G), 3)$ *is a* $\frac{4}{3}$*-approximation algorithm for the problem* ETP-5 *on graphs* G *that do not contain a Hajós graph.*

Theorem 3.2 *The algorithm* ET5 *is a* $\frac{4}{3}$*-approximation algorithm. Furthermore, its ratio* $\frac{4}{3}$ *is tight and it can be implemented to run in* $O(n^3)$ *time, where* n *is the order of the input graph.*

4. Packing edges and triangles

In this section we focus on the $\{K_2, K_3\}$-packing problem. For simplicity, we abbreviate it as $\mathrm{K_2K_3}$ -*packing problem*. We recall that when the objective is to maximize the number of covered vertices, the corresponding problem is polynomially solvable (see Section 1). In view of this fact, it is rather surprising that the following result holds.

Theorem 4.1 *The* $\mathrm{K_2K_3}$ *-packing problem is* APX*-hard even on graphs with maximum degree* 4.

The proof of this result is very elaborate. We show an L-reduction from the MAX2SAT3 problem, which is known to be APX-complete. We based our proof on the idea used in [Caprara and Rizzi 2002] to show the APX-completeness of the VTP problem. We also have a simpler proof showing that the $\mathrm{K_2K_3}$ -packing problem is APX-hard on graphs with maximum degree 5.

4.1. Approximation algorithms for $\mathrm{K}_2\mathrm{K}_3$ -packing

We consider in this section an algorithm denoted by $\mathcal{A}_{2,3}$ that works as follows: first it finds in the input graph G a maximal collection $\mathcal{T}$ of vertex-disjoint triangles, then removes this collection of triangles from G and finds in the remaining graph a maximum matching $\mathcal{E}$; then returns $\mathcal{T} \cup \mathcal{E}$.

Theorem 4.2 *Let A_{VTP} be a ρ-approximation algorithm for the* VTP *problem which produces for any input graph G a maximal vertex-disjoint set of triangles in G. Then the algorithm $\mathcal{A}_{2,3}$ is a $(1 + \frac{1}{3}\rho)$-approximation algorithm for the $\mathrm{K}_2\mathrm{K}_3$ -packing problem.*

Proof. [sketch] Let G be a graph and $\mathcal{A}$ the solution returned by the algorithm $\mathcal{A}_{2,3}$ applied to G. Let $\mathcal{O}$ be an optimal solution for the $\mathrm{K}_2\mathrm{K}_3$ -packing problem on G with the largest possible number of triangles in common with $\mathcal{A}$. Let t_i (resp. o_i), $0 \leq i \leq 3$, be the number of triangles of $\mathcal{A}$ (resp. $\mathcal{O}$) that intersect exactly i vertices of $\mathcal{T}_{\mathcal{O}}$ (resp. $\mathcal{T}_{\mathcal{A}}$). Observe that t_0 and o_0 must be zero. We now define e_1 (resp. e_0) as the number of edges in $\mathcal{E}_{\mathcal{O}}$ that intersect at least one (resp. none of) triangle of $\mathcal{A}$. Let $G' := G - \bigcup_{T \in \mathcal{T}_{\mathcal{A}}} V_T$. Note that a matching of G' can be obtained by taking one edge of each triangle of $\mathcal{O}$ that has exactly one vertex in common with a triangle of $\mathcal{A}$, and taking the edges of $\mathcal{E}_{\mathcal{O}}$ that have no vertex in common with any triangle of $\mathcal{A}$. Hence, as $\mathcal{E}_{\mathcal{A}}$ is a maximum matching of G', we have $|\mathcal{E}_{\mathcal{A}}| \geq o_1 + e_0$. From this inequality, the facts that $e_1 \leq 2t_1 + t_2$ and $3t_3 + 2t_2 + t_1 = 3o_3 + 2o_2 + o_1$ we obtain the desired ratio. □

Corollary 4.3 *There is a $(\frac{3}{2} + \varepsilon)$-approximation algorithm for the $\mathrm{K}_2\mathrm{K}_3$ -packing problem.*

Proof. Apply Theorem 4.2 taking $A_{\mathrm{VTP}} := \mathrm{HS}(\mathcal{T}_V(G), 3)$. □

Corollary 4.4 *There is a 1.4-approximation algorithm for the $\mathrm{K}_2\mathrm{K}_3$ -packing problem on graphs with maximum degree 4.*

Proof. It follows from Theorem 4.2 and Theorem 2.1. □

5. Concluding remarks

A more precise analysis of the algorithm $\mathcal{A}_{2,3}$, using some ideas similar to the one we used in Theorem 4.2, shows that this algorithm is in fact a $3/2$-approximation (a result obtained by R. Yuster). Recently, we have generalized the ideas used in this proof and obtained approximation results for the $\{K_2, \ldots, K_r\}$-packing problem (see [Chataigner et al. 2007]). Basically, we have shown that for this problem, a simple greedy algorithm has approximation ratio 2; and a more sophisticated algorithm that uses the heuristic of Hurkens and Schrijver yields ratios smaller than 2 for $r = 4$ and $r = 5$. The results mentioned in Sections 2–4 appear in [Manić and Wakabayashi 2005]. They are part of the thesis of the first author (supervised by the second author). The full text of the thesis can be found at http://www.ime.usp.br/∼gocam/tese_gordana_www.ps.

References

Baker, B. S. (1994). Approximation algorithms for NP-complete problems on planar graphs. *J. Assoc. Comput. Mach.*, 41(1):153–180.

Caprara, A. and Rizzi, R. (2002). Packing triangles in bounded degree graphs. *Inform. Process. Lett.*, 84(4):175–180.

Chataigner, F., Manić, G., Wakabayashi, Y., and Yuster, R. (2007). Approximation algorithms and hardness results for the clique packing problem. Extended abstract accepted to EuroComb 2007, to appear in *Electronic Notes in Discrete Mathematics* (full paper submitted to a journal).

Chlebík, M. and Chlebíková, J. (2004). On approximability of the independent set problem for low degree graphs. In *Lecture Notes in Computer Science 3104*, pages 47–56. Springer.

Guruswami, V., Pandu Rangan, C., Chang, M. S., Chang, G. J., and Wong, C. K. (2001). The K_r-packing problem. *Computing*, 66(1):79–89.

Hell, P. and Kirkpatrick, D. (1983). On the complexity of general graph factor problems. *SIAM J. Comput.*, 12(3):601–609.

Hell, P. and Kirkpatrick, D. (1984). Packings by cliques and by finite families of graphs. *Discrete Math.*, 49:45–59.

Holyer, I. (1981). The NP-completeness of some edge-partition problems. *SIAM J. Comput.*, 10(4):713–717.

Hurkens, C. A. J. and Schrijver, A. (1989). 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):68–72.

Karp, R. M. (1975). On the computational complexity of combinatorial problems. *Networks*, 5(1):45–68.

Looges, P. J. and Olariu, S. (1993). Optimal greedy algorithms for indifference graphs. *Comput. Math. Appl.*, 25(7):15–25.

Manić, G. and Wakabayashi, Y. (2005). Packing triangles in low degree graphs and indifference graphs. In *Discrete Math. and Theoretical Computer Science*, volume AE – EuroComb 2005 special volume, pages 251–256 (full paper submitted to a journal).