Computing 66, 79–89 (2001)

Computing

The K_r-Packing Problem*

V. Guruswami†, Cambridge, **C. Pandu Rangan**, Madras, **M. S. Chang** and **G. J. Chang**‡, Taiwan, and **C. K. Wong**§, Hong Kong

Received September 27, 1999; revised August 14, 2000

Abstract

For a fixed integer $r \geq 2$, the K_r-packing problem is to find the maximum number of pairwise *vertex-disjoint* K_r's (complete graphs on r vertices) in a given graph. The K_r-factor problem asks for the existence of a partition of the vertex set of a graph into K_r's. The K_r-packing problem is a natural generalization of the classical matching problem, but turns out to be much harder for $r \geq 3$ – it is known that for $r \geq 3$ the K_r-factor problem is NP-complete for graphs with clique number r [16]. This paper considers the complexity of the K_r-packing problem on restricted classes of graphs.

We first prove that for $r \geq 3$ the K_r-packing problem is NP-complete even when restrict to chordal graphs, planar graphs (for $r = 3$, 4 only), line graphs and total graphs. The hardness result for K_3-packing on chordal graphs answers an open question raised in [6]. We also give (simple) polynomial algorithms for the K_3-packing and the K_r-factor problems on split graphs (this is interesting in light of the fact that K_r-packing becomes NP-complete on split graphs for $r \geq 4$), and for the K_r-packing problem on cographs.

AMS Subject Classifications: 05C70, 05C85, 68Q20.

Key Words: Matching, K_r-packing, K_r-factor, NP-completeness, chordal graph, split graph, cograph, line graph.

1. Introduction

A *matching of a graph G* is a subset M of $E(G)$ such that any two distinct edges in M are not adjacent. A matching M is *perfect* (also called a 1-*factor*) if every vertex in G is incident to exactly one edge in M. The notion of matching not only has a beautiful mathematical theory associated with it, but also has many applications in such diverse fields as transversal theory, assignment problems, network flows, multiprocessor scheduling, and the Chinese postman and traveling salesman problems.

---

* A preliminary version of this paper, under the title "The vertex-disjoint triangles problem" [11], was presented at the 25th Workshop on Graph-Theoretic Concepts in Computer Science, WG'98.

† Work done while at the Department of Computer Science & Engineering, Indian Institute of Technology, Madras.

‡ Supported in part by the National Science Council under grant NSC88-2115-M009-009 and the Lee and MTI Center for Networking at NCTU.

§ On leave from IBM T. J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York, USA.

Generalization of the classical matching problem is motivated by both theoretical and practical constraints and has also motivated a lot of research though most of them have only negative NP-completeness results [13, 15, 16]. For a fixed family $\mathscr{G}$ of graphs, a (*strict*) $\mathscr{G}$*-packing* of a graph H is a set $\{G_1, G_2, \ldots, G_d\}$ of vertex-disjoint (induced) subgraphs of H such that each G_i is isomorphic to some G in $\mathscr{G}$. A (*strictly*) *perfect* $\mathscr{G}$*-packing* or (*strict*) $\mathscr{G}$*-factor* of a graph H is a (strict) $\mathscr{G}$-packing such that the sets $V(G_i)$ $(1 \leq i \leq d)$ partition $V(H)$. We write G-packing for $\{G\}$-packing and G-factor for $\{G\}$-factor. Clearly, a K_2-packing is just a matching and a K_2-factor is a perfect matching. The G-packing problem is to find the maximum size $p_G(H)$ of a G-packing of a given graph H. The G-factor problem asks if a graph has a G-packing. The well-known results by Kirkpatrick and Hell [16] are that

(1) if G is not of the form $\alpha \cdot K_1 \cup \beta \cdot K_2$, then the G-factor problem is NP-complete (and consequently, the G-packing problem is NP-complete);

(2) if G has at least three vertices, then the strict G-factor problem is NP-complete (and consequently, the strict G-packing problem is NP-complete).

The focus of this paper is for the case when $G = K_r$. In this case a K_r-packing (K_r-factor) is the same as a strict K_r-packing (strict K_r-factor). We use $p_r(H)$ for $p_{K_r}(H)$. Our goal is to determine the complexity of K_r-packing (and K_r-factoring) on some interesting classes of graphs. The only prior work in this direction seems to be [6], where a polynomial time algorithm is presented for the K_r-factor problem on chordal graphs for all $r \geq 3$. The complexity of the K_3-packing problem on chordal graphs is left open in [6] (for $r \geq 4$, the K_r-packing is easily seen to be NP-complete even on split graphs which form a subclass of chordal graphs). We answer the question raised in [6] and prove that the K_r-packing problem is NP-complete for chordal graphs for all $r \geq 3$. We also prove that the K_r-packing problem is NP-complete on planar graphs (for $r = 3$ only), line graphs and total graphs. We also provide a polynomial-time algorithm for the K_3-packing and the K_r-factor problems on split graphs; and for the K_r-packing problem on cographs.

In the rest of this section, we review some terminology. A graph is *chordal* if it contains no induced cycle of length greater than three. It is well-known [10] that a graph is chordal if and only if it has a *perfect elimination ordering*, i.e. an ordering $v_1, v_2, \ldots, v_n$ of $V(G)$ such that $N_i[v_i]$ is a clique, where

$$N_i[v_j] = \{v_j\} \cup \{v_k : k > j \text{ and } (v_j, v_k) \in E(G)\}$$

for $j \geq i$. We use $N[v_j]$ for $N_1[v_j]$. A *cograph* is a graph that has no induced P_4. A graph is *split* if its vertex set can be partitioned into an independent set and a clique. For a comprehensive treatment of these classes of graphs, see [10].

The *line graph* of a graph G is the graph $L(G)$ whose vertex set equals to the edge set of G and two vertices in $L(G)$ are adjacent if their corresponding edges in G are adjacent. The *total graph* of a graph G is the graph $T(G)$ whose vertex set is $V(G) \cup E(G)$ and two vertices of $T(G)$ are adjacent if the corresponding vertices or edges of G are adjacent. Note that both G and its line graph $L(G)$ are induced subgraphs of $T(G)$. For all graph-theoretic terms not defined explicitly here, see [12].

For a graph G, we use $\alpha(G)$ and $\omega(G)$ to denote for the size of a largest independent set in G and the clique number of G, respectively.

2. NP-Completeness Results

In this section we prove that for any fixed $r \geq 3$, the K_r-packing problem is NP-complete for chordal graphs, planar graphs (for $r = 3$, 4 only), line graphs and total graphs.[1]

2.1. Chordal Graphs

In this subsection, we prove that for any fixed $r \geq 3$, the K_r-packing problem is NP-complete on chordal graphs by reducing the satisfiability problem to it.[2]

Problem. The satisfiability problem (SAT).

Instance. A collection $C = \{c_1, c_2, \ldots, c_m\}$ of clauses over a set $U = \{u_1, u_2, \ldots, u_n\}$ of variables.

Question. Is there a truth assignment for U that satisfies all the clauses in C?

Theorem 2.1. *For any fixed $r \geq 3$, the K_r-packing problem is NP-complete in chordal graphs.*

Proof: The (decision version of the) problem is clearly in NP, we only establish NP-hardness by reducing SAT to the K_r-packing problem on chordal graphs. Given an instance of SAT, consider the following graph G with vertex set $V(G) = \cup_{k=1}^{r} X_k \cup Y$ and edge set $E(G) = KK \cup XC \cup XC' \cup XY$, where the vertices and edges are described below.

- For each $1 \leq k \leq r$, we construct $X_k = \{x[i,j,k], \bar{x}[i,j,k] : 1 \leq i \leq n, 1 \leq j \leq m\}$.
- For each clause c_j, we construct a vertex $y[j]$. Let $Y = \{y[j] : 1 \leq j \leq m\}$.
- $K = X_r \cup Y$ form a clique, i.e., we have the edges $KK = \{(u,v) : u, v \in K, u \neq v\}$.
- $XC = \{(x[i,j,k], \ x[i,j,k']), \ (\bar{x}[i,j,k], \bar{x}[i,j,k'])\text{:} \ 1 \leq i \leq n, \ 1 \leq j \leq m, 1 \leq k < k' \leq r \text{ with } (k,k') \neq (1,r)\}$.

[1] Throughout the paper, whenever we say a problem is NP-complete, we will implicitly mean the decision version of the problem, but for convenience sake the optimization versions of the problems are considered in the reductions.

[2] For $r \geq 4$ an easier proof can be given as the problem is easily seen to be NP-complete even on the subclass of split graphs.

- $XC' = \{(x[i,j,k], \quad \bar{x}[i,j+1,r]), \quad (\bar{x}[i,j,k], x[i,j,r]): \quad 1 \le i \le n, \quad 1 \le j \le m, 2 \le k \le r-1\}$, where $m+1$ is considered as 1.
- $XY = \{(x[i,j,k], y[j]): \quad u_i \in c_j, \quad 1 \le k \le r-1\} \cup \{(\bar{x}[i,j,k], y[j]): \quad \bar{u}_i \in c_j, 1 \le k \le r-1\}$.

It is straightforward to verify that G is a chordal graph with $2rmn + m$ vertices. We claim that the SAT instance C is satisfiable if and only if G has $mn + m$ vertex-disjoint K_r's, and this will clearly complete the proof.

First, suppose that C is satisfiable. Then we have the following set $\mathscr{P}$ of $mn + m$ vertex-disjoint K_r's. If u_i is assigned a false value, then we have the following m K_r's:

$$A_{ij} = \{x[i,j,2], x[i,j,3], \ldots, x[i,j,r], \bar{x}[i,j+1,r]\},$$

where $1 \le j \le m$. If u_i is assigned a true value, then we have the following m K_r's:

$$\bar{A}_{ij} = \{\bar{x}[i,j,2], \bar{x}[i,j,3], \ldots, \bar{x}[i,j,r], x[i,j,r]\},$$

where $1 \le j \le m$. These give mn K_r's for $\mathscr{P}$. Next, for each clause c_j, $1 \le j \le m$, there exists a variable to satisfy it. Suppose $u_i \in c_j$ and u_i is assigned a true value. Then $B_j = \{x[i,j,1], x[i,j,2], \ldots, x[i,j,r-1], y[j]\}$ is a K_r for $\mathscr{P}$. Similarly, $B_j = \{\bar{x}[i,j,1], \bar{x}[i,j,2], \ldots, \bar{x}[i,j,r-1], y[j]\}$ is a K_r for $\mathscr{P}$ if $\bar{u}_i \in c_j$ and u_i is assigned a false value. These give further m K_r's for $\mathscr{P}$. Therefore, there are $mn + m$ vertex-disjoint K_r's in G if C is satisfiable.

On the other hand, suppose G has a set $\mathscr{P}$ of $mn + m$ vertex-disjoint K_r's. Since $X_1, X_2, \ldots, X_{r-1}$ are independent, every K_r in G has at least one vertex from K. Suppose $\mathscr{P}$ has p K_r's containing exactly one vertex in K which we call a Type-1 K_r, and q K_r's containing at least two vertices in K which we call a Type-2 K_r. Then $p + q = mn + m$. Since no vertex in X_1 is adjacent to a vertex in X_r, the vertex of K in a Type-1 K_r must be in Y and so $p \le m$. Consequently,

$$2mm + m = 2(mm + m) - m \le 2(p+q) - p = p + 2q \le |K| = 2mm + m.$$

Therefore, $p = m$, $q = mm$ and each Type-2 K_r contains exactly two vertices in X_r. By the definition of G, Type-1 K_r's are B_j or $\bar{B}_j$, and Type-2 K_r's are A_{ij} or $\bar{A}_{ij}$ as above. Furthermore, for each variable u_i, if some A_{ij} (respectively, $\bar{A}_{ij}$) is in $\mathscr{P}$, then all $A_{i1}, A_{i2}, \ldots, A_{im}$ (respectively, all $\bar{A}_{i1}, \bar{A}_{i2}, \ldots, \bar{A}_{im}$) are in $\mathscr{P}$, for which case we assign u_i false (respectively, true). Then, for each clause c_j, if B_j (respectively, $\bar{B}_j$) in $\mathscr{P}$ is caused by $u_i \in c_j$ (respectively, $\bar{u}_i \in c_j$), then $\bar{A}_{ij}$ (respectively, A_{ij}) is in $\mathscr{P}$ and so, u_i is assigned true (respectively, false) which implies that c_j is true. Thus, C is satisfiable.

2.2. *Planar Graphs, Line Graphs and Total Graphs*

We now consider the K_r-packing problem for planar graphs (for $r = 3, 4$ only), line graphs and total graphs. The reductions are similar for these three classes of graphs.

Theorem 2.2. *The K_3-packing problem is NP-complete for planar graphs.*

Proof: The reduction is from the independent set problem which is NP-complete on planar cubic graphs [9]. Suppose H is an arbitrary cubic planar graph with $V(H) = \{1, 2, \ldots, n\}$ and $E(H) = \{e_1, e_2, \ldots, e_m\}$, where $e_i = (f(i), g(i))$ for $1 \leq i \leq m$. We will construct a planar graph G such that $p_3(G) = \alpha(H) + m$, and this will complete the proof of the theorem. To construct G, first construct a graph H' from H by subdividing each edge into 3 edges. More formally,

$$V(H') = V(H) \cup \{u_i, w_i : 1 \leq i \leq m\} \text{ and } E(H') = \cup_{i=1}^{m} \{(f(i), u_i), (u_i, w_i), (w_i, g(i))\}.$$

It is clear that H' is a K_3-free planar graph in which all vertices in $V(H)$ are of degree 3 and others are of degree 2. Next, construct H'' from H' by attaching a new pendant vertex to each vertex of degree 2.

The final graph G will now be the line graph $L(H'')$ of H''. It is easy to see that $L(H'')$ is planar. Since H'' is K_3-free with maximum degree 3, a K_3 in $L(H'')$ comprises of the 3 edges incident to a same vertex of H'', which is in fact in $V(H')$. Also, two K_3's in $L(H'')$ are vertex-disjoint if and only if the corresponding vertices are not adjacent in H'' (and hence in H'). It therefore follows that $p_3(L(H'')) = p_3(G) = \alpha(H')$.

Next, we show that $\alpha(H') = \alpha(H) + m$. Suppose S is a maximum independent set of H. Then for each edge e_i, either $f(i) \notin S$ or $g(i) \notin S$. So,

$$S \cup \{u_i : f(i) \notin S, 1 \leq i \leq m\} \cup \{w_i : f(i) \in S, g(i) \notin S, 1 \leq i \leq m\}$$

is an independent set of H' of size $|S| + m$. Therefore, $\alpha(H') \geq \alpha(H) + m$. On the other hand, suppose S' is a maximum independent set of H'. For each e_i, at most one of u_i and w_i is in S'. In the case of neither u_i nor w_i is in S', both $f(i)$ and $g(i)$ are in S' for otherwise S' is not maximum. We can then replace $f(i)$ by u_i to get a new maximum independent set. So, without loss of generality, we may assume that exactly one of u_i and w_i is in S' for $1 \leq i \leq m$, so that at most one of $F(i)$, $g(i)$ belongs to S' for each $i, 1 \leq i \leq m$. Hence, $S' \cap V$ is an independent set of H of size $|S'| - m$. This proves that $\alpha(H) \geq \alpha(H') - m$. Therefore, $p_3(G) = \alpha(H') = \alpha(H) + m$. □

A similar NP-completeness holds for the K_4-packing problem on planar graphs as well (note that K_r-packing for $r \geq 5$ is trivial on planar graphs). The reduction is very similar to that of Theorem 2.2, the only change is that we construct H'' from H by adding two pendant vertices to each vertex in H' of degree 2 and one pendant to each vertex in H' of degree 3. Once again $G = L(H'')$ will be planar and will satisfy $p_4(G) = \alpha(H') = \alpha(H) + m$. Hence we get

Theorem 2.3. *The K_4-packing problem is NP-complete on planar graphs.*

Theorem 2.4. *For any fixed $r \geq 3$, the K_r-packing problem is NP-complete on line graphs.*

Proof: The proof is essentially the same as the proof of Theorem 2.2 except when constructing H'' from H' we add $r - \deg_{H'}(x)$ new pendant vertices to each vertex x in H'. Note that this makes H'' a K_3-free graph in which all vertices of $V(H')$ are of degree r and the new pendant vertices are of degree one. □

Theorem 2.5. *For any fixed $r \geq 3$, the K_r-packing problem is NP-complete on total graphs.*

Proof: The proof is again a modification of the proof of Theorem 2.2 according to the following two cases.

For the case of $r \geq 4$, when constructing H'' from H' we add $r - 1 - \deg_{H'}(x)$ new pendant vertices to each vertex x in H'. This makes H'' a K_3-free graph in which all vertices of $V(H')$ are of degree $r - 1$ and the new pendant vertices are of degree one. In this case, a K_r in $T(H'')$ comprises of a vertex of $V(H')$ together with the $r - 1$ edges adjacent to it in H''. Then again $p_r(T(H'')) = \alpha(H') = \alpha(H) + m$.

For the case of $r = 3$, when constructing H'' from H' we add $4 - \deg_{H'}(x)$ new pendant vertices (with a special one called x') to each vertex x in H'. This makes H'' a K_3-free graph in which all vertices of $V(H')$ are of degree 4 and the new pendant vertices are of degree one. In this case, there are three possibilities for a K_3 in $T(H'')$:

1. An edge of H'' together with its two end vertices.
2. A vertex of $V(H'')$ together with two edges adjacent to it in H''.
3. Three edges in H'' that are adjacent to a vertex of $V(H')$.

Let $\mathscr{P}$ be a maximum K_3-packing of $T(H'')$. For each $x \in V(H')$, at least one element in $C_x = \{x, (x, x'), x'\}$ is in some K_3 in $\mathscr{P}$. It is easy to see that $\mathscr{P}$ may be modified without reducing its cardinality to include C_x for all $x \in V(H')$. Then, these C_x's are the only Type (1) K_3's in $\mathscr{P}$, and there is no Type (2) K_3 in $\mathscr{P}$. Therefore, $p_3(T(H'')) = n + p_3(L(\bar{H}'')) = n + \alpha(H') = n + \alpha(H) + m$, where $\bar{H}''$ is the graph obtained from H'' by deleting x' and (x, x') for each $x \in V(H')$ from H''. (Recall that $\alpha(H') = \alpha(H) + m$ was shown in the proof of Theorem 2.2.) □

3. K_r-Packings and K_r-Factors for Split Graphs

We now show that the K_3-packing and the K_r-factor problems can be solved in polynomial time for split graphs.[3] For $r \geq 4$, a simple reduction from K_3-packing on general graphs proves that K_r-packing is NP-complete on split graphs [6]. So this completely characterizes the complexity of the K_r-packing and K_r-factoring problems on the class of split graphs.

[3] The result for K_r-factors follows from the more general algorithm in [6] that works for chordal graphs. We present our algorithm as it follows from the same approach we use for K_3-packing.

Suppose G is a split graph where $V(G)$ is the disjoint union of an independent set S and a clique K. First note that to solve the K_r-packing problem, it is clearly enough to solve the *K_r^S-packing problem* which is to find the maximum number $p_r^S(G)$ of vertex-disjoint K_r's in G such that each K_r has one vertex in S and $r-1$ vertices in K. The following lemma is obvious as K is a clique.

Lemma 3.1. *If G is a split graph in which $V(G)$ is the disjoint union of an independent set S and a clique K, then $p_r(G) = \lfloor (p_r^S(G) + |K|)/r \rfloor$.*

We now proceed to transform the K_3^S-packing problem in a split graph G with $V(G) = S \cup K$ as above, to the maximum matching problem in a suitably defined graph H_3. The construction of the graph H_r for a general $r \geq 3$ is described below.

For each $u \in S \subseteq V$, split it into $r-1$ copies of u each adjacent to all neighbours of u in G. Also, make these $r-1$ vertices a clique. And make K an independent set in H_r. Let H_r be the resulting graph. More precisely, $H_r = (V', E')$ where

$$V' = K \cup \{u_i : u \in S, 1 \leq i \leq r-1\}$$

$$\begin{aligned} E' = &\{(u_i, u_j) : u \in S, 1 \leq i < j \leq r-1\} \\ &\cup \{(u_i, v) : (u, v) \in E, u \in S, 1 \leq i \leq r-1, v \in K\}. \end{aligned}$$

Lemma 3.2. *If $m(H_3)$ denotes the matching number of H_3, then $p_3^S(G) = m(H_3) - |S|$.*

Proof: Suppose M is a maximum matching of H_3. A vertex x is said to *M-match* another vertex y if $(x, y) \in M$. For every $u \in S \subseteq V$, exactly one of the following three cases occurs.

(1) Both u_1 and u_2 M-match vertices in K.

(2) Exactly one of u_1 and u_2 M-matches a vertex in K.

(3) u_1 M-matches u_2.

Note that Case (1) contributes two edges to M, and Case (2) or (3) one edge. It is clear that $\{\{u, v_1, v_2\} : (u_1, v_1) \in M, v_1 \in K, (u_2, v_2) \in M, v_2 \in K\}$ is a K_4^S-packing of G whose size is $m(H_3) - |S|$. Then $p_3^S(G) \geq m(H_3) - |S|$. On the other hand, suppose $\mathscr{P}$ is a K_3^S-packing of G of size $p_3^S(G)$. Then

$$\begin{aligned} M = &\{(u_1, v_1), (u_2, v_2) : \{u, v_1, v_2\} \in \mathscr{P}, u \in S, v_1 \in K, v_2 \in K\} \cup \\ &\{\{u_1, u_2\} : \text{there is no } K_3 \text{ in } \mathscr{P} \text{ containing } u\} \end{aligned}$$

is a matching of H_3 of size $p_3^S(G) + |S|$. Thus, $m(H_3) \geq p_3^S(G) + |S|$. Both inequalities together imply that $p_3^S(G) = m(H_3) - |S|$. $\square$

Theorem 3.3. *The K_3-packing problem can be solved in $O(m\sqrt{n})$ time for a split graph with n vertices and m edges.*

Proof: According to Lemmas 3.1 and 3.2, it is enough to find a maximum matching of H_3. Note that $|V(H_3)| = |K| + 2|S| \leq 2n$ and $|E(H_3)| \leq |E(G)| + |S| \leq m + n$. Hence a maximum matching of H_3 can be determined in $O(m\sqrt{n})$ time [17]. This completes the proof of the theorem. □

The method above can be adapted to one for the K_r-factor problem. First modify the construction of H_r into the *bipartite* graph H'_r by deleting the edges $\{(u_i, u_j) : u \in S, 1 \leq i < j < r\}$. Now it is easy to see that G has a K_r-factor if and only if r is a divisor of $|V(G)|$ and H'_r has a matching of size $(r-1)|S|$.

Theorem 3.4. *Suppose G is a split graph in which $V(G)$ is the disjoint union of an independent set S and a clique K. Then G has a K_r-factor if and only if r is a divisor of $|V(G)|$ and H'_r has a matching of size $(r-1)|S|$.*

4. K_r-Packings of Cographs

In this section, we devise a polynomial-time algorithm for the K_r-packing problem in cographs. The algorithm is based on dynamic programming.

Recall that cographs are graphs with no induced P_4. The class of cographs may also be defined recursively as follows [4]:

- K_1 is a cograph.
- If G' and G'' are cographs, then so is $G' \cup G''$ and $G' \times G'' (= (G'^c \cup G''^c)^c)$.

For technical reasons, we consider a more general problem as follows. For a graph G and nonnegative integers $n_3, \ldots, n_r$, let $f(G, n_3, \ldots, n_r)$ be the maximum integer n_2 such that G has a $\{K_2, \ldots, K_r\}$-packing in which there are exactly n_i copies of K_i's for $2 \leq i \leq r$; $f(G, n_3, \ldots, n_r) = -\infty$ if G has no such $\{K_2, \ldots, K_r\}$-packing. It is then easy to see that $p_r(G)$ is the maximum nonnegative integer n_r such that $f(G, 0, \ldots, 0, n_r) \geq 0$.

Lemma 4.1. *If $G = G' \cup G''$, then*

$$f(G, n_3, \ldots, n_r) = \max\{f(G', n'_3, \ldots, n'_r) + f(G'', n_3 - n'_3, \ldots, n_r - n'_r) : 0 \leq n'_i \leq n_i \text{ for } 3 \leq i \leq r\}.$$

Proof: The lemma follows from the fact that a $\{K_2, \ldots, K_r\}$-packing of G is the union of a $\{K_2, \ldots, K_r\}$-packing of G' and a $\{K_2, \ldots, K_r\}$-packing of G''. □

Lemma 4.2. *If $\mathscr{P}$ is a $\{K_2, \ldots, K_r\}$-packing of $G = G' \times G''$, then there exists a $\{K_2, \ldots, K_r\}$-packing $\mathscr{P}'$ covering precisely the same vertices as $\mathscr{P}$* does and *$\mathscr{P}'$ does not contain C' and C'' such that $C' \subseteq V(G')$ and $G'' \subseteq V(G'')$.*

Proof: Suppose $\mathscr{P}$ has $C' = \{u_1, u_2, \ldots, u_s\} \subseteq V(G')$ and $C'' = \{v_1, v_2, \ldots, v_t\} \subseteq V(G'')$, where $s, t \geq 2$. Set $D' = \{v_1, u_2, \ldots, u_s\}$ and $D'' = \{u_1, v_2, \ldots, v_t\}$. Then $D' \cup D'' = C' \cup C''$. Replace C' and C'' by D' and D'' respectively in $\mathscr{P}$ and continue the process until the resulting $\mathscr{P}'$ has the required property. □

Lemma 4.3. *If $G = G' \times G''$ with $n' = |V(G')|$ and $n'' = |V(G'')|$, then*

$$f(G, n_3, \ldots, n_r) = \max n_2 = \max\{n_{2,0} + n_{2,1} + n_{2,2}\},$$

where the maximum runs over all parameters satisfying the following conditions.

(1) *For $1 \leq i \leq r$, $n_i = \sum_{j=0}^{i} n_{i,j}$ where $n_{i,j} \geq 0$ for $0 \leq j \leq i$.*

(2) *$n'_j = \sum_{i=j}^{r}$ and $n''_j = \sum_{i=j}^{r} n_{i,i-j}$ for $1 \leq j \leq r$.*

(3) *$f(G', n'_3, \ldots, n'_r) \geq n'_2$ and $f(G'', n''_3, \ldots, n''_r) \geq n''_2$.*

(4) *$n' = \sum_{j=1}^{r} jn'_j$ and $n'' = \sum_{j=1}^{r} jn''_j$.*

(5) *$\sum_{i=2}^{r} n_{i,i} = 0$ or $\sum_{i=2}^{r} n_{i,0} = 0$.*

Proof: Suppose $\mathscr{P}$ is a $\{K_2, \ldots, K_r\}$-packing of G. Adding those K_1's which are not covered by $\mathscr{P}$, we have a $\{K_1, \ldots, K_r\}$-factor $\mathscr{F}$ of G which has n_i copies of K_i's for $1 \leq i \leq r$. Let $n_{i,j}$ be the number of K_i's of $\mathscr{F}$ whose intersection with $V(G')$ are K_j's and with $V(G'')$ are K_{i-j}'s.

Condition (1) follows from the definition of $n_{i,j}$.

In (2), n'_j (respectively, n''_j) is just the number of elements of Q whose intersection with $V(G')$ (respectively, $V(G'')$) are K_j's.

Condition (3) guarantees that there are enough K_2's in $V(G')$ and $V(G'')$.

Condition (4) counts the total numbers of vertices in $V(G')$ and $V(G'')$.

Condition (5) reflects the property in Lemma 4.2. □

Theorem 4.4. *For any fixed $r \geq 3$, the K_r-Packing problem can be solved in polynomial time for cographs.*

Proof: For any G' and G'' and $n_3, \ldots, n_r$ determining $f(G' \cup G'', n_3, \ldots, n_r)$ needs $O(n^{r-2})$ time by Lemma 4.1 and determining $f(G' \times G'', n_3, \ldots, n_r)$ needs $O(n^{1+2+\cdots+r})$ time by Lemma 4.3. In fact, the later can be reduced to $O(n^{1+2+\cdots+(r-1)})$ according to Condition (5). More careful analysis also leads to less time, e.g. $O(n)$ for $r = 3$.

The parse tree associated with a cograph G can be computed in linear time (see [3]). Once this has been done, the computation of $f(G, n_3, \ldots, n_r)$ is done at $O(n)$ internal nodes of the parse tree. The total time complexity is then a polynomial of n according to the arguments in the first paragraph. In particular $O(n^3)$ for $r = 3$ with a careful analysis. □

5. Conclusions

The main effort of this paper is to investigate the complexity of the K_r-packing and the K_r-factor problems on some well-known classes of graphs.

We prove that, for $r \geq 3$, the K_r-packing problem remains NP-complete on chordal graphs, planar graphs (for $r = 3, 4$ only), line graphs and total graphs. In contrast the K_r-factor problem can be solved in polynomial time on chordal graphs for all r [6]. This shows that the complexity of the factoring and packing problem can differ widely for specific classes of graphs.

On the algorithmic side, we gave polynomial algorithms for the K_3-packing and the K_r-factor problems on split graphs, and this completely settles the complexity of K_r-packing on split graphs since, for $r \geq 4$, the K_r-packing problem becomes NP-complete on split graphs. We also gave a polynomial time algorithm for the K_r-packing problem on cographs.

Acknowledgements

We thank Elias Dahlhaus for bringing [6] to our notice and for providing us with a copy of it. We also thank the referees for many useful suggestions.

References

[1] Chang, G. J., Farber M., Tuza, Z.: Algorithmic aspects of neighborhood numbers. SIAM J. Discr. Math. *6*, 24–29 (1993).
[2] Chang, M. S., Chan, Y. H., Chang, G. J., Yan, J. H.: Algorithmic aspects of the generalized clique-transversal problem on chordal graphs. Discr. Appl. Math. *66*, 189–203 (1996).
[3] Corneil, D. G., Perl, Y., Stewart, L. K.: A Linear recognition algorithm for cographs. SIAM J. Comput. *14*, 926–934 (1985).
[4] Corneil, D. G., Lerchs, H., Burlingham, L. S.: Complement reducible graphs. Discr. Appl. Math. *3*, 163–174 (1981).
[5] Cornuejols, G., Hartvigsen, D., Pulleyblank, W.: Packing subgraphs in a graph. Oper. Res. Lett. *1*, 139–143 (1982).
[6] Dahlhaus, E., Karpinski, M.: Matching and multidimensional matching in chordal and strongly chordal graphs. Discr. Appl. Math. *84*, 79–91 (1998).
[7] Edmonds, J.: Paths, trees and flowers. Can. J. Math. *17*, 449–469 (1965).
[8] Ferber, M.: Characterization of strongly chordal graphs. Discr. Math. *43*, 173–189 (1983).
[9] Garey, M. R., Johnson, D. S.: Computers and Intractability: A Guide to the Theory of NP-completeness. San Francisco: Freeman, 1979.
[10] Golumbic, M. C.: Algorithmic graph theory and perfect graphs. New York: Academic Press, 1980.
[11] Guruswami, V., Pandu Rangan, C., Chang, M. S., Chang, G. J., Wong, C. K.: The vertex-disjoint triangles problem. Proceedings of WG'98, Smolenice Castle, Slovakia, June 1998.
[12] Harary, F.: Graph Theory. Reading: Addison-Wesley, 1969.
[13] Hell, P., Kirkapatrick, D. G.: On generalized matching problems. Inform. Proc. Lett. *12*, 33–35 (1981).
[14] Hwang, S. F., Chang, G. J.: *k*-neighborhood-covering and -independence problems for chordal graphs. SIAM J. Discr. Math. *11*, 633–643 (1998).
[15] Kirkpatrick, D. G., Hell, P.: On the complexity of a generalized matching problem. In: Proc. 10th Annual Symposium on Theory of Computing, pp 240–245. New York: ACM, 1978.
[16] Kirkpatrick, D. G., Hell, P.: On the complexity of general graph factor problems. SIAM J. Comput. *12*, 601–609 (1983).

[17] Micali, S., Vazirani, V. V.: An $O(\sqrt{|V|}|E|)$ algorithm for finding maximum matching in general graphs. Proc. 21st Annual Symposium on the Foundation of Computer Science, pp 17–27 (1980).

Venkatesan Guruswami
MIT Laboratory for Computer Science
Cambridge, MA 01239, USA
venkat@theory.lcs.mit.edu

C. Pandu Rangan
Department of Computer Science and Engineering
Indian Institute of Technology
Madras-600 036, India
rangan@iitm.ernet.in

M. S. Chang
Department of Computer Science
and Information Engineering
National Chung Cheng University
Ming-Hsiun, Chiayi 621
Taiwan, Republic of China
mschang@cs.ccu.edu.tw

G. J. Chang
Department of Applied Mathematics
National Chaio Tung University
Hsinchu 30050
Taiwan, Republic of China
gjchang@math.nctu.edu.tw

C. K. Wong
Department of Computer Science and Engineering
Chinese University of Hong Kong, Hong Kong
wongck@cse.cuhk.edu.hk