Labelled packing functions in graphs☆

Erica G. Hinrichsen[1], Valeria A. Leoni[1,2,*], Martín D. Safe[3]

Abstract

Given a positive integer k and a graph G, a *k-limited packing* in G is a subset B of its vertex set such that each closed vertex neighborhood of G has at most k vertices of B (Gallant et al., 2010). A first generalization of this concept deals with a subset of vertices that cannot be in the set B and also, the number k is not a constant but it depends on the vertex neighborhood (Dobson et al., 2010). As another variation, a $\{k\}$-*packing function* f of G assigns a non-negative integer to the vertices of G in such a way that the sum of the values of f over each closed vertex neighborhood is at most k (Hinrichsen et al., 2014). The three associated decision problems are NP-complete in the general case. We introduce *L-packing functions* as a unified notion that generalizes all limited packing concepts introduced up to now. We present a linear time algorithm that solves the problem of finding the maximum weight of an L-packing function in strongly chordal graphs when a strong elimination ordering is given that includes the linear algorithm for $\{k\}$-packing functions in strongly chordal graphs (2014). Besides, we show how the algorithm can be used to solve the known clique-independence problem on strongly chordal graphs in linear time (G. Chang et al., 1993).

Keywords: computational complexity, labelled packing problem, strongly chordal graph, linear time algorithm, clique-independence

1. Introduction and notation

Packing problems are special location problems, challenging both from a theoretical and a computational point of view, even for several graph classes where other general hard problems are tractable, like bipartite graphs or chordal graphs.

Throughout this work, G will be a simple undirected graph with vertex set $V(G)$. For a given graph G and a function $f : V(G) \to \mathbb{R}$, we denote $f(T) = \sum_{v \in T} f(v)$, where $T \subseteq V(G)$. The *weight* of f is $f(V(G))$.

A *pendant vertex* in G is a vertex of degree one. For $v \in V(G)$, $N_G[v]$ denotes the closed neighborhood in G of vertex v. Two vertices $v, w \in V(G)$ are *true twins* if $N_G[v] = N_G[w]$; they are *false twins* if $N_G(v) = N_G(w)$ and v and w are not adjacent in G. A *clique* in G is a subset of pairwise adjacent vertices in G. An *independent set* (also called *stable set*) in G is a subset of pairwise non adjacent vertices in G.

The notion of a k-limited packing in a graph was introduced as a generalization of a 2-packing (or 1-limited packing) in a graph [10]. Given a graph G and a positive integer k, a set $B \subseteq V(G)$ is a *k-limited packing* in G if each

☆Partially supported by 1ING504 and PICT ANPCyT 2016-0410. M.D. Safe acknowledges partial support from ANPCyT Grant PICT-2017-1315 and Universidad Nacional del Sur Grants PGI 24/L103 and 24/L115.

*Corresponding author

Email address: valeoni@fceia.unr.edu.ar (Valeria A. Leoni)

[1]Depto. de Matemática, Universidad Nacional de Rosario, Rosario, Argentina.

[2]Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina.

[3]Departamento de Matemática, Universidad Nacional del Sur (UNS), Bahía Blanca, Argentina and INMABB, Universidad Nacional del Sur (UNS)-CONICET, Bahía Blanca, Argentina

closed neighborhood has at most k vertices of B, i.e. $|N_G[v] \cap B| \leq k$, for every $v \in V(G)$. For a fixed positive integer k, the *k-limited packing problem* consists in finding the maximum size of a k-limited packing in a given graph.

Among others, this notion has many applications in network security, in problems of placements of obnoxious facilities or outlets in a saturated market and in coding theory. For instance, it is a good model for location problems, where the interest is to place the maximum number of required facilities (such as garbage dumps) in such a way that —due to environmental restrictions— no more than a given number of them is near each of the agents in a given scenario [10].

Concerning computational complexity results, it is known that the k-limited packing problem is NP-complete, even for instances given by chordal graphs and bipartite graphs and it is polynomial time solvable for quasi-spiders and P_4-tidy graphs [6]. To the best of our knowledge, the most general result concerning polynomial time solvable instances of this problem is devoted to graphs of bounded clique-width [13]. Some recent results concerning polynomial-time solvable instances with unbounded clique-width, but only for the 2-limited packing problem ($k = 2$), are given by some grid graphs [3].

k-limited packings have a quite natural generalization in the following sense. First, different agents may have different facility requirements. Second, some places where to locate the facilities may be "not allowed". In this sense, a generalized version was introduced in [5]: given a vector of capacities $\mathbf{k} = (k(v)) \in \mathbb{Z}_+^{V(G)}$ and a set $\mathcal{A} \subseteq V(G)$ of "allowed vertices", a set $B \subseteq V(G)$ is a *$(\mathbf{k}, \mathcal{A})$-limited packing* in G if $B \subseteq \mathcal{A}$ and $|N_G[v] \cap B| \leq k(v)$, for every $v \in V(G)$. The *generalized limited packing problem* consists in finding the maximum size of a $(\mathbf{k}, \mathcal{A})$-limited packing in a given graph. When $k(v) = k$ for every $v \in V(G)$ and $\mathcal{A} = V(G)$, $(\mathbf{k}, \mathcal{A})$-limited packings are k-limited packings. Within graph classes that are closed under certain graph operations, the k-limited packing problem is "as hard as" the generalized limited packing problem [15]. As a corollary of a result from multiple domination, it is known that this generalized version is polynomial time solvable for strongly chordal graphs [5]. However, for the particular case of the k-limited packing problem for fixed k, no explicit algorithm is provided for strongly chordal graphs. More polynomial time solvable instances of it are presented in [7]

Observe that a k-limited packing in G can be considered as a function $f : V(G) \to \{0, 1\}$ such that $f(N_G[v]) \leq k$ for all $v \in V(G)$. In view of other applications, the notion of a *$\{k\}$-packing function* considers that in each vertex of the graph we are allowed to locate more than one facility. Precisely, given a graph G and a positive integer k, f is a *$\{k\}$-packing function* of G if it assigns non negative integers to the vertices of G in such a way that for all $v \in V(G)$, $f(N_G[v]) \leq k$ [13]. For a fixed positive integer k, the problem of finding the maximum weight of a $\{k\}$-packing function in a given graph is proved to be NP-complete [8]. Within graph classes that are closed under certain graph operations, the k-limited packing problem is "as hard as" the $\{k\}$-packing function problem [13]. Besides, an $O(n + m)$-time algorithm for a strongly chordal graph —when its strong elimination ordering is given— has been developed in [13], where n and m are respectively the number of vertices and edges of the given strongly chordal graph.

Strongly chordal graphs are up to now, the only graph class with unbounded clique-width for which all these problems are "tractable". Also, let us point out that the list of algorithms for the different versions of the problem on strongly chordal graphs mentioned above use different strategies. The aim of this work is to give a unified approach to all notions of limited packings in graphs introduced up to now in the literature; in particular, a unified linear time algorithm for strongly chordal graphs.

1.1. Structure of the paper

In Section 2, we introduce L-packing functions. We prove that —within graph classes that are closed under adding true twins and pendant vertices— the problem of finding the maximum weight of an L-packing function is not "harder" than the k-limited packing problem by Gallant et al. This is the case of the strongly chordal graph class.

Independently, we present a unified linear time algorithm that returns as output, an optimal L-packing function for a strongly chordal graph when a strong elimination ordering is given.

In Section 3, we relate L-packing functions with clique-independent sets in strongly chordal graphs. Our approach gives an alternative proof of the linearity of the clique-independence problem on this graph class already proved in [1].

We conclude the paper with some open problems in Section 4.

2. L-packing functions

We start this section by introducing the following definition:

Definition 1. *Given a graph G and a labelling function L, i.e. $L(v) = (t(v), k(v))$ defined on $V(G)$, where $t(v) \in \{0, \dots, t\} \cup \{A\}$ (t, a nonnegative fixed integer) and $k(v)$ are nonnegative integer numbers, an* L-packing function *of G is a function $f : V(G) \to \{0, \dots, t\}$ that satisfies the following two conditions:*

- *If $t(v) \neq A$ then $f(v) = t(v)$, and*
- *for all $v \in V(G)$, $f(N_G[v]) \leq k(v)$.*

Remark 2. *Notice that, when $k(v) > t|N_G[v]|$ for some $v \in V(G)$, we can equivalently consider the instance given by $k'(v) = t|N_G[v]|$, since every L-packing function f satisfies $f(N_G[v]) \leq t|N_G[v]|$. Besides, the element A serves as an indicator of whether an L-packing function f is allowed or not for a vertex v to assume a value from $\{0, \dots, t\}$ different from $t(v)$.*

Definition 3. ***Labelled packing problem***
Instance: *A graph G, a labelling function L on $V(G)$, an integer l.*
Question: *Is there an L-packing function of G with weight at least l?*

Remark 4. *The following statements are straightforward for a graph G and integer k fixed:*

- *When $t = 1$ and $L(v) = (A, k)$ for every $v \in V(G)$, if f is an L-packing function of G then $\{v \in V(G) : f(v) = 1\}$ is a k-limited packing of G and conversely, if $B \subseteq V(G)$ is a k-limited packing of G, then the function $f(v) := 1$ for $v \in B$ and $f(v) := 0$ otherwise is an L-packing function of G.*
- *When $t = 1$ and $t(v) \neq 1$ for every $v \in V(G)$, if f is an L-packing function of G then $\{v \in V(G) : f(v) = 1\}$ is a $(\mathbf{k}, \mathcal{A})$-limited packing of G for $\mathbf{k} = (k(v))$ and $\mathcal{A} = \{v \in V(G) : t(v) = A\}$, and conversely.*
- *When $t = k$ and $L(v) = (A, k)$ for every $v \in V(G)$, if f is an L-packing function of G then f is a $\{k\}$-packing function of G, and conversely.*

The following proposition shows that it is enough to consider only those instances where the labelling function takes the value 0 for every vertex v with $t(v) \neq A$:

Proposition 5. *Let G, $L = (t(v), k(v))$ and l define an instance of the labelled packing problem. Let also $L^*(v) = (t^*(v), k^*(v))$ on $V(G)$, where $t^*(v) := 0$ whenever $t(v) \neq A$, $t^*(v) := A$ otherwise and $k^*(v) := k(v) - t(\{w \in N_G[v] : t(w) \neq A\})$, for every $v \in V(G)$. Then there exists an L-packing function of G with weight at least l if and only if there exists an L^*-packing function of G with weight at least $l - t(\{w \in V(G) : t(w) \neq A\})$.*

Proof. Let f be an L-packing function of G with weight at least l. We define f^* on $V(G)$ such that $f^*(v) := 0$ if $t(v) \neq A$ and $f^*(v) := f(v)$ in other case.

Since for each $v \in V(G)$,

$$\begin{aligned} f^*(N_G[v]) &= f(\{w \in N_G[v] : t(v) = A\}) + t(\{w \in N_G[v] : t(w) \neq A\}) - t(\{w \in N_G[v] : t(w) \neq A\}) \\ &\leq k(v) - t(\{w \in N_G[v] : t(w) \neq A\}) = k^*(v), \end{aligned}$$

f^* is an L^*-packing function of G (where the labelling function L^* is defined in the statement) and

$$f^*(V(G)) = f(V(G)) - t(\{w \in V(G) : t(w) \neq A\}) \geq l - t(\{w \in V(G) : t(w) \neq A\}).$$

Conversely, let f^* be an L^*-packing function of G with weight at least $l - t(\{w \in V(G) : t(w) \neq A\})$. We define f on $V(G)$ such that $f(v) := t(v)$ if $t(v) \neq A$ and $f(v) := f^*(v)$, otherwise.

Since for each $v \in V(G)$,

$$\begin{aligned} f(N_G[v]) &= f^*(\{w \in N_G[v] : t(v) = A\}) + t(\{w \in N_G[v] : t(w) \neq A\}) \\ &\leq k^*(v) + t(\{w \in N_G[v] : t(w) \neq A\}) \leq k(v), \end{aligned}$$

it turns out that f is an L-packing function of G with weight

$$f(V(G)) = f^*(V(G)) + t(\{w \in V(G) : t(w) \neq A\}) \geq l. \qquad \square$$

It is clear that the reduction in the above proposition can be performed in linear time.

Proposition 5 will allow us to prove respectively in Theorems 8 and 9 below, that both variations —the generalized limited packing and the $\{k\}$-packing function problems— are "as hard as" the labelled packing problem introduced in this work. To this purpose, recall the following definitions. In particular, Definition 6 is in fact a generalization from [13].

Definition 6. *Given a graph G, vector of capacities $\mathbf{k} = (k(v))$ and a subset $R \subseteq V(G)$, $T_{R,\mathbf{k}}(G)$ denotes the graph obtained from G by replacing each vertex v of R with the complete graph $K_{k(v)}$ on $k(v)$ vertices.*

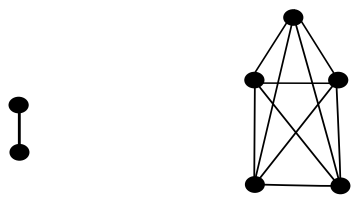

Figure 1: $T_{R,\mathbf{k}}(G)$ for $G = K_2$, $R = V(G)$ and $\mathbf{k} = (3, 2)$.

Definition 7. *Given a graph G and a subset $R \subseteq V(G)$, $P_R(G)$ denotes the graph obtained from G by adding for each $v \in R$, a pendant vertex adjacent to v.*

For each positive integer k, let us define the *k-labelled packing problem* as the labelled packing problem restricted to those instances such that $k(v) \leq k$ for each $v \in V(G)$. Similarly, we call *generalized k-limited packing problem* to the restriction of the generalized limited packing problem to those instances where $k(v) \leq k$ for each $v \in V(G)$. Notice that every instance of the generalized limited packing problem on some graph G is equivalent to an instance of the generalized $(\Delta(G) + 1)$-limited packing problem on the same graph G (where $\Delta(G)$ denotes the maximum degree of G) because any $k(v) > \Delta(G) + 1$ can be replaced by $\Delta(G) + 1$ without altering the result of the problem. We have:

Theorem 8. *For each positive integer k, the k-labelled packing problem can be reduced in linear time to the generalized k-limited packing problem.*

Proof. Take the instance of the k-labelled packing problem given by a graph G and a labelling function $L(v) = (t(v), k(v))$ defined on $V(G)$. From Proposition 5, we can assume $t(v) := 0$ whenever $t(v) \neq A$.

Now, consider the instance of the generalized k-limited packing problem given by the graph $G' := T_{R,\mathbf{k}}(G)$ for $R := \{v \in V(G) : t(v) = A\}$ (cf. Definition 6), vector of capacities $\widehat{\mathbf{k}}$, where $\widehat{k}(w) = k(v)$ if $w \in V(K_{k(v)})$ and $\widehat{k}(w) = 0$ in other case, and

$$\mathcal{A} = \bigcup_{v \in R} V(K_{k(v)}).$$

Given an L-packing function f of G, consider any subset of $f(v)$ vertices chosen from $V(G') \cap V(K_{k(v)})$, for each $v \in V(G)$. It is clear that this set is a $(\widehat{\mathbf{k}}, \mathcal{A})$-limited packing in G' with weight $f(V(G))$.

Conversely, given a $(\widehat{\mathbf{k}}, \mathcal{A})$-limited packing B in G', the function f defined by $f(v) := |B \cap V(K_{k(v)})|$ if $t(v) = A$ and $f(v) := 0$ if $t(v) \neq A$ is clearly an L-packing function of G with weight $f(V(G)) = |V(G)|$.

It is clear that the reduction performed takes at most $O(kn + k^2 m)$ time, where n and m denote the number of vertices and edges of G. Thus, for each k, it takes $O(n + m)$ time. □

Notice that, as a consequence of the above proof, every instance (G, L, l) of the labelled packing problem (where $L = (t(v), k(v))$) can be reduced to an instance of the generalized k-limited packing problem in time which is polynomial in the size G and in k, where $k = \max_{v \in V(G)} k(v)$.

Also, the following reduction can be proved. Some details are omitted due to the space constraints.

Theorem 9. *For each positive integer k, the k-labelled packing problem can be reduced in linear time to the $\{k\}$-packing function problem.*

Proof. Take the instance of the k-labelled packing problem given by a graph G and a labelling function $L(v) = (t(v), k(v))$ defined on $V(G)$. From Proposition 5, we can assume $t(v) := 0$ whenever $t(v) \neq A$.

Take the instance of the $\{k\}$-packing function problem given by the graph $G'' := P_R(G)$ obtained from Definition 7 with $R := V(G)$. Let us call w'' the vertex of degree one in $V(G'') \setminus V(G)$ adjacent to $w \in V(G)$.

We will prove that there exists an L-packing function of G with weight at least l if and only if there exists a $\{k\}$-packing function of G'' with weight at least $l' := l + \sum_{v \in V(G'') \setminus V(G)} [k - k(v)]$.

Given an L-packing function f of G with weight at least l, it can be proved that the function g over $V(G'')$ defined as follows is a $\{k\}$-packing function of G'' with weight at least l':

$$g(v) = \begin{cases} k - k(v) & \text{if} \quad v \in V(G'') \setminus V(G) \\ f(v) & \text{if} \quad v \in V(G). \end{cases}$$

Conversely, given a $\{k\}$-packing function g of G'' with weight at least l', g' defined as follows is another $\{k\}$-packing function of G'' with weight at least l':

$$g'(v) = \begin{cases} k(v) - g(v) & \text{if} \quad v \in V(G) \text{ and } t(v) = A \\ g(w) + g(v) & \text{if} \quad v = w'' \text{ and } t(w) = A \\ 0 & \text{if} \quad v \in V(G) \text{ and } t(v) = 0 \\ g(w) + g(v) & \text{if} \quad v = w'' \text{ and } t(w) = 0. \end{cases}$$

Now, the restriction of g' to $V(G)$ is an L-packing function of G with weight at least l.

Clearly, the reduction takes $O(n + m)$ time, where n and m denote the number of vertices of G □

The reduction in the above proof shows that every instance (G, L, l) of the labelled packing problem (where $L = (t(v), k(v)))$ can be reduced to an instance of the $\{k\}$-packing function problem for any k such that $k \geq \max_{v \in V(G)} k(v)$, in linear time of G (independently of k).

The reductions performed in the proofs of Theorems 8 and 9 above allow us to state:

Corollary 10. *Let k be a positive integer. Given two graph classes C and C' such that if $G \in C$ then $G^t \in C'$, where G^t is the graph obtained from G by adding a true twin to some vertex of G, then if the generalized k-limited packing problem is polynomial time solvable on C', the k-labelled packing problem is polynomial time solvable on C. Moreover, if the k-labelled packing problem is NP-complete on C, then the generalized k-limited packing problem is NP-complete on C'.*

Similarly, given two graph classes C'' and C''' such that if $G \in C''$ then $G^p \in C'''$, where G^p is the graph obtained from G by adding a pendant vertex to some vertex of G, then if the $\{k\}$-packing function problem is polynomial time solvable on C''', the k-labelled packing problem is polynomial time solvable on C''. Moreover, if the k-labelled packing problem is NP-complete on C'', then the $\{k\}$-packing function problem is NP-complete on C'''.

The appropriate combinations of Corollary 10 with Proposition 11 from [15] or Proposition 12 from [13] (both recalled below) provide NP-complete and/or polynomial-time solvable instances of the labelled packing problem. These facts prove that the k-limited packing problem (fixed k) is "as hard" as the labelled packing problem on graph classes that are closed under the transformations involved.

Proposition 11. *[15] Let G, $\mathcal{A} \subseteq V(G)$ and vector of capacities $\mathbf{k} = (k(v))$. Let $k := \max\{k(v) : v \in V(G)\}$ and consider $P_{V(G),k}(G)$ from Definition 7. Then G has a $(\mathbf{k}, \mathcal{A})$-limited packing of size at least α if and only if $P_{V(G),k}(G)$ has a k-limited packing of size at least $\alpha + \sum_{v \in V(G)}(k - k(v))$.*

Proposition 12. *[13] Let G be a graph and k a positive integer. Then G has a $\{k\}$-packing function of weight at least l if and only if $T_{V(G),\mathbf{k}}(G)$ (from Definition 7 with $k(v) = k$ for every v) has a k-limited packing of size at least l.*

Example 13. *Distance-hereditary graphs are characterized as the graphs that can be constructed from a single vertex by a sequence of additions of false twins, true twins and pendant vertices [4]. The linearity of the k-limited packing problem on distance-hereditary graphs for each positive integer k follows from a result in [13], since they have clique-width bounded by three and a 3-expression of them can be found in linear time [11]. By combining Corollary 10 with Proposition 11 we obtain that the k-labelled packing problem is linear time solvable on distance hereditary graphs for each positive integer k.*

Example 14. *Strongly chordal graphs (defined in the next section) are closed under adding twins and pendant vertices. The polinomiality of the k-limited packing problem on strongly chordal graphs can be derived from a result in [6]. By combining Corollary 10 with Proposition 12 we obtain that the labelled packing problem is polynomial time solvable on strongly chordal graphs. In fact, in the remainder of this section we present a linear time algorithm for strongly chordal graphs that returns as output an optimal L-packing function for any labelling function L. It includes the algorithm developed in [13].*

2.1. An algorithm for strongly chordal graphs

In this section we present a linear time algorithm for a strongly chordal graph if a strong elimination ordering is given, that in particular includes the algorithm in [13] and more generally, returns as output an optimal L-packing function for any labelling function L. The algorithm follows some ideas from [14] in the context of Y-dominating functions on graphs.

A *chord* of a cycle C in a graph G is an edge (u, v) not in C such that u and v lie in C. A graph G is *chordal* if it does not contain an induced chordless cycle on n vertices for any $n \geq 4$. A graph G is *strongly chordal* if it is chordal and every cycle of even length in G has an odd chord, i.e. a chord that connects two vertices that are at odd distance apart from each other in the cycle.

A vertex v of a graph G is called *simplicial* in G if $N_G[v]$ is a clique in G. A *strong elimination ordering* of a graph G is an ordering $(v_1, \ldots, v_n)$ of its vertices such that:

1. v_i is simplicial in the subgraph G_i induced by $\{v_i, v_{i+1} \ldots, v_n\}$, and
2. for $i \leq j \leq k$, if v_j and v_k belong to $N_{G_i}[v_i]$ then $N_{G_i}[v_j] \subseteq N_{G_i}[v_k]$.

A graph is *strongly chordal* if and only if it has a strong elimination ordering [9].

Algorithm ME

Input

A graph G with a strong elimination ordering $(v_1, v_2, \ldots, v_n)$, a labelling function $L(v) = (t(v), k(v))$ for each $v \in V(G)$

1. for $i = 1$ to n do:
2. $\quad f(v_i) := 0$ if $t(v_i) = A$, $f(v_i) := t(v_i)$ if $t(v_i) \neq A$;
3. end;
4. for $i = 1$ to n do:
5. $\quad$ if $f(N_G[v_i]) > k(v_i)$ stop, the problem is "infeasible";
6. end;
7. for $i = 1$ to n do
8. $\quad$ if $t(v_i) = A$, $M := \min\{k(v) - f(N_G[v]) : v \in N_G[v_i]\}$ and $f(v_i) := M$
9. end;
10. return function f.

Proposition 15. *Given a strongly chordal graph G with a strong elimination ordering* $(v_1, v_2, \ldots, v_n)$ *and a labelling function* $L(v) = (t(v), k(v))$ *defined on* $V(G)$*, Algorithm ME below decides if there exists an L-packing function f of G. In case such a function exists, ME returns one of maximum weight in time* $O(n + m)$*.*

Proof. Let f be the function that algorithm ME returns. Clearly, the function f at the beginning of the first iteration in Steps 7-9 is an L-packing function of G. Assume that the function f in the i-th iteration of Steps 7-9 is an L-packing function of G $(1 \leq i \leq n)$. For each $v \in N_G[v_i]$, $M \leq k(v) - f(N_G[v])$, then $f(N_G[v]) + M \leq k(v)$ and therefore the new function f obtained changing the value of $f(v_i)$ in Step 8 is still an L-packing function of G. Thus, the function that returns ME in Step 10 is an L-packing function of G.

Next we prove that the function returned from ME in Step 10 has maximum weight.

Let h be an L-packing function of G with maximum weight such that the size of $W = \{v \in V : f(v) \neq h(v)\}$ is minimum. Next we will show that $W = \emptyset$. Suppose that $W \neq \emptyset$ and let l be the minimum index such that $v_l \in W$. So $t(v_l) \neq A$ and $h(v_x) = f(v_x)$ if $x < l$.

Case 1. $h(v_l) > f(v_l)$. At the l-th iteration in Steps 7-9, $h(v_x) \geq 0 = f(v_x)$ for $x > l$. At Step 8, let $v_c \in N_G[v_l]$ such that $M = k_{v_c} - f(N_G[v_c])$. Before Step 8, since $f(v_l) = 0$, we have $f(N_G[v_c]) = f(N_G[v_c] - \{v_l\}) = k_{v_c} - M$. Then,

$$\begin{aligned} h(N_G[v_c]) = h(N_G[v_c] - \{v_l\}) + h(v_l) \geq f(N_G[v_c] - \{v_l\}) + h(v_l) \\ > f(N_G[v_c] - \{v_l\}) + f(v_l) \geq k_{v_c} - M + M = k_{v_c}, \end{aligned}$$

contradicting the fact that h is an L-packing function of G.

Case 2. $h(v_l) < f(v_l)$. Let $i = f(v_l)$, $j = h(v_l)$ and

$$P = \{v \in N_G[v_l] : h(N_G[v]) + i - j > k(v)\}.$$

We will prove that $P \neq \emptyset$. If $P = \emptyset$, $h(N_G[v]) + i - j \leq k(v)$ for all $v \in N_G[v_l]$. The function g defined by $g(v_l) := h(v_l) + i - j = i$ and $g(v) := h(v)$ in other case, is an L-packing function of G with $g(V(G)) > h(V(G))$, contradicting the fact that h is an L-packing function of G with maximum weight.

For each $v \in P$, $h(N_G[v]) + i - j > k(v) \geq f(N_G[v])$ thus $h(N_G[v]) - h(v_l) > f(N_G[v]) - f(v_l)$ and $h(N_G[v] - \{v_l\}) > f(N_G[v] - \{v_l\})$.

Then, for each $v \in P$,

$$N_G[v] \cap \{v_x \in W : l < x,\ h(v_x) > f(v_x)\} \neq \emptyset.$$

Let s be the minimum index among all the vertices in P and b be the minimum index among all the vertices in $N_G[v_s] \cap \{v_x \in W : l < x,\ h(v_x) > f(v_x)\}$.

There exist two positive integers c_1 and c_2 such that $f(v_l) = h(v_l) + c_1$ and $h(v_b) = f(v_b) + c_2$. Let $c = \min\{c_1, c_2\}$.

We define a function h' over $V(G)$ in the following way:

$$h'(v_l) = h(v_l) + c\,,\ \ h'(v_b) = h(v_b) - c \ \text{ and } \ h'(v) = h(v) \text{ in other case.}$$

Clearly $h(V(G)) = h'(V(G))$ and

$$|\{v \in V(G) : f(v) = h'(v)\}| \geq |\{v \in V(G) : f(v) = h(v)\}| + 1$$

since $h'(v_l) = h(v_l) + c_1 = f(v_l)$ if $c_1 \leq c_2$ and $h'(v_b) = h(v_b) - c_2 = f(v_b)$ if $c_1 > c_2$.

In order to prove that h' is an L-packing function of G, we show that $h'(N_G[v]) \leq k(v)$ for each $v \in N_G[v_l]$.

Observe that $P \subseteq N_G[v_l]$ and it can be verified that $P \subseteq N_G[v_b]$ in two cases:

- $s \leq l$. Then $s \leq l < b$. Since v_l and v_b belong to $N_{G_s}[v_s]$, by definition of strong elimination ordering, $N_{G_s}[v_l] \subseteq N_{G_s}[v_b]$. Since $P \subseteq N_{G_s}[v_l]$ then $P \subseteq N_G[v_b]$.
- $l < s$. Let $v_k \in P$, then $l < s \leq k$. Since v_s and v_k belong to $N_{G_l}[v_l]$, by definition of strong elimination ordering, $N_{G_l}[v_s] \subseteq N_{G_l}[v_k]$ for each $v \in P$. Since $v_b \in N_{G_l}[v_s]$, then $v_b \in N_{G_l}[v_k]$. In other words, $P \subseteq N_G[v_b]$.

If $v \in N_G[v_l]$ and $v \notin N_G[v_b]$, we have $h'(N_G[v]) = h'(N_G[v] - \{v_l\}) + h'(v_l) = h(N_G[v] - \{v_l\}) + h(v_l) + c = h(N_G[v]) + c \leq h(N_G[v]) + c_1 \leq k(v)$. Since $v \notin P$, $h(N_G[v]) + f(v_l) - h(v_l) \leq k(v)$.

If $v \in N_G[v_l] \cap N_G[v_b]$, then $h'(N_G[v]) = h'(N_G[v] - \{v_l, v_b\}) + h'(v_l) + h'(v_b) = h(N_G[v] - \{v_l, v_b\}) + h(v_l) + c + h(v_b) - c = h(N_G[v]) \leq k(v)$.

We conclude that h' is an L-packing function of G, such that $|\{v \in V : f(v) \neq h'(v)\}| < |\{v \in V : f(v) \neq h(v)\}|$, which contradicts the fact that $|W|$ is minimum.

The initialization in Steps 1-3 is done in $O(|V(G)|)$. At the i-th iteration in Steps 7-9, M can be calculated in $O(|N_G[v_i]|)$ verifying $k(v) - f(N_G[v])$ for each $v \in N_G[v_i]$. Therefore, the running time of algorithm ME is $O(\sum_{v \in V} |N_G[v]|) = O(n + m)$. $\square$

3. Further study: L-packing functions and clique-independence

In this section we show that the algorithm presented in the previous section can be applied to solve the clique-independence problem on a strongly chordal graph.

A *maximal clique* in a graph is a clique that is not properly contained in any other clique of the graph. A *clique-independent set* in a graph is a collection of pairwise vertex-disjoint maximal cliques.

Clique-independence problem:

Instance: A graph $G = (V, E)$ and an integer q.

Question: Is there a clique-independent set in G of size at least q?

Clique-independence problem is NP-complete on split graphs [1], k-tree graphs (with unbounded k) and undirected path graphs [2] and on complements of bipartite graphs [12].

In [1], Chang et al. present the clique-independence problem on strongly chordal graphs in linear time when a strong elimination order is given.

For a strongly chordal graph G, let $C(G) = \{Q_1, Q_2, \ldots, Q_p\}$ denote the set of all maximal cliques in G. The *vertex-clique incidence graph* $H(G)$ of G has vertex set $V(G) \cup X$, where $X = \{x_1, \ldots, x_n\}$ is an independent set and the adjacencies in $H(G)$ are given by the same adjacencies between vertices of $V(G)$ and, for $1 \leq i \leq p$, $x_i \in X$ is adjacent to $v_j \in V(G)$ if $v_j \in Q_i$.

Lemma 16. *(Guruswami and Pandu Rangan[12]) For a given strongly chordal graph G with $|V(G)| = n$ and $|E(G)| = m$, the vertex-clique incidence graph $H(G)$ is a strongly chordal graph and a strong elimination ordering of it can be obtained from a given one of G in $O(n + m)$-time.*

In the following we present a linear time reduction from the clique-independence problem on a strongly chordal graph to the labelled packing problem.

Theorem 17. *For a given strongly chordal graph G with $|V(G)| = n$ and $|E(G)| = m$ and $H(G)$ its vertex-clique incidence graph, the maximum size of a clique-independent set of G equals the maximum weight of an L-packing function of $H(G)$, where the labelling function L is defined on $V(H(G))$ as follows:*

$$L(v) = \begin{cases} (A, 1) & \textit{if} \quad v = x_i \\ (0, 1) & \textit{if} \quad v \in V(G). \end{cases}$$

Proof. Let P be a clique-independent set in G and set $H = H(G)$. We will define an L-packing function h of H with weight $h(V(H)) = |P|$.

Let

$$h(x_i) = \begin{cases} 1 & \text{if } Q_i \in P \\ 0 & \text{if not} \end{cases}$$

and $h(v) = 0$ for every $v \in V(G)$.

Then we have $h(N_H[x_i]) = h(x_i) \leq 1$ for $i \in \{1, \ldots, p\}$ and, since P is a clique-independent set in G, from the definition of h it cannot happen that $h(x_j) = h(x_r) = 1$ for $x_j, x_r \in N_H[v]$ and $j \neq r$; thus $h(N_H[v]) \leq 1$.

Now, let h be an L-packing function of H. Let us prove that $P = \{Q_i \in C(G) : h(x_i) = 1\}$ is a clique-independent set in G. If there exists $v \in V(G)$ such that $v \in Q_l \cap Q_s$ with $Q_l, Q_s \in P$ and $l \neq s$ then $h(N_H[v]) \geq h(x_l) + h(x_s) = 2$, which contradicts the definition of L-packing function for h. Clearly, $|P| = h(V(H))$. □

We conclude that the clique-independence problem in a strongly chordal graph G can be solved in linear time alternatively by Algorithm ME of the previous section.

4. Conclusions

In this paper we presented a generalized version —L-packing functions— of all limited packing concepts in graphs known from the literature up to now, namely, k-limited packings, $(\mathbf{k}, \mathcal{A})$-limited packings and $\{k\}$-packing functions. In particular, the linear time algorithm that solves the problem of finding the maximum weight of an L-packing function in strongly chordal graphs generalizes that in [13] for the $\{k\}$-packing function problem. Our work leaves open several related problems, including the study of the labelled packing problem beyond graph classes that are closed under the transformations in the reductions performed in this work. Also, it leaves open the question whether one can find a graph class where the labelled packing problem is "hard" to solve whereas at least one of the others is "tractable". Finally, it would also be interesting to follow the approach of Section 3 for strongly chordal graphs to extend the linearity of clique-independence to other graph classes, or from clique-independence to weighted clique-independence.

[1] Chang G. J., M. Farber, and Zs. Tuza, *Algorithmic aspects of neighbourhood numbers*, SIAM Journal on Discrete Mathematics **6** (1993), 24–29.

[2] Chang M. S., Y.-H. Chen, G. J. Chang, and J.-H. Yan, *Algorithmic aspects of the generalized clique-transversal problem on chordal graphs*, Discrete Applied Mathematics **66** (1966), 189–203.

[3] Clarke N., and R. Gallant, *On 2-limited packings of complete grid graphs*, Discrete Mathematics **340** (2017), 1705–1715.

[4] Damiand G., M. Habib, and C. Paul, *A simple paradigm for graph recognition: application to cographs and distance hereditary graphs*, Theoretical Computer Science **263** (2001) 99–111.

[5] Dobson, M. P., Leoni, V., and Nasini, G., *The k-limited packing and k-tuple domination problems in strongly chordal, P_4-tidy and split graphs*. Electronic Notes in Discrete Mathematics **36** (2010), pp. 559–566.

[6] Dobson M. P., V. Leoni, and G. Nasini, *The multiple domination and limited packing problems in graphs*, Information Proccesing Letters **111** (2011), 1108–1113.

[7] Dobson M. P., E. Hinrichsen, and V. Leoni, *Generalized limited packings of some graphs with a limited number of P_4-partners*. Theoretical Computer Science **579** (2015), 1–8.

[8] Dobson M. P., E. Hinrichsen, and V. Leoni, *On the complexity of the $\{k\}$-packing function problem*, International Transactions in Operational Research **24** (2017), 347–354.

[9] Farber M., *Characterizations of strongly chordal graphs*, Discrete Mathematics **43** (1983), 173–189.

[10] Gallant R., G. Gunther, B. Hartnell, and D. Rall, *Limited packing in graphs*, Discrete Applied Mathematics **158** (2010), 1357–1364.

[11] Golumbic M. C., and U. Rotics, *On the clique-width of perfect graph classes*, Lecture Notes in Computer Science **1665** (1999), 135–147.

[12] Guruswami V. and C. Pandu Rangan, *Algorithmc aspects of clique-transersal and clique-independent sets*, Discrete Applied Mathematics **100** (2000), 183–202.

[13] Hinrichsen E., and V. Leoni, *$\{k\}$-packing functions in graphs*, Lecture Notes in Computer Science **8596** (2014), 325–335.

[14] Lee C. M., and M. S. Chang, *Variations of Y-dominating functions on graphs*, Discrete Mathematics **308** (2008), 4185–4204.

[15] Leoni V., and N. Nasini, *Limited packing and multiple domination problems: polynomial time reductions*, Discrete Applied Mathematics **164** (2014) 547–553.