SOME PROPERTIES OF $\{k\}$-PACKING FUNCTION PROBLEM IN GRAPHS

JOZEF KRATICA, ALEKSANDAR SAVIĆ, and ZORAN MAKSIMOVIĆ*

Communicated by Ioan Tomescu

The recently introduced $\{k\}$-packing function problem is considered in this paper. Relationship between cases when $k = 1$, $k \geq 2$ and linear programming relaxation are introduced with sufficient conditions for optimality. For arbitrary simple connected graph G, we propose a construction procedure for finding values of k for which $L_{\{k\}}(G)$ can be determined in the polynomial time. Additionally, relationship between $\{1\}$-packing function and independent set number is established. Optimal values for some special classes of graphs and general upper and lower bounds are introduced, as well.

AMS 2020 Subject Classification: 05C69, 05C12.

Key words: $\{k\}$-packing function problem, independent set, dominating set, integer linear programming.

1. INTRODUCTION

1.1. Problem definition

In this paper, we will consider simple, finite and undirected graphs. For a given graph G, let $V(G)$ and $E(G)$ denote its vertex and edge sets, respectively. For any $v \in V(G)$, its open neighborhood $N_G(v)$ is the set of all vertices that are adjacent to v, and its closed neighborhood is $N_G[v] = N_G(v) \cup \{v\}$. For a function $f : V(G) \to \mathbb{N} \cup \{0\}$, and $A \subseteq V(G)$ the sum $\sum\limits_{v \in A} f(v)$ will be denoted as $f(A)$. Let $|V(G)| = n$ and $A_G = [a_{ij}]_{n \times n}$ where

$$a_{ij} = \begin{cases} 1, i = j \vee (i,j) \in E(G) \\ 0, \text{otherwise.} \end{cases}$$

For a graph G and a positive integer k, a function $f : V(G) \to \mathbb{N} \cup \{0\}$, is a $\{k\}$-packing function of graph G, if for each vertex $v \in V(G)$ value $f(N_G[v])$

---

*Corresponding author.

MATH. REPORTS **25(75)** (2023), *2*, 263–277
doi: 10.59277/mrar.2023.25.75.2.263

is at most k. The maximum possible value of $f(V(G))$ over all $\{k\}$-packing functions of graph G is denoted as $L_{\{k\}}(G)$. Formally,

$$L_{\{k\}}(G) = \max_{f:V(G)\to\mathbb{N}\cup\{0\}} \{f(V(G))|(\forall v \in V(G))f(N_G[v]) \leq k\}.$$

For a given graph G and $k \in \mathbb{N}$, a set of vertices $B \subseteq V(G)$ is called a k-limited packing in G if for all $v \in V(G)$ it holds that $|N[v] \cap B| \leq k$. The maximum size of k-limited packing of a graph G is denoted with $L_k(G)$.

The distance between vertices u and v, denoted as $d_G(u, v)$ is the length of the shortest $u - v$ path. The square of a graph G, named G^2, is the graph obtained from G by adding all edges between vertices from $V(G)$ that have a common neighbor, i.e. $G^2 = (V(G), E(G^2))$, where $E(G^2) = \{(u, v) \in V(G) \times V(G) \,|\, d_G(u, v) \leq 2\}$. The complement of a graph G, named $\overline{G}$, is defined as $\overline{G} = (V(G), \overline{E(G)})$, where $\overline{E(G)} = \{(u, v) \in V(G) \times V(G) \,|\, u \neq v \wedge (u, v) \notin E(G)\}$. The independent set $I(G)$ of a graph is a subset of $V(G)$, such that there are no edges between them, i.e. $(u, v \in I(G) \Rightarrow (u, v) \notin E(G))$. The independence number of a graph, named $\alpha(G)$ is the cardinality of a maximal independent set $I(G)$.

1.2. **Previous work**

For k being fixed positive integer, Meir and Moon [12] introduced k-packing set $P \subset V(G)$ as a set of vertices such that the distance between u and v is greater than k for distinct $u, v \in P$, and k-packing number ($\rho_k(G)$) as the number of vertices of such largest set. It stands that $\rho_1(G) = \alpha(G)$ is the independence number.

Gallant et al. in [7] introduced k-limited packing as a modification of packing number problem allowing that intersection of each closed neighborhood with a given set contains no more than k vertices. In [2, 3] Dobson et al. proved that k-limited packing is NP-complete for split and bipartite graphs. It was also shown that the problem (in the case when the input is a P_4 tidy graph) is solvable in polynomial time.

1.3. **Preliminaries and useful results**

The notion of $\{k\}$-packing function was introduced by Leoni and Hinrichsen [11] as a variation of k-limited packing in order to solve the problem of locating garbage dumps in a given city. In this scenario, it is possible to place more than one dump in a certain location, requesting that no more than k dumps are placed in each vertex and its neighborhood. Relationship between k-limited packing and $\{k\}$-packing function stating that $L_{\{k\}}(G) \geq L_k(G)$ is established in [10]. Additionally, in [11], it is shown that $L_{\{k\}}(G) = L_k(G \otimes K_k)$ ($\otimes$ is a strong product of graphs). Therefore, when $k = 1$ it holds $L_{\{1\}}(G) = L_1(G)$.

Three other useful propositions and one theorem are given bellow.

Proposition 1. ([4]) *For a graph G and a positive integer k it holds $L_{\{k\}}(G) \geq k \cdot L_1(G)$*

Proposition 2. ([13]) *For any connected graph G and integer $k \in \{1, 2\}$ $L_k(G) \geq \lceil \frac{k \cdot \mathrm{diam}(G) + k}{3} \rceil$*

Proposition 3. ([7, 5, 4]) *For path P_n holds $L_{\{k\}}(P_n) = \lceil \frac{n}{3} \rceil \cdot k$.*

The proposition directly holds from the facts that $L_1(P_n) = \lceil \frac{n}{3} \rceil$ ([7]), $\gamma(P_n) = \lceil \frac{n}{3} \rceil$ ([5]) and $\gamma(G) = L_1(G) \Rightarrow L_{\{k\}}(P_n) = k \cdot L_1(P_n)$ ([4]).

Theorem 1. [4] *The $\{k\}$-packing function problem is NP-complete for all integer k fixed.*

The polynomial equivalence between $\{k\}$-packing function problem and k-limited packing in graphs is discussed in [10].

2. NEW PROPERTIES OF $\{k\}$-PACKING FUNCTION PROBLEM

2.1. Relationship between $\{k\}$-packing and $\{1\}$-packing

In this section, we establish relationship between $\{k\}$-packing, $\{1\}$-packing problem and relaxation of $\{1\}$-packing, as well as some properties of $\{k\}$-packing function problem for certain classes of graphs. Without loss of generality, we assume that considered graphs are connected and have at least two vertices. This is because if the graph is not connected, we can consider connected components instead, using the following simple property.

Property 1. *If G is not connected and has connected components $\mathrm{Con}_1(G)$, $\mathrm{Con}_2(G)$, $\ldots \mathrm{Con}_{nc}(G)$ then $L_{\{k\}}(G) = \sum_{i=1}^{nc} L_{\{k\}}(\mathrm{Con}_i(G))$.*

Proof. Let $v \in V$ be an arbitrary vertex from a connected component $\mathrm{Con}_j(G)$. Since $v \in \mathrm{Con}_j(G) \Rightarrow N[v] \subseteq \mathrm{Con}_j(G)$, then all constraints $f(N_G[v]) \leq k$ can be grouped by connected components and considered independently. $\square$

Let $Z^*_{rlx}(G)$ be an optimal solution of the relaxed $\{1\}$-packing problem. Relaxation is performed by

$$Z^*_{rlx}(G) = \max_{f: V(G) \to [0, +\infty)} \{f(V(G)) | (\forall v \in V(G)) f(N_G[v]) \leq 1\},$$

i.e. relaxed packing function can take fractional (real) values.

Now we can formulate relationship among $L_{\{k\}}(G)$, $L_{\{1\}}(G)$ and $Z^*_{rlx}(G)$.

Proposition 4. *For arbitrary $k \in \mathbb{N}$ it stands that*

$$k \cdot L_{\{1\}}(G) \leq L_{\{k\}}(G) \leq k \cdot Z^*_{rlx}(G).$$

Proof. It should be noted that $L_{\{k\}}(G) \geq k \cdot L_{\{1\}}(G)$ directly follows from Proposition 1 and the fact that $L_{\{1\}}(G) = L_1(G)$.

Let $f_{rlx} : V(G) \to [0, +\infty)$ be a relaxed $\{1\}$-packing function with maximum value of all such functions. As it stands that

$$(\forall v \in V(G)) f_{rlx}(N_G[v]) \leq 1 \Rightarrow k \cdot f_{rlx}(N_G[v]) \leq k$$

and $\{k\}$-packing function has non negative integer values, then

$$L_{\{k\}}(G) = \max_{h:V(G)\to\mathbb{N}\cup\{0\}} \{h(V(G)) | (\forall v \in V(G)) h(N_G[v]) \leq k\}$$

which is less than or equal to

$$k \cdot \max_{f:V(G)\to[0,+\infty)} \{f(V(G)) | (\forall v \in V(G)) f(N_G[v]) \leq 1\} = k \cdot Z^*_{rlx}(G).$$

□

It is interesting to find when equalities hold, i.e. when $k \cdot L_{\{1\}}(G) = L_{\{k\}}(G)$ or $k \cdot L_{\{1\}}(G) = k \cdot Z^*_{rlx}(G)$. Sufficient condition for both equalities will be given in the following theorem.

Theorem 2. *If A_G is a totally unimodular matrix, then*

$$L_{\{k\}}(G) = k \cdot L_{\{1\}}(G) = k \cdot Z^*_r(G).$$

Proof. Let $G = (V, E)$ be a graph whose A_G is a totally unimodular matrix. Let us consider $\{k\}$-packing function problem. The problem can be formulated as a following integer linear program. Let us denote the variables x_i, $i = 1, \ldots, |V|$ such that $x_i = f(i)$. Then, $\{k\}$-packing function problem can be formulated as

$$\max \sum_{i=1}^{|V|} x_i \tag{1}$$

subject to

$$\sum_{j \in N_G[i]} x_j \leq k, \quad i = 1, \ldots, |V| \tag{2}$$

$$x_i \in \{0, 1, \ldots, k\}, \quad i = 1, \ldots, |V| \tag{3}$$

It is easy to see that condition $\sum\limits_{i \in N_G[j]} x_i \leq k$ could be replaced with $\sum\limits_{j=1}^{|V|} a_{ij} x_j \leq k$ where a_{ij} are elements of matrix A_G. Now, the formulation is

$$\max \sum_{i=1}^{|V|} x_i \tag{4}$$

subject to

$$\sum_{j=1}^{|V|} a_{ij} x_j \leq k, \quad i = 1, \ldots, |V| \tag{5}$$

$$x_i \in \{0, 1, \ldots, k\}, \quad i = 1, \ldots, |V|. \tag{6}$$

Since this is Integer Linear Programming (ILP) formulation, it is natural to consider its relaxation. Instead of integer constraint $x_i \in \{1, \ldots, k\}$, let us consider non-negativity constraint $x_i \geq 0$. From the first constraint, it is obvious that for every vertex i, $x_i \leq k$ will hold. Let us now consider linear programming formulation

$$\max \sum_{i=1}^{|V|} x_i \tag{7}$$

subject to

$$\sum_{j=1}^{|V|} a_{ij} x_j \leq k, \quad i = 1, \ldots, |V| \tag{8}$$

$$x_i \geq 0, \quad i = 1, \ldots, |V|. \tag{9}$$

Note that this formulation for $k = 1$ is exactly Linear Programming (LP) formulation of $Z^*_{rlx}(G)$:

$$\max \sum_{i=1}^{|V|} x_i \tag{10}$$

subject to

$$\sum_{j=1}^{|V|} a_{ij} x_j \leq 1, \quad i = 1, \ldots, |V| \tag{11}$$

$$x_i \geq 0, \quad i = 1, \ldots, |V|. \tag{12}$$

Since at least one feasible solution of the formulation above exists, $x_i = 0$, $i = 1, \ldots, |V|$, and all variables have upper bound, an optimal solution also exists. It is known from the theory of integer linear programming (Theorem 19.1 in [14]) that polyhedron $X(b)$, defined as $X(b) = \{x|Ax \geq b\}$ for any integer vector b, is an integer if and only if the matrix A is totally unimodular. Since polyhedron of relaxation of our problem is $X(b) = \{x|A_G x \leq k \cdot e_{|V|}\}$, where $e_{|V|} = (1, \ldots, 1)^T$ is vector of ones and dimension equal to $|V|$, has totally unimodular matrix A_G, it can be concluded that all of polyhedron nodes are integer. This means that all optimal solutions of the relaxation problem are integer. As ILP and LP formulations differ only in the condition of integrality, it can be concluded that optimal solutions of the relaxation and ILP formulation are the same under the conditions of this theorem.

We have proved that $L_{\{1\}}(G) = Z^*_{rlx}(G)$. From Proposition 4 which states that

$$k \cdot L_{\{1\}}(G) \leq L_{\{k\}}(G) \leq k \cdot Z^*_{rlx}(G)$$

and equality of the first and the third term directly holds

$$k \cdot L_{\{1\}}(G) = L_{\{k\}}(G) = k \cdot Z^*_{rlx}(G).$$

□

From the well-known fact that any LP problem has a polynomial complexity, the following assertion holds.

COROLLARY 1. *If A_G is a totally unimodular matrix, then $\{k\}$-packing function problem can be solved in polynomial time.*

However, total unimodularity of matrix A_G is not necessary condition for $k \cdot L_{\{1\}}(G) = L_{\{k\}}(G) = k \cdot Z^*_{rlx}(G)$ to hold, which is illustrated by the following example.

Example 1. Let graph G be a claw graph with four vertices, i.e. $G = (V, E)$, where $V = \{1, 2, 3, 4\}$ and $E = \{\{1, 2\}, \{1, 3\}, \{1, 4\}\}$. Matrix A_G is not totally unimodular since $det(A_G) = -2$. Since $N[1] = V(G)$, taking into consideration $L_{\{1\}}(G)$, we have $f(V(G)) = f(N[1]) \leq 1$. We can construct $\{1\}$-packing function f where $f(V(G)) = 1$: $f(1) = 1$ and $f(2) = f(3) = f(4) = 0$. It is obvious that constructed function f is also maximum $Z^*_{rlx}(G)$ of the relaxation problem. From the previous facts, clearly $L_{\{1\}}(G) = Z^*_{rlx}(G) = 1$. Because of Proposition 4, it holds $k \cdot L_{\{1\}}(G) = L_{\{k\}}(G) = k \cdot Z^*_{rlx}(G) = k$.

The following example illustrates the case when $k \cdot L_{\{1\}}(G) < L_{\{k\}}(G)$.

Example 2. Let us consider graph G given in Figure 1.

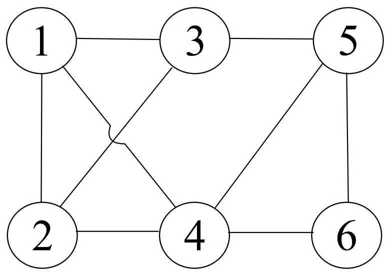

Figure 1 – An example of a graph G where $k \cdot L_{\{1\}}(G) < L_{\{k\}}(G)$.

For graph G presented in Figure 1, $2 \cdot L_{\{1\}}(G) = 2 < L_{\{2\}}(G) = 3$ holds, since values $L_{\{1\}}(G) = 1$ and $L_{\{2\}}(G) = 3$ are obtained by a total enumeration. For $k = 1$, $\{1\}$-packing function with maximal value is defined as follows: $f_1(1) = 1$; $f_1(2) = f_1(3) = f_1(4) = f_1(5) = f_1(6) = 0$. For $k = 2$, $\{2\}$-packing function with maximal value is defined as follows: $f_2(2) = f_2(3) = f_2(6) = 1$; $f_2(1) = f_2(4) = f_2(5) = 0$.

Next, we present an example where $L_{\{k\}}(G) < k \cdot Z_R^*(G)$.

Example 3. Let graph G be defined by $V(G) = \{1, 2, \ldots, 30\}$ and adjacency matrix A_G shown in Figure 2. For G presented in Figure 2, $L_{\{1\}}(G) = 1 < \lfloor Z_{rlx}^*(G) \rfloor = 2$ holds. Values $L_{\{1\}}(G) = 1$ can be obtained using ILP formulation (4)-(6), while $Z_{rlx}^*(G) = \frac{7}{3}$ can be obtained from relaxed LP formulation (10)-(12). Values of function f which correspond to $Z_{rlx}^*(G)$ are:
$f(3) = f(4) = \frac{1}{7}$;
$f(5) = f(6) = \frac{2}{21}$;
$f(8) = f(13) = f(20) = f(24) = \frac{4}{21}$;
$f(10) = f(16) = f(18) = \frac{5}{21}$;
$f(19) = \frac{1}{21}$ and $f(11) = \frac{1}{3}$.
For any other vertex v, $f(v) = 0$.

2.2. Relationship between $\{k\}$-packing and relaxation of $\{1\}$-packing

In the sequel, we will prove that equality $L_{\{k\}}(G) = k \cdot Z_{rlx}^*(G)$ holds for all graphs, but only for certain values of k.

THEOREM 3. *For an arbitrary graph G, $(\exists q \in \mathbb{N})(\forall k_1 \in \mathbb{N}) \quad L_{\{k_1 \cdot q\}}(G) = k_1 \cdot q \cdot Z_{rlx}^*(G)$.*

Proof. For an arbitrary graph G, let $(x_1^*, \ldots, x_n^*)$ be an optimal solution of linear programming formulation (10)-(12), with objective function value

$$
A_G = \begin{bmatrix}
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&0&0&0&1&1&1&0\\
0&0&0&1&1&1&0&0&0&0&0&1&0&0&1&1&0&1&0&0&1&0&1&1&1&0&0&0&0&0\\
0&0&0&0&1&0&0&1&0&0&0&1&0&0&0&0&1&1&0&0&0&1&0&1&0&1&0&0&1&0\\
0&1&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&1&0&1&0&0&0&0&0&0&0&1&0\\
1&1&1&0&0&0&1&0&1&0&0&0&1&1&1&0&1&1&0&0&0&1&1&1&0&0&0&1&0&1\\
0&1&0&0&0&0&0&1&1&0&1&0&1&1&0&0&0&0&0&1&0&0&1&0&1&0&0&0&0&1\\
0&0&0&1&1&0&0&0&1&0&1&0&1&0&1&0&1&0&1&0&0&0&0&1&0&1&0&0&1&1\\
0&0&1&0&0&1&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&1&1&1&0&0&1&1&0&0&1&1&0&0&0&1&1&1&0&1&0&1&0&1&0&1&0\\
0&0&0&0&0&0&0&0&1&0&0&0&1&0&0&1&0&0&1&1&0&0&0&0&0&0&1&0&1&0\\
0&0&0&0&0&1&1&0&1&0&0&0&1&0&0&0&0&1&0&0&0&1&1&0&0&0&1&0&0&0\\
0&1&1&0&0&0&0&0&0&0&0&0&1&0&1&0&1&1&1&0&1&0&1&1&1&1&0&0&0&0\\
0&0&0&0&1&1&1&0&0&1&1&1&0&0&0&0&0&0&1&0&0&0&0&0&0&1&1&0&0&1\\
0&0&0&0&1&1&0&1&1&0&0&0&0&0&0&0&1&1&0&1&1&0&1&1&1&0&0&0&1&0\\
0&1&0&1&1&0&1&0&1&0&0&1&0&0&0&1&0&1&1&1&0&1&1&0&1&0&0&0&1&1\\
0&1&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&1&0&1&0&1&1&0&1&1&1\\
0&0&1&0&1&0&1&1&0&0&0&1&0&1&0&0&0&1&0&1&1&0&1&0&1&0&1&1&1&1\\
1&1&1&0&1&0&0&0&0&0&1&1&0&1&1&0&1&0&0&0&1&1&0&1&1&1&1&0&0&0\\
1&0&0&1&0&0&1&0&1&1&0&1&1&0&1&0&0&0&0&1&0&1&1&1&0&0&0&1&1&0\\
1&0&0&0&0&1&0&0&1&1&0&0&0&1&1&1&1&0&1&0&1&0&1&1&0&1&0&1&0&1\\
1&1&0&1&0&0&0&0&1&0&0&1&0&1&0&1&1&1&0&1&0&1&1&1&1&0&0&1&1&0\\
1&0&1&0&1&0&0&0&0&0&1&0&0&0&1&0&0&1&1&0&1&0&1&0&1&0&1&0&1&1\\
1&1&0&0&1&1&0&0&1&0&1&1&0&1&1&1&1&0&1&1&1&1&0&0&1&1&1&1&1&1\\
0&1&1&0&1&0&1&0&0&0&0&1&0&1&0&0&0&1&1&1&1&0&0&0&1&0&0&1&0&1\\
0&1&0&0&0&1&0&0&1&0&0&1&0&1&1&1&1&1&0&0&1&1&1&1&0&0&1&1&1&1\\
0&0&1&0&0&0&1&0&0&0&0&1&1&0&0&1&0&1&0&1&0&0&1&0&0&0&1&1&1&0\\
1&0&0&0&0&0&0&0&1&1&1&0&1&0&0&0&1&1&0&0&0&1&1&0&1&1&0&1&1&1\\
1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&1&0&1&1&1&0&1&1&1&1&1&0&1&1\\
1&0&1&1&0&0&1&1&1&1&0&0&0&1&1&1&1&0&1&0&1&1&1&0&1&1&1&1&0&1\\
0&0&0&0&1&1&1&0&0&0&0&0&1&0&1&1&1&0&0&1&0&1&1&1&1&0&1&1&1&0
\end{bmatrix}
$$

Figure 2 – An example of a graph G where $L_{\{k\}}(G) < \lfloor k \cdot Z^*_{rlx}(G) \rfloor$

$Z^*_{rlx}(G)$. Since constraint matrix A_G is an integer matrix and right-hand side vector $b = (1\,1 \ldots 1)^T$ is also the integer vector, then each feasible solution must be a vector with rational coordinates. Therefore, it also holds for optimal solution, i.e. $(\forall i)(x^*_i = \frac{p_i}{q_i}$ where $p_i \in \mathbb{Z}$, $q_i \in \mathbb{N}$ and $gcd(p_i, q_i) = 1$ where $gcd(a, b)$ is the greatest common divisor of a and b. Let us introduce $q = lcm(q_1, \ldots, q_n)$ where lcm is the least common multiple. From the definition it is obvious that $q_1, \ldots q_n \in \mathbb{N} \Rightarrow q \in \mathbb{N}$. If $x^*_i = 0$ then $p_i = 0$, let fix $q_i = 1$ in that case. If (10)-(12) has multiple optimal solutions we will assume that we can arbitrarily choose one of them.

Let $k = k_1 \cdot q$ and let $(y^*_1, \ldots, y^*_n)$ denote optimal solution of the dual problem of the linear programming formulation (10)-(12). This solution satisfies $A_G \cdot (y^*_1 \ldots y^*_n)^T \geq (1\ 1 \ldots 1)^T$. Since $(x^*_1, \ldots, x^*_n)$ and $(y^*_1, \ldots, y^*_n)$ are optimal solutions of the mutually dual problems it follows that values of corresponding objective functions are equal, that is $\sum\limits_{i=1}^{n} x^*_i = \sum\limits_{i=1}^{n} y^*_i$. Dual problem of the problem (7)-(9) is

$$\max \sum_{i=1}^{|V|} k \cdot Y_i = k \cdot \sum_{i=1}^{|V|} Y_i \tag{13}$$

subject to

$$(14) \qquad \sum_{i=1}^{|V|} a_{ij} Y_i \geq 1, \quad j = 1, \ldots, |V|$$

$$(15) \qquad Y_i \geq 0, \quad i = 1, \ldots, |V|.$$

As it can be seen, the value of objective function is k times the value of objective function of the dual of problem (10)-(12). Now, it can be concluded that the optimal value of objective function (7) is equal to $k \cdot \sum_{i=1}^{n} x_i^*$ and consequently that $(k \cdot x_1^*, \ldots, k \cdot x_n^*)$ is the optimal solution of linear programming formulation (7)-(9). As $k = q \cdot k_1$, such that $q = lcm(q_1, \ldots, q_n)$ and $(\forall i) x_i^* = \frac{p_i}{q_i}$ then $k_1 \cdot q \cdot x_i^* = k_1 \cdot q \cdot \frac{p_i}{q_i} \in \mathbb{Z}$. Since $(k_1 \cdot q \cdot x_1^*, \ldots, k_1 \cdot q \cdot x_n^*)$ is a vector of integers, and it is optimal solution of linear programming formulation (7)-(9) then it is also optimal solution of integer linear programming formulation (4)-(6) with the optimal value $k_1 \cdot q \cdot Z_{rlx}^*$. Therefore, $L_{\{k_1 \cdot q\}}(G) = k_1 \cdot q \cdot Z_{rlx}^*$, which confirms the statement of the theorem. $\square$

Corollary 2. $\overline{\lim\limits_{k \to +\infty}} \frac{L_{\{k\}}(G)}{k} = Z_{rlx}^*(G)$.

Proof. For a given graph G let us consider sequence $(L_{\{k\}}(G))_{k \in \mathbb{N}}$ and its subsequence $(L_{\{l \cdot q\}}(G))_{l \in \mathbb{N}}$ and $q \in \mathbb{N}$ as defined in Theorem 3. From Property 4 it follows that

$$(\forall k) L_{\{k\}}(G) \leq k \cdot Z_{rlx}^*(G)$$

implying

$$(\forall k) \frac{L_{\{k\}}(G)}{k} \leq Z_{rlx}^*(G).$$

For subsequence $(L_{\{l \cdot q\}}(G))_{l \in \mathbb{N}}$ from Theorem 3 it holds

$$(\forall l) L_{\{l \cdot q\}}(G) = l \cdot q \cdot Z_{rlx}^*(G),$$

so

$$(\forall l) \frac{L_{\{l \cdot q\}}(G)}{l \cdot q} = Z_{rlx}^*(G),$$

implying $\overline{\lim\limits_{l \to +\infty}} \frac{L_{\{l \cdot q\}}(G)}{l \cdot q} = Z_{rlx}^*(G)$, which directly confirms the statement. $\square$

Corollary 3. *For any graph G there exists $q \in \mathbb{N}$ such that $L_{\{k_1 \cdot q\}}(G)$ can be found in polynomial time for any $k_1 \in \mathbb{N}$.*

Proof. Let us consider q as defined in Theorem 3. If $k = q \cdot q_1$ then by Theorem 3, optimal solution of $L_{\{k\}}(G)$ can be obtained as optimal solution of linear programming formulation (7)-(9). Since it can be achieved in polynomial time, then in this case $L_{\{k\}}(G)$ can be obtained in polynomial time. $\square$

Observation 1. It should be noted that in Theorem 1 ([4]) the word "fixed" is necessary. Although for each simple connected graph G and for some values of k, $L_{\{k\}(G)}$ it can be determined in polynomial time, the considered problem is still NP-complete for k fixed.

Observation 2. It should be noted that q defined in Theorem 3 is not necessarily minimal in the case with multiple optimal solution of (10)-(12). The number of optimal solutions can be, in the worst case, infinite (even uncountable), though they all have the same optimal value, the minimal value of q defined in Theorem 3 may not be obtained in polynomial time.
Even in the case with single optimal solution of (10)-(12), $q = lcm(q_1, \cdots q_n)$ may not be the minimal k for which (10)-(12) has integer optimal solution.

2.3. New properties of $\{1\}$-packing function

Previous considerations were based on the Integer Linear Programming formulation of the proposed problem and its relaxation. Now, let us present several properties of $\{k\}$-packing function problem which are not derived from ILP formulation. In the following proposition, it will be proven that $\{1\}$-packing function problem of an arbitrary graph G can be reduced to vertex independence number problem on a graph G^2.

PROPOSITION 5. $L_{\{1\}}(G) = \alpha(G^2)$.

Proof. ($\Rightarrow$) Let f be a 1-packing function whose value $f(V(G)) = L_{\{1\}}(G)$. We define $I = \{v \in V(G) \,|\, f(v) = 1\}$. Let $u, v \in V(G)$, $u \neq v$ and $(u, v) \in E(G^2)$, i.e. $d(u, v) \leq 2$. Then we have two cases:

CASE 1: $v \in N(u)$. Since f is 1-packing function then

$$f(N[u]) = \sum_{v \in N[u]} f(v) \leq 1$$

implying $f(u) + f(v) \leq 1$.

CASE 2: $u, v \in N(w)$. Since f is 1-packing function then

$$f(N[w]) = \sum_{v \in N[w]} f(v) \leq 1$$

implying $f(u) + f(v) \leq 1$.

In both cases, we have $f(u) + f(v) \leq 1$ implying that $(u \notin I \vee v \notin I)$. Since for each edge from $E(G^2)$ it has at least one endpoint in I, then I is the independent set of G^2.

($\Leftarrow$) Let I be an independent set of G^2. We define $f(v) = \begin{cases} 1, & v \in I \\ 0, & v \notin I \end{cases}$.

Let v be an arbitrary vertex from $V(G)$, and $u, w \in N(v)$ and $u \neq w$. Then, $d(u,w) \leq 2$. Since I is an independent set of G^2 at most one of vertices u, w is in I, so $f(u) + f(v) + f(w) \leq 1$. Since u and w are arbitrary vertices from $N(v)$, then

$$f(N[v]) = \sum_{w \in N[v]} f(w) \leq 1.$$

In the case when v has only one neighbor u, it holds that $f(N[v]) = f(u) + f(v) \leq 1$. Since v is an arbitrary vertex from $V(G)$ it follows that f is 1-packing function of G. □

Corollary 4. *$L_{\{1\}}(G) = \rho_2(G)$.*

Corollary 5. *If* $\text{diam}(G) = 2$, *then* $L_{\{1\}}(G) = 1$.

Proof. If $\text{diam}(G) = 2$, then $G^2 = K_{|V(G)|}$, and consequently, $L_{\{1\}}(G) = \alpha(K_{|V(G)|}) = 1$. □

Next, we propose computationally efficient lower bound based on the graph diameter.

Proposition 6. *$L_{\{k\}}(G) \geq \lceil \frac{1+\text{diam}(G)}{3} \rceil \cdot k$.*

Proof. From Proposition 2 it stands that $L_1(G) \geq \lceil \frac{\text{diam}(G)+1}{3} \rceil$. On the other hand, from Proposition 1 it stands that $L_{\{k\}} \geq k \cdot L_1$. By combining the mentioned inequalities, we obtain

$$L_{\{k\}} \geq k \cdot L_1 \geq k \cdot \lceil \frac{\text{diam}(G)+1}{3} \rceil.$$

□

This lower bound is tight as it can be seen from Proposition 3.

Next, we introduce the upper bound based on the vertices' degree.

Proposition 7. *$L_{\{k\}}(G) \leq \lfloor \frac{nk}{1+\delta(G)} \rfloor$.*

Proof. For each vertex $v \in V(G)$ it holds that $f(N[v]) \leq k$. Summing previous inequalities over all vertices from V we obtain:

$$n \cdot k \geq \sum_{v \in V} f(N[v]) = \sum_{v \in V} \sum_{w \in N[v]} f(w).$$

On the other hand, for arbitrary vertex u from V, $f(u)$ appears exactly $1 + deg(u)$ times in previous sums: once for the vertex u and $deg(u)$ times for each vertex that is adjacent to the vertex u. Therefore, we get:

$$\sum_{v\in V}\sum_{w\in N[v]} f(w) = \sum_{u\in V}(1+deg(u))\cdot f(u) \geq \sum_{u\in V}(1+\delta)\cdot f(u) =$$

$$= (1+\delta)\cdot\sum_{u\in V} f(u) = (1+\delta)\cdot f(V(G)).$$

As a consequence, the following holds

$$f(V(G)) \leq \frac{n\cdot k}{1+\delta} \Rightarrow f(V(G)) \leq \left\lfloor\frac{n\cdot k}{1+\delta}\right\rfloor.$$

This inequality holds because $f(V(G)) \in \mathbb{N}\cup\{0\}$. □

COROLLARY 6. *If G is a regular graph of degree r, then $L_{\{k\}}(G) \leq \lfloor\frac{nk}{1+r}\rfloor$.*

Bounds in Proposition 7 are tight as it can be seen from the two following statements.

PROPERTY 2. *For complete graph (clique) K_n holds $L_{\{k\}}(K_n) = k$.*

PROPOSITION 8. *For cycle C_n it holds that $L_{\{k\}}(C_n) = \lfloor\frac{n\cdot k}{3}\rfloor$.*

Proof. Let graph C_n be a cycle, i.e. $C_n = (V, E)$ where

$$V = \{0, 1, 2, \ldots, n-1\} \text{ and}$$

$$E = \{\{0,1\},\{1,2\},\{2,3\},\ldots,\{n-2,n-1\},\{n-1,0\}\}.$$

Let us define a function f as follows

$$f(v_i) = \begin{cases} \lfloor\frac{k}{3}\rfloor, & i\equiv 0 \pmod 3,\\ \lfloor\frac{k}{3}+0.5\rfloor, & i\equiv 1 \pmod 3,\\ \lceil\frac{k}{3}\rceil, & i\equiv 2 \pmod 3.\end{cases}$$

All possible cases are presented in Table 1.

It is obvious that in each case $f(N[w]) \leq k$ and $f(V(G)) = \lfloor\frac{n\cdot k}{3}\rfloor$. Therefore, we proved that $L_{\{k\}}(C_n) \geq \lfloor\frac{n\cdot k}{3}\rfloor$. Since C_n is a regular graph with $r = 2$ it holds that

$$L_{\{k\}}(G) \leq \lfloor\frac{nk}{1+2}\rfloor = \lfloor\frac{nk}{3}\rfloor.$$

Consequently, equality $L_{\{k\}}(G) = \lfloor\frac{nk}{3}\rfloor$ holds. □

Table 1 – $f(N[v])$ for C_n

n	k	v	$f(N[v])$
$3m$	$3l$	$v_i, i = 0, \ldots, 3m-1$	$\lfloor \frac{3l}{3} \rfloor + \lfloor \frac{3l}{3} + 0.5 \rfloor + \lceil \frac{3l}{3} \rceil = l + l + l = 3l = k \leq k$
$f(V(C_{3m})) = m \cdot 3l = \lfloor \frac{nk}{3} \rfloor$			
$3m$	$3l+1$	$v_i, i = 0, \ldots, 3m-1$	$\lfloor \frac{3l+1}{3} \rfloor + \lfloor \frac{3l+1}{3} + 0.5 \rfloor + \lceil \frac{3l+1}{3} \rceil = l + l + l + 1 = 3l + 1 = k \leq k$
$f(V(C_{3m})) = m \cdot (3l+1) = \lfloor \frac{nk}{3} \rfloor$			
$3m$	$3l+2$	$v_i, i = 0, \ldots, 3m-1$	$\lfloor \frac{3l+2}{3} \rfloor + \lfloor \frac{3l+2}{3} + 0.5 \rfloor + \lceil \frac{3l+2}{3} \rceil = l + l + 1 + l + 1 = 3l + 2 = k \leq k$
$f(V(C_{3m})) = m \cdot (3l+2) = \lfloor \frac{nk}{3} \rfloor$			
$3m+1$	$3l$	v_0	$\lfloor \frac{3l}{3} \rfloor + \lfloor \frac{3l}{3} \rfloor + \lfloor \frac{3l}{3} + 0.5 \rfloor = l + l + l = 3l = k \leq k$
$3m+1$	$3l$	$v_i, i = 1, \ldots, 3m-1$	$\lfloor \frac{3l}{3} \rfloor + \lfloor \frac{3l}{3} + 0.5 \rfloor + \lceil \frac{3l}{3} \rceil = l + l + l = 3l = k \leq k$
$3m+1$	$3l$	v_{3m}	$\lceil \frac{3l}{3} \rceil + \lfloor \frac{3l}{3} \rfloor + \lfloor \frac{3l}{3} \rfloor = l + l + l = 3l = k \leq k$
$f(V(C_{3m+1})) = (m+1) \cdot l + ml + ml = 3ml + l = l(3m+1) = \lfloor \frac{(3m+1)3l}{3} \rfloor = \lfloor \frac{nk}{3} \rfloor$			
$3m+1$	$3l+1$	v_0	$\lfloor \frac{3l+1}{3} \rfloor + \lfloor \frac{3l+1}{3} \rfloor + \lfloor \frac{3l+1}{3} + 0.5 \rfloor = l + l + l = 3l = k - 1 \leq k$
$3m+1$	$3l+1$	$v_i, i = 1, \ldots, 3m-1$	$\lfloor \frac{3l+1}{3} \rfloor + \lfloor \frac{3l+1}{3} + 0.5 \rfloor + \lceil \frac{3l+1}{3} \rceil = l + l + l + 1 = 3l + 1 = k \leq k$
$3m+1$	$3l+1$	v_{3m}	$\lceil \frac{3l+1}{3} \rceil + \lfloor \frac{3l+1}{3} \rfloor + \lfloor \frac{3l+1}{3} \rfloor = l + 1 + l + l = 3l + 1 = k \leq k$
$f(V(C_{3m+1})) = (m+1) \cdot l + ml + m(l+1) = 3ml + m + l = \lfloor \frac{(3m+1)(3l+1)}{3} \rfloor = \lfloor \frac{nk}{3} \rfloor$			
$3m+1$	$3l+2$	v_0	$\lfloor \frac{3l+2}{3} \rfloor + \lfloor \frac{3l+2}{3} \rfloor + \lfloor \frac{3l+2}{3} + 0.5 \rfloor = l + l + l + 1 = 3l + 1 = k - 1 \leq k$
$3m+1$	$3l+2$	$v_i, i = 1, \ldots, 3m-1$	$\lfloor \frac{3l+2}{3} \rfloor + \lfloor \frac{3l+2}{3} + 0.5 \rfloor + \lceil \frac{3l+2}{3} \rceil = l + l + 1 + l + 1 = 3l + 2 = k \leq k$
$3m+1$	$3l+2$	v_{3m}	$\lceil \frac{3l+2}{3} \rceil + \lfloor \frac{3l+2}{3} \rfloor + \lfloor \frac{3l+2}{3} \rfloor = l + 1 + l + l = 3l + 1 = k - 1 \leq k$
$f(V(C_{3m+1})) = (m+1) \cdot l + m(l+1) + m(l+1) = 3ml + l + 2m = \lfloor \frac{(3m+1)(3l+2)}{3} \rfloor = \lfloor \frac{nk}{3} \rfloor$			
$3m+2$	$3l$	v_0	$\lfloor \frac{3l}{3} + 0.5 \rfloor + \lfloor \frac{3l}{3} \rfloor + \lfloor \frac{3l}{3} + 0.5 \rfloor = l + l + l = 3l = k \leq k$
$3m+2$	$3l$	$v_i, i = 1, \ldots, 3m$	$\lfloor \frac{3l}{3} \rfloor + \lfloor \frac{3l}{3} + 0.5 \rfloor + \lceil \frac{3l}{3} \rceil = l + l + l = 3l = k \leq k$
$3m+2$	$3l$	v_{3m+1}	$\lfloor \frac{3l}{3} \rfloor + \lfloor \frac{3l}{3} + 0.5 \rfloor + \lfloor \frac{3l}{3} \rfloor = l + l + l = 3l = k \leq k$
$f(V(C_{3m+2})) = (m+1) \cdot l + (m+1) \cdot l + m \cdot l = 3ml + 2l = \lfloor \frac{(3m+2)3l}{3} \rfloor = \lfloor \frac{nk}{3} \rfloor$			
$3m+2$	$3l+1$	v_0	$\lfloor \frac{3l+1}{3} + 0.5 \rfloor + \lfloor \frac{3l+1}{3} \rfloor + \lfloor \frac{3l+1}{3} + 0.5 \rfloor = l + l + l = 3l = k - 1 \leq k$
$3m+2$	$3l+1$	$v_i, i = 1, \ldots, 3m$	$\lfloor \frac{3l+1}{3} \rfloor + \lfloor \frac{3l+1}{3} + 0.5 \rfloor + \lceil \frac{3l+1}{3} \rceil = l + l + l + 1 = 3l + 1 = k \leq k$
$3m+2$	$3l+1$	v_{3m+1}	$\lfloor \frac{3l+1}{3} \rfloor + \lfloor \frac{3l+1}{3} + 0.5 \rfloor + \lfloor \frac{3l+1}{3} \rfloor = l + l + l = 3l = k - 1 \leq k$
$f(V(C_{3m+2})) = (m+1) \cdot l + (m+1) \cdot l + m(l+1) = 3ml + m + 2l = \lfloor \frac{(3m+2)(3l+1)}{3} \rfloor = \lfloor \frac{nk}{3} \rfloor$			
$3m+2$	$3l+2$	v_0	$\lfloor \frac{3l+2}{3} + 0.5 \rfloor + \lfloor \frac{3l+2}{3} \rfloor + \lfloor \frac{3l+2}{3} + 0.5 \rfloor = l+1+l+l+1 = 3l+2 = k \leq k$
$3m+2$	$3l+2$	$v_i, i = 1, \ldots, 3m$	$\lfloor \frac{3l+2}{3} \rfloor + \lfloor \frac{3l+2}{3} + 0.5 \rfloor + \lceil \frac{3l+2}{3} \rceil = l + l + 1 + l + 1 = 3l + 2 = k \leq k$
$3m+2$	$3l+2$	v_{3m+1}	$\lfloor \frac{3l+2}{3} \rfloor + \lfloor \frac{3l+2}{3} + 0.5 \rfloor + \lfloor \frac{3l+2}{3} \rfloor = l + l + 1 + l = 3l + 1 = k - 1 \leq k$
$f(V(C_{3m+2})) = (m+1) \cdot l + (m+1)(l+1) + m(l+1) = 3ml + 2l + 2m + 1 = \lfloor \frac{(3m+2)(3l+2)}{3} \rfloor = \lfloor \frac{nk}{3} \rfloor$			

3. CONCLUSIONS

In this paper, the $\{k\}$-packing function problem is studied. First, the special relation was established between the cases when $k = 1$, $k \geq 2$, and the optimal solution of the linear programming relaxation. Second, sufficient conditions for optimality were introduced. It was proven that, for an arbitrary simple connected graph G and some values of k, $L_{\{k\}}(G)$ can be determined in the polynomial time. Next, $\{1\}$-packing function problem was studied and its connection with the independent set number and 2-packing problem. Finally, lower and upper bound were introduced as well as optimal values for some special classes of graphs.

The future work could be directed to the problem of finding the $\{k\}$-packing function number of some graph products.

Acknowledgments. This work is partially supported by the Ministry of Science, Technology and Development, Republic of Serbia, grants 174010 and 174033.

REFERENCES

[1] A. Brandstädt, V.D. Chepoi, and F.F. Dragan, *The algorithmic use of hypertree structure and maximum neighbourhood orderings*. In: E.W. Mayr et al. (Eds.), *International Workshop on Graph-Theoretic Concepts in Computer Science*. Lecture Notes in Comput. Sci. **903** (1994), 65–80.

[2] M.P. Dobson, V. Leoni, and G. Nasini, *The k-limited packing and k-tuple domination problems in strongly chordal, p 4-tidy and split graphs*. Electron. Notes Discrete Math. **36** (2010), 559–566.

[3] M.P. Dobson, V. Leoni, and G. Nasini, *The multiple domination and limited packing problems in graphs*. Inform. Process. Lett. **111** (2011), *23*, 1108–1113.

[4] P. Dobson, E. Hinrichsen, and V. Leoni, *On the complexity of the $\{k\}$-packing function problem*. Int. Trans. Oper. Res. **24** (2017), 347–354.

[5] A. Frendrup, M.A. Henning, B. Randerath, and P.D. Vestergaard, *An upper bound on the domination number of a graph with minimum degree 2*. Discrete Math. **309** (2009), *4*, 639–646.

[6] M. Gairing, S.T. Hedetniemi, P. Kristiansen, and A.A. McRae, *Self-stabilizing algorithms for $\{k\}$-domination*. In: S.T. Huang and T. Herman (Eds.), *Symposium on Self-Stabilizing Systems*. Lecture Notes in Comput. Sci. **2704** (2003), 49–60.

[7] R. Gallant, G. Gunther, B.L. Hartnell, and D.F. Rall, *Limited packings in graphs*. Discrete Appl. Math. **158** (2010), *12*, 1357–1364.

[8] R.M. Karp, *Reducibility among combinatorial problems*. In: R.E. Miller et al. (Eds.), *Complexity of computer computations*. Proceedings of a symposium on the complexity of computer computations. Plenum Press, New York, London, 1972, pp. 85–103.

[9] V. Leoni, M.P. Dobson, and E. Hinrichsen, *NP-completeness of the $\{k\}$-packing function problem in graphs*. Electron. Notes Discrete Math. **50** (2015), 115–120.

[10] V. Leoni and M.P. Dobson, *Towards a Polynomial Equivalence Between $\{k\}$-Packing Functions and k-Limited Packings in Graphs*. In: R. Cerulli et al. (Eds.), Fourth International Symposium, ISCO 2016. Lecture Notes in Comput. Sci. **9849** (2016), 160–165.

[11] V. Leoni and E. Hinrichsen, *k-Packing Functions of Graphs*. In: P. Fouilhoux et al. (Eds.), Third International Symposium, ISCO 2014. Lecture Notes in Comput. Sci. **8596** (2014), 325–335.

[12] A. Meir and J. Moon, *Relations between packing and covering numbers of a tree*. Pacific J. Math. **61** (1975), *1*, 225–233.

[13] D.A. Mojdeh and B. Samadi, *Packing parameters in graphs: New bounds and a solution to an open problem*. J. Comb. Optim. **38** (2019), *3*, 739-747. arXiv:1705.08667.

[14] A. Schrijver, *Theory of Linear and Integer Programming*. Wiley-Interscience Series in Discrete Mathematics, John Wiley & Sons, Chichester, 1986.

Received October 24, 2019

Jozef J. Kratica
Serbian Academy of Sciences and Arts
Mathematical Institute
Kneza Mihaila 36/III, 11000 Belgrade, Serbia
jkratica@mi.sanu.ac.rs

Aleksandar Lj. Savić
University of Belgrade
Faculty of Mathematics
Studentski trg 16/IV, 11000 Belgrade, Serbia
asavic@matf.bg.ac.rs

Zoran Lj. Maksimović
Military Academy
University of Defence
Generala Pavla Jurišića Šturma 33,
11000 Belgrade, Serbia
zoran.maksimovic@gmail.com