Results on the domination number and the total domination number of Lucas cubes*

Zülfükar Saygı

Department of Mathematics, TOBB University of Economics and Technology, Turkey

Received 27 June 2019, accepted 2 July 2020

Abstract

Lucas cubes are special subgraphs of Fibonacci cubes. For small dimensions, their domination numbers are obtained by direct search or integer linear programming. For larger dimensions some bounds on these numbers are given. In this work, we present the exact values of total domination number of small dimensional Lucas cubes and present optimization problems obtained from the degree information of Lucas cubes, whose solutions give better lower bounds on the domination numbers and total domination numbers of Lucas cubes.

Keywords: Lucas cube, Fibonacci cube, domination number, total domination number, integer linear programming.

Math. Subj. Class.: 05C69, 68R10, 11B39

1 Introduction

Fibonacci cubes and Lucas cubes are special subgraphs of the hypercube graph, which are introduced as an alternative model for interconnection networks [6, 11]. The structural and enumerative properties of these graphs are considered from various point of view in the literature [2, 6, 7, 8, 9, 10, 11, 15].

Let $Q_n$ denote the hypercube of dimension $n \geq 1$. It is the graph with vertex set represented by all binary strings of length $n$ and two vertices in $Q_n$ are adjacent if they differ in one coordinate. For convenience $Q_0 = K_1$. *Fibonacci strings* of length $n$ are defined as the binary strings $b_1b_2\ldots b_n$ such that $b_i \cdot b_{i+1} = 0$ for all $i = 0, 1, \ldots, n-1$, that is, binary strings of length $n$ not containing two consecutive 1s. Using this representation $n$ dimensional *Fibonacci cube* $\Gamma_n$ is defined as the subgraph of $Q_n$ induced by the vertices

*The author is grateful to the anonymous reviewers for their valuable comments and careful reading and would like to thank Ömer EŞecioğlu and Elif Saygı for useful discussions and suggestions.

E-mail address: zsaygi@etu.edu.tr (Zülfükar Saygı)

©© This work is licensed under https://creativecommons.org/licenses/by/4.0/
whose string representations are Fibonacci strings. Lucas strings of length $n$ are defined as the Fibonacci strings $b_1 b_2 \ldots b_n$ such that $b_1 \cdot b_n = 0$. Similar to the Fibonacci cubes $n$ dimensional Lucas cube $\Lambda_n$ is defined as the subgraph of $\Gamma_n$ induced by the vertices whose string representations are Lucas strings.

Let $G = (V, E)$ be a graph with vertex set $V$ and edge set $E$. $D \subseteq V$ is called a dominating set of $G$ if every vertex in $V$ either belongs to $D$ or is adjacent to some vertex in $D$. Then the domination number $\gamma(G)$ of $G$ is defined as the minimum cardinality of a dominating set of the graph $G$. Similarly, $D \subseteq V$ is called a total dominating set of a graph $G$ without isolated vertex if every vertex in $V$ is adjacent to some vertex in $D$ and the total domination number $\gamma_t(G)$ of $G$ is defined as the minimum cardinality of a total dominating set of $G$. The domination numbers of $\Gamma_n$ and $\Lambda_n$ are first considered in [2, 12]. Using integer linear programming, domination numbers of $\Gamma_n$ and $\Lambda_n$ are considered in [7] and total domination number of $\Gamma_n$ is considered in [1]. Furthermore, upper bounds and lower bounds on $\gamma(\Gamma_n), \gamma_t(\Gamma_n), \gamma(\Lambda_n)$ are obtained in [1, 2, 13] and they are improved for $\Gamma_n$ in [14].

In this work, we present optimization problems obtained from the degree information of Lucas cubes, whose solutions give better lower bounds on the domination numbers and total domination numbers of Lucas cubes. Our aim is to improve the known results on $\gamma(\Lambda_n)$ and present new results on $\gamma_t(\Lambda_n)$. Furthermore, we introduce the up-down degree polynomials for $\Lambda_n$ containing the degree information of all vertices $V(\Lambda_n)$ in more detail. Using these polynomials we define optimization problems whose solutions give lower bound on $\gamma(\Lambda_n)$ and $\gamma_t(\Lambda_n)$.

2 Preliminaries

For $n \geq 2$ we will use the fundamental decomposition of $\Gamma_n$ (see, [8]):

$$\Gamma_n = 0\Gamma_{n-1} + 10\Gamma_{n-2}, \quad (2.1)$$

where $\Gamma_0 = Q_0$ and $\Gamma_1 = Q_1$. Here note that $0\Gamma_{n-1}$ is the subgraph of $\Gamma_n$ induced by the vertices that start with 0 and $10\Gamma_{n-2}$ is the subgraph of $\Gamma_n$ induced by the vertices that start with 10. Furthermore, $0\Gamma_{n-1}$ has a subgraph isomorphic to $00\Gamma_{n-2}$, and there exists a perfect matching between $00\Gamma_{n-2}$ and $10\Gamma_{n-2}$. Similar to this decomposition for $n \geq 3$ Lucas cubes can be written as

$$\Lambda_n = 0\Gamma_{n-1} + 10\Gamma_{n-3}0, \quad (2.2)$$

where $10\Gamma_{n-3}0$ is the subgraph of $\Lambda_n$ induced by the vertices that start with 10 and end with 0. Here, there exists a perfect matching between $10\Gamma_{n-3}0$ and $00\Gamma_{n-3}0 \subseteq 0\Gamma_{n-1}$. By convention, $\Lambda_1 = \Gamma_0$ and $\Lambda_2 = \Gamma_2$.

The number of vertices of the $\Gamma_n$ is $f_{n+2}$, where $f_n$ are the Fibonacci numbers defined as $f_0 = 0$, $f_1 = 1$ and $f_n = f_{n-1} + f_{n-2}$ for $n \geq 2$. Similarly, the number of vertices of the $\Lambda_n$ is $L_n$, where $L_n$ are the Lucas numbers defined as $L_0 = 2$, $L_1 = 1$ and $L_n = L_{n-1} + L_{n-2}$ for $n \geq 2$.

Let $x, y$ be two binary strings of length $n$. Then the Hamming distance between $x$ and $y$, $d_H(x, y)$ is the number of coordinates in which they differ. The Hamming weight of $x$, $w(x)$ is the number of nonzero coordinates in $x$. Note that Hamming distance is the usual graph distance in $Q_n$.

In Figure 1 we present small dimensional Lucas cubes and a minimal total dominating set with circled vertices for $2 \leq n \leq 5$. 
3 Integer linear programming for domination numbers

In this section, we describe a linear programming problem used in [7] for finding the domination number of $\Gamma_n$ and $\Lambda_n$. A similar approach is used in [1] for finding the total domination number of $\Gamma_n$. The main difficulty for these methods are the number of variables and the number of constraints which are equal to the number of vertices in $\Gamma_n$ and $\Lambda_n$. Using this approach we obtain the total domination number of $\Lambda_n$ for $n \leq 12$.

Let $N(v)$ denote the set of vertices adjacent to $v$ and $N[v] = N(v) \cup \{v\}$. Suppose each vertex $v \in V(\Lambda_n)$ is associated with a binary variable $x_v$. The problems of determining $\gamma(\Lambda_n)$ and $\gamma_t(\Lambda_n)$ can be expressed as a problem of minimizing the objective function

$$
\sum_{v \in V(\Lambda_n)} x_v
$$

subject to the following constraints for every $v \in V(\Lambda_n)$:

$$
\sum_{a \in N[v]} x_a \geq 1 \quad \text{(for domination number)},
\sum_{a \in N(v)} x_a \geq 1 \quad \text{(for total domination number)}.
$$

The value of the objective function gives $\gamma(\Lambda_n)$ and $\gamma_t(\Lambda_n)$ respectively. Note that this problem has $L_n$ variables and $L_n$ constraints. In [7] $\gamma(\Lambda_n)$ is obtained up to $n = 11$ and for larger values of $n$ as the number of variables increases no results are presented.

We implemented the integer linear programming problem (3.1) using CPLEX in NEOS Server [3, 4, 5] for $n \leq 12$ and obtain the values of $\gamma_t(\Lambda_n)$ for $n \leq 12$ and obtain the estimates $49 \leq \gamma(\Lambda_{12}) \leq 54$ (takes approximately 2 hours). We collect the known values of $\gamma(\Lambda_n)$ for $n \leq 11$ (see, [7]) and the new values of $\gamma_t(\Lambda_n)$ that we obtained from (3.1) for $n \leq 12$ in Table 1.

The fundamental decompositions (2.1) and (2.2) of $\Gamma_n$ and $\Lambda_n$ are used to obtain the following relations between $\gamma(\Lambda_n)$ and $\gamma(\Gamma_n)$. The main idea in the proof is to partition the set of vertices into disjoint subsets.
Proposition 3.1 ([2, Proposition 3.1]). Let \( n \geq 4 \), then

(i) \( \gamma(\Lambda_n) \leq \gamma(\Gamma_{n-1}) + \gamma(\Gamma_{n-3}) \)

(ii) \( \gamma(\Lambda_n) \leq \gamma(\Gamma_n) \leq \gamma(\Lambda_n) + \gamma(\Gamma_{n-4}) \)

Using a similar idea we obtain the following result.

Proposition 3.2. Let \( n \geq 4 \), then

(i) \( \gamma_t(\Lambda_n) \leq \gamma_t(\Gamma_{n-1}) + \gamma_t(\Gamma_{n-3}) \).

(ii) \( \gamma_t(\Lambda_n) \leq \gamma_t(\Gamma_n) \leq \gamma_t(\Lambda_n) + \gamma_t(\Gamma_{n-4}) \).

Proof. The proof mimics the proof of [2, Proposition 3.1].

(i) The vertices of \( \Lambda_n \) can be partitioned into vertices that start with 0 and vertices that start with 1. The subgraphs induced by these vertices are isomorphic to \( \Gamma_{n-1} \) and \( \Gamma_{n-3} \) respectively, hence we infer that \( \gamma_t(\Lambda_n) \leq \gamma_t(\Gamma_{n-1}) + \gamma_t(\Gamma_{n-3}) \).

(ii) Let \( D_T \) be a minimal total dominating set of \( \Gamma_n \) and set

\[
D'_T = \{ \alpha \mid \alpha \text{ is a Lucas string from } D_T \} \cup \{ 0\alpha0 \mid 1\alpha1 \in D_T \}.
\]

Note that \( |D'_T| \leq |D_T| \) and a vertex of the form \( 1\alpha1 \) dominates two Lucas vertices of the form \( 0\alpha1 \) and \( 1\alpha0 \). Since these two vertices are dominated by \( 0\alpha0 \), we say that \( D'_T \) is a dominating set of \( \Lambda_n \). Then we need to show that it is also a total dominating set. We know that every vertex in \( v \in V(\Lambda_n) \subseteq V(\Gamma_n) \) is adjacent to some vertex \( \beta \in D_T \). Then if \( \beta \in D'_T \), we are done. Otherwise, \( \beta \) must be of the form \( 1\alpha1 \in D_T \). In this case \( v \in V(\Lambda_n) \) must be of the form \( 1\alpha0 \) or \( 0\alpha1 \), which means that \( v \) is also adjacent to a vertex of the form \( 0\alpha0 \in D'_T \). It follows that \( \gamma_t(\Lambda_n) \leq \gamma_t(\Gamma_n) \). On the other hand, a total dominating set of \( \Lambda_n \) dominates all vertices of \( \Gamma_n \) but the vertices of the form \( 10\alpha01 \) where the subgraph induced by these vertices is isomorphic to \( \Gamma_{n-4} \). Hence we have \( \gamma_t(\Gamma_n) \leq \gamma_t(\Lambda_n) + \gamma_t(\Gamma_{n-4}) \).

\[ \blacksquare \]

Considering the vertices of high degrees a lower bound on \( \gamma(\Lambda_n) \) is obtained in [2, Theorem 3.5.] as \( \gamma(\Lambda_n) \geq \left\lceil \frac{L_{n-2n}}{n-3} \right\rceil \) where \( n \geq 7 \). Combining this result with the fact that \( \gamma_t(\Lambda_n) \geq \gamma(\Lambda_n) \) we have the following lower bound on \( \gamma_t(\Lambda_n) \).

Proposition 3.3. For any \( n \geq 7 \), we have \( \gamma_t(\Lambda_n) \geq \left\lceil \frac{L_{n-2n}}{n-3} \right\rceil \).
4 Up-down degree enumerator polynomial

In this section we present the up-down degree enumerator polynomial for $\Lambda_n$ similar to the one for $\Gamma_n$ given in [14]. Using this polynomial we write optimization problems whose solutions give lower bounds on $\gamma(\Lambda_n)$ and $\gamma_t(\Lambda_n)$.

By the definition of the edge set $E(\Lambda_n)$, $(v, v') \in E(\Lambda_n)$ if and only if the number of different coordinates of $v$ and $v'$ is 1, that is, the Hamming distance $d_H(v, v') = 1$. Here we have at most two kinds of neighbor $v'$ for a vertex in $v \in V(\Lambda_n)$, whose weights can take the values $w(v) \pm 1$. If $w(v') = w(v) + 1$ we call $v'$ an up neighbor of $v$ and if $w(v') = w(v) - 1$ we call $v'$ a down neighbor of $v$. We denote the number of up neighbors of $v$ by $u$ and the number of down neighbors of $v$ by $d$ which is equal to the $w(v)$ by the definition of $\Lambda_n$. Note that if the degree of $v$ is $k$ then we have $u = k - d$.

For each fixed $v \in V(\Lambda_n)$ having degree $k = \deg(v)$, we write a monomial $x^u y^d$ where $d = w(v)$ is the Hamming weight of $v$ and $u$ is $k - d$. We call the polynomial

$$P_{\Lambda_n}(x, y) = \sum_{v \in V(\Lambda_n)} x^{\deg(v) - w(v)} y^{w(v)} = \sum_{v \in V(\Lambda_n)} x^u y^d$$

as the up-down degree enumerator polynomial of $\Lambda_n$.

We need the following useful result given in [10] to obtain the recursive structure of $P_{\Lambda_n}(x, y)$. Let $\ell_{n,k,w}$ be the number of vertices in $\Lambda_n$ of degree $k$ and weight $w$.

**Theorem 4.1** ([10, Theorem 5.2]). For all $n, k, w$ such that $n \geq 2$, $1 \leq k \leq n$ and $0 \leq w \leq n$,

$$\ell_{n,k,w} = \binom{w - 1}{2w + k - n} \binom{n - 2w}{k - w} + 2 \binom{w}{2w + k - n} \binom{n - 2w - 1}{k - w}.$$

Let $\ell'_{n,u,d}$ be the number of vertices in $\Lambda_n$ whose number of up neighbors are $u$ and number of down neighbors are $d$. Setting $k = u + d$ and $w = d$ in Theorem 4.1 we have

$$\ell'_{n,u,d} = \binom{d - 1}{3d + u - n} \binom{n - 2d}{u} + 2 \binom{d}{3d + u - n} \binom{n - 2d - 1}{u}. \quad (4.1)$$

Then using (4.1) we can write the up-down degree enumerator polynomial of $\Lambda_n$ as

$$P_{\Lambda_n}(x, y) = \sum_{u,d} \ell'_{n,u,d} x^u y^d, \quad (4.2)$$

where $0 \leq u, d \leq n$. Furthermore, using (4.1) and (4.2) we obtain the following recursive relation which is very useful to calculate $P_{\Lambda_n}(x, y)$.

**Theorem 4.2.** Let $P_{\Lambda_n}(x, y)$ be the up-down degree enumerator polynomial of $\Lambda_n$. Then for $n \geq 5$ we have

$$P_{\Lambda_n}(x, y) = xP_{\Lambda_{n-1}}(x, y) + yP_{\Lambda_{n-2}}(x, y) + (y - xy)P_{\Lambda_{n-3}}(x, y), \quad (4.3)$$

where $P_{\Lambda_2}(x, y) = x^2 + 2y$, $P_{\Lambda_3}(x, y) = x^3 + 3y$ and $P_{\Lambda_4}(x, y) = x^4 + 4xy + 2y^2$.

**Proof.** The initial conditions are clear from the definition of $\Lambda_n$. For fixed integers $1 \leq u < n$ and $2 \leq d < \lceil \frac{n}{2} \rceil$, the coefficient of the monomial $x^u y^d$ in the right hand side of
the equation (4.3) is the sum of \( \ell_{n-1,u-1,d} \) coming from \( xP_{\Lambda_{n-1}}(x, y) \), \( \ell'_{n-2,u,d-1} \) coming from \( yP_{\Lambda_{n-2}}(x, y) \), \( \ell_{n-3,u,d-1} \) coming from \( yP_{\Lambda_{n-3}}(x, y) \) and \( -\ell'_{n-3,u-1,d-1} \) coming from \( -xyP_{\Lambda_{n-3}}(x, y) \). Then we need to show that

\[
\ell'_{n,u,d} = \ell'_{n-1,u-1,d} + \ell'_{n-2,u,d-1} + \ell'_{n-3,u,d-1} - \ell'_{n-3,u-1,d-1}.
\]

By setting \( X = 3d + u - n \) and \( Y = n - 2d \) in (4.1) and using the binomial identities

\[
\binom{m}{k} = \frac{m!}{(m-k)!k!}
\]

we have

\[
\ell'_{n-1,u-1,d} + \ell'_{n-2,u,d-1} + \ell'_{n-3,u,d-1} - \ell'_{n-3,u-1,d-1} = \left( \binom{d-1}{X} \binom{Y-1}{u} + 2 \binom{d}{X} \binom{Y-2}{u} + \binom{d-2}{Y-1} \binom{u-1}{X} + 2 \binom{d-1}{Y-1} \binom{u-1}{X} \right)
\]

\[
\left( \binom{d-1}{X} \binom{Y-1}{u} - \binom{d-2}{Y-1} \binom{u-1}{X} + 2 \binom{d-1}{Y-1} \binom{u-1}{X} + 2 \binom{d}{Y-2} \binom{u}{X} \right)
\]

\[
\left( \binom{d-1}{X} \binom{Y}{u} \frac{X}{d-1} + \binom{d-2}{Y-1} \binom{u}{X} \frac{X}{d-1} \right)\left( \binom{d}{Y-2} \binom{u}{X} \frac{X}{d-1} + \binom{d-1}{Y-1} \binom{u-1}{X} \frac{X}{d-1} \right)
\]

\[
\left( \binom{d-1}{X} \binom{Y}{u} + 2 \binom{d}{X} \binom{Y-1}{u} \right)
\]

In particular, the case \( d = 0 \) corresponds to the all 0 vertex in \( \Lambda \) and we have \( \ell'_{n,n,0} = 1 \), which means that the coefficient of the terms \( x^ny^0 \) in both sides of (4.3) are 1. Similarly, the case \( u = 0 \) corresponds to the vertices in \( \Lambda \) whose weights are \( \left\lfloor \frac{n}{2} \right\rfloor \) and we have (see, Remark 5.1)

\[
\ell'_{n,0,\left\lfloor \frac{n}{2} \right\rfloor} = \begin{cases} n & \text{if } n \text{ is odd}, \\ 2 & \text{if } n \text{ is even}. \end{cases}
\]

Then one can easily see that the coefficient of the terms \( x^0y^{\left\lfloor \frac{n}{2} \right\rfloor} \) in both sides of (4.3) are equal to each other. The only remaining particular case is \( d = 1 \). For \( P_{\Lambda_n}(x, y) \) this case corresponds to the vertices in \( \Lambda \) whose weights are 1. We know that there are \( n \) such vertices in \( \Lambda \) and their number of up neighbors are \( n - 3 \). That is, the coefficient of the term \( x^{n-3}y \) in \( P_{\Lambda_n}(x, y) \) is \( n \). On the other hand the coefficient of the term \( x^{n-3}y \) is \( n - 1 \) in \( xP_{\Lambda_{n-1}}(x, y); 0 \) in \( yP_{\Lambda_{n-2}}(x, y) \) and 1 in \( (y - xy)P_{\Lambda_{n-3}}(x, y) \) respectively. Hence the coefficient of the terms \( x^ny^d \) in both sides of (4.3) are equal to each other for all cases. \( \square \)

**Remark 4.3.** The recursive relation for the up-down degree enumerator polynomial of \( \Lambda \) in Theorem 4.2 is the same with the recursive relation for the up-down degree enumerator polynomial of \( \Gamma \), which is proved using the fundamental decomposition \( \Gamma = 0 \Gamma_{n-1} + 10 \Gamma_{n-2} \). The only differences are the initial polynomials. For the proof we directly used the degree information of \( \Lambda \) obtained in [10], since \( \Lambda \) do not have a decomposition like \( 0 \Lambda_{n-1} + 10 \Lambda_{n-2} \).
5 Lower bounds on domination numbers using optimization problems

In this section, we present optimization problems giving lower bounds on \( \gamma(\Lambda_n) \) and \( \gamma_t(\Lambda_n) \), whose number of variables and number of constraints are fewer than the general optimization problem described in Section 3.

We use the up-down degree enumerator polynomial \( P_{\Lambda_n}(x, y) \) to construct an optimization problem, which is similar to the optimization problem given in [14]. Let \( D \) and \( D_T \) be a dominating set and a total dominating set of \( \Lambda_n \) respectively. Let \( v_D \in D \) (\( v_D \in D_T \) respectively) and \( x^u y^d \) be its corresponding monomial in \( P_{\Lambda_n}(x, y) \). Then \( v_D \) dominates \( u \) distinct vertices \( v \in V(\Lambda_n) \) having weight \( w(v) = w(v_D) + 1 \) and \( d \) distinct vertices \( v \in V(\Lambda_n) \) having weight \( w(v) = w(v_D) - 1 \). Note that for all \( v_D \in D \) (\( v_D \in D_T \) respectively) some of the vertices of \( \Lambda_n \) may be dominated more than one times. Note that for every vertex \( v \in V(\Lambda_n) \) there must exist at least one vertex \( v_D \in N[v] \cap D_T \) with \( w(v_D) = w(v) \mp 1 \) or \( v_D = v \) for the dominating set \( D \) and \( v_D \in N(v) \cap D_T \) with \( w(v_D) = w(v) \mp 1 \) for the total dominating set \( D_T \).

Now we write the up-down degree enumerator polynomial of \( \Lambda_n \) (see, 4.2) as

\[
P_{\Lambda_n}(x, y) = \sum_{u, d} c_d^u x^u y^d, \tag{5.1}
\]

where \( c_d^u = \ell'_{n, u, d} \). For each pair \((u, d)\) in the monomials of the up-down degree enumerator polynomial \( P_{\Lambda_n}(x, y) \) we associate an integer variable \( z_d^u \) which counts the number of vertices in \( D \) or \( D_T \) having \( d \) down neighbors and \( u \) up neighbors. For any fixed value of \( d \), the number of vertices having weight \( d \) gives the bounds \( 0 \leq z_d^u \leq c_d^u \). Our aim is to minimize \( |D| \) for domination number and to minimize \( |D_T| \) for total domination number. Hence our objective function is to minimize

\[
\sum_{u, d} z_d^u.
\]

To dominate all the vertices having a fixed weight \( d \) such that \( 1 \leq d \leq \lfloor \frac{n}{2} \rfloor - 1 \) we must have the following constraints \( r_d \) for domination number and \( r_d' \) for the total domination number.

\[
r_d : \sum_u (u \cdot z_{d-1}^u + z_d^u + (d + 1) \cdot z_{d+1}^u) \geq \sum_u c_d^u
\]

\[
r_d' : \sum_u (u \cdot z_{d-1}^u + (d + 1) \cdot z_{d+1}^u) \geq \sum_u c_d^u
\]

since any vertex corresponding to the monomial \( x^u y^{d-1} \) can dominate \( u \) distinct vertices \( (u \) up neighbors) having weight \( d \) and any vertex corresponding to the monomial \( x^u y^{d+1} \) can dominate \( d + 1 \) distinct vertices \( (d + 1 \) down neighbors) having weight \( d \). By the same argument, for \( d = 0 \) we must have

\[
r_0 : \sum_u z_0^u + z_1^u \geq \sum_u c_0^u = 1 \quad \text{and} \quad r_0' : \sum_u z_0^u \geq \sum_u c_0^u
\]

and for \( d = \lfloor \frac{n}{2} \rfloor \) we must have

\[
r_{\lfloor \frac{n}{2} \rfloor} : \sum_u u \cdot z_{\lfloor \frac{n}{2} \rfloor - 1}^u + z_{\lfloor \frac{n}{2} \rfloor}^u \geq \sum_u c_{\lfloor \frac{n}{2} \rfloor}^u = \begin{cases} n & \text{if } n \text{ is odd}, \\ 2 & \text{if } n \text{ is even}. \end{cases}
\]
Then using the recursion in Theorem 4.2 as
\[
\sum_{u} u \cdot z_{\lfloor \frac{n}{2} \rfloor}^{u} - 1 \geq \sum_{u} c_{\lfloor \frac{n}{2} \rfloor}^{u} = \begin{cases} 
\frac{n}{2} & \text{if } n \text{ is odd}, \\
2 & \text{if } n \text{ is even}.
\end{cases}
\]

Now subject to the above constraints \( r_0, \ldots, r_{\lfloor \frac{n}{2} \rfloor} \) (constraints \( r'_0, \ldots, r'_{\lfloor \frac{n}{2} \rfloor} \)) the value of the objective function will be a lower bound on \( \gamma(\Lambda_n) \) (\( \gamma_t(\Lambda_n) \), respectively).

**Remark 5.1.** The number of vertices of \( \Lambda_n \) having weight \( d \) is equal to the right hand side of the above constraints \( r_d \) and \( r'_d \). By setting \( k = u + d \) and \( w = d \) in [10, Corollary 5.3] we have
\[
\sum_{u} c_{d}^{u} = \sum_{u=0}^{n-d} l'_{n,u,d} = \binom{n-d}{d} + \binom{n-d-1}{n-2d}.
\]

**Remark 5.2.** The number of variables \( z_d^u \) in our optimization problem is equal to the number of monomials in \( P_{\Lambda_n}(x,y) \). Assume that \( n \) is even. By the string representation of the vertices in \( \Lambda_n \) we have \( n - 3d \leq u \leq n - 2d - 1 \). The bounds come from the maximum number of the sub-strings 010 and 10 in the representation of the vertices. That is, \( u \) can take \( n - 2d - 1 - (n - 3d) + 1 = d \) distinct values when \( d \) ranges from 1 up to \( \lfloor \frac{n}{3} \rfloor \) and \( u \) can take \( n - 2d \) distinct values when \( \lfloor \frac{n}{4} \rfloor + 1 \leq d < \lfloor \frac{n}{2} \rfloor \). Furthermore, \( u \) can take only one values for \( d = 0 \) and \( d = \lfloor \frac{n}{2} \rfloor \). Therefore, the number of variables \( z_d^u \) becomes
\[
2 + \sum_{d=1}^{\lfloor \frac{n}{4} \rfloor} d + \sum_{d=\lfloor \frac{n}{2} \rfloor+1}^{\lfloor \frac{n}{3} \rfloor} (n - 2d)
\]
which is equal to
\[
2 + \frac{3}{2} \lfloor \frac{n}{3} \rfloor \left( \left\lfloor \frac{n}{3} \right\rfloor + 1 - \frac{2n}{3} \right) + \frac{n}{2} \left( n - \left\lfloor \frac{n}{2} \right\rfloor + 1 \right) - n. \tag{5.2}
\]

For \( n \geq 2 \) this sequence starts as 2, 2, 3, 3, 5, 5, 7, 8, 10, 11, 14, 15, 18, 20, 23, 25, 29, \ldots. Note that in (3.1) the number of variables is \( L_n \), which exhibit exponential growth. In our case, if we omit the floor functions in (5.2) then the number of variables \( z_d^u \) is approximately equals to \( 2 + \frac{n^2}{12} \).

For \( n = 12 \) we illustrate our optimization problem as follows. First we obtain \( P_{\Lambda_{12}}(x,y) \) by using the recursion in Theorem 4.2 as
\[
P_{\Lambda_{12}}(x,y) = \begin{cases} 
2y^6 + \\
12y^5x + 24y^5 + \\
12y^4x^2 + 54y^4x^2 + 36y^4x + 3y^4 + \\
12y^3x^3 + 60y^3x^3 + 40y^3x^3 + \\
12y^2x^4 + 42y^2x^4 + \\
x^5 + \end{cases}
\]

Then using \( P_{\Lambda_{12}}(x,y) \) we have the corresponding optimization problem:
/* Objective function */

\[ \text{minimize} : \quad \sum_{i=0}^{12} z_i^2 + \sum_{i=1}^{9} z_i + \sum_{i=2}^{7} z_i^2 + \sum_{i=6}^{6} z_i + \sum_{i=5}^{5} z_i + \sum_{i=4}^{4} z_i + \sum_{i=3}^{3} z_i + z_2^3 + z_1^2 + z_0^2 \]

/* Constraints for \( \gamma(\Lambda_{12}) \) */

\[
\begin{align*}
r_6 : & \quad z_3^3 + z_2^2 + z_1^2 + z_0^2 \geq 1; \\
r_5 : & \quad 3z_4^3 + 2z_4^2 + z_3^2 + z_2^2 + z_1^2 + z_0^2 + 6z_6^0 \geq 36; \\
r_4 : & \quad 5z_4^5 + 4z_4^4 + 3z_4^3 + z_4^2 + z_4^1 + 5z_4^0 \geq 105; \\
r_3 : & \quad 7z_4^7 + 6z_4^6 + z_3^5 + z_3^4 + z_3^3 + 4z_4^2 + 4z_4^1 + 4z_4^0 \geq 112; \\
r_2 : & \quad 9z_4^9 + z_2^7 + z_2^6 + 3z_3^5 + 3z_3^4 + 3z_3^3 \geq 54; \\
r_1 : & \quad 12z_4^{12} + z_1^9 + 2z_2^7 + 2z_2^6 \geq 12; \\
r_0 : & \quad z_0^{12} + z_1^9 \geq 1;
\end{align*}
\]

/* Constraints for \( \gamma_t(\Lambda_{12}) \) */

\[
\begin{align*}
r_6' : & \quad z_5^1 \geq 2; \\
r_5' : & \quad 3z_4^3 + 2z_4^2 + z_3^2 + z_2^2 + z_1^2 + z_0^2 \geq 36; \\
r_4' : & \quad 5z_4^5 + 4z_4^4 + 3z_4^3 + z_4^2 + z_4^1 + 5z_4^0 \geq 105; \\
r_3' : & \quad 7z_4^7 + 6z_4^6 + z_3^5 + z_3^4 + z_3^3 + 4z_4^2 + 4z_4^1 + 4z_4^0 \geq 112; \\
r_2' : & \quad 9z_4^9 + 3z_3^5 + 3z_3^4 + 3z_3^3 \geq 54; \\
r_1' : & \quad 12z_4^{12} + z_1^9 \geq 12; \\
r_0' : & \quad z_1^9 \geq 1;
\end{align*}
\]

/* Bounds */

\[
\begin{align*}
\sum_{i=0}^{12} z_i^2 & \leq 1; \quad \sum_{i=1}^{9} z_i & \leq 12; \quad \sum_{i=2}^{7} z_i^2 & \leq 12; \quad \sum_{i=6}^{6} z_i & \leq 12; \quad \sum_{i=5}^{5} z_i & \leq 60; \quad \sum_{i=4}^{4} z_i & \leq 40; \\
\sum_{i=3}^{3} z_i & \leq 12; \quad \sum_{i=2}^{2} z_i^2 & \leq 54; \quad \sum_{i=1}^{1} z_i^3 & \leq 36; \quad \sum_{i=0}^{0} z_i^4 & \leq 24; \quad \sum_{i=0}^{0} z_i^5 & \leq 2; 
\end{align*}
\]

Depending on the constraints \( r_d \) and \( r_d' \) (\( d = 0, 1, \ldots, 6 \)) the value of the objective function gives a lower bound on \( \gamma(\Lambda_{12}) \) and \( \gamma_t(\Lambda_{12}) \) respectively. The above problem has only 14 variables and 7 constraints (instead of having \( L_{12} = 322 \) variables and 322 constraints as in (3.1)). To find lower bounds on \( \gamma(\Lambda_n) \) and \( \gamma_t(\Lambda_n) \) one can use the up-down degree enumerator polynomial \( P_{\Lambda_n}(x, y) \) of \( \Lambda_n \) in Theorem 4.2 and one can write an optimization problem having fewer number of variables \( z_d^u \) (see Remark 5.2) and \( \left\lceil \frac{n}{2} \right\rceil + 1 \) constraints \( r_d \) or \( r_d' \). The solutions of the optimization problems give lower bounds on \( \gamma(\Lambda_n) \) and \( \gamma_t(\Lambda_n) \). It is easy to see that the number of variables and the number of constraints in our optimization problems are very smaller than the ones in the optimization problem (3.1).

For illustration we implemented the above integer linear programming problem using CPLEX in NEOS Server [3, 4, 5] for \( 12 \leq n \leq 26 \) and immediately (less than 0.02 seconds) obtain the lower bounds on \( \gamma(\Lambda_n) \) and \( \gamma_t(\Lambda_n) \) presented in Table 2 and Table 3 (better than the ones in Proposition 3.3). Note that for \( n = 26 \), the number of variables in our optimization problem is 58 by Remark 5.2 and the number of constraints is 14, on the other hand, these numbers are equal to \( L_{26} = 271443 \) for the general optimization problem (3.1). In addition, the upper bounds in these tables are obtained by Proposition 3.1 and Proposition 3.2 by using the upper bounds on the values of \( \gamma(\Gamma_n) \) and \( \gamma_t(\Gamma_n) \) given in [14] for \( n \geq 14 \).
Table 2: Current best bounds on $\gamma(\Lambda_n)$, $12 \leq n \leq 26$.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$\gamma(\Lambda_n)$</th>
<th>$n$</th>
<th>$\gamma(\Lambda_n)$</th>
<th>$n$</th>
<th>$\gamma(\Lambda_n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>49*-54</td>
<td>17</td>
<td>310-555</td>
<td>22</td>
<td>2686-6140</td>
</tr>
<tr>
<td>13</td>
<td>61*-86</td>
<td>18</td>
<td>471-895</td>
<td>23</td>
<td>4184-9935</td>
</tr>
<tr>
<td>14</td>
<td>89-132</td>
<td>19</td>
<td>725-1450</td>
<td>24</td>
<td>6519-16075</td>
</tr>
<tr>
<td>15</td>
<td>134-215</td>
<td>20</td>
<td>1114-2345</td>
<td>25</td>
<td>10163-26010</td>
</tr>
<tr>
<td>16</td>
<td>203-340</td>
<td>21</td>
<td>1724-3795</td>
<td>26</td>
<td>15835-42085</td>
</tr>
</tbody>
</table>

Table 3: Current best bounds on $\gamma_t(\Lambda_n)$, $12 \leq n \leq 26$.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$\gamma_t(\Lambda_n)$</th>
<th>$n$</th>
<th>$\gamma_t(\Lambda_n)$</th>
<th>$n$</th>
<th>$\gamma_t(\Lambda_n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>58*</td>
<td>17</td>
<td>340-567</td>
<td>22</td>
<td>2893-6140</td>
</tr>
<tr>
<td>13</td>
<td>77*-95</td>
<td>18</td>
<td>514-909</td>
<td>23</td>
<td>4490-9935</td>
</tr>
<tr>
<td>14</td>
<td>101-145</td>
<td>19</td>
<td>787-1450</td>
<td>24</td>
<td>6974-16075</td>
</tr>
<tr>
<td>15</td>
<td>151-231</td>
<td>20</td>
<td>1205-2345</td>
<td>25</td>
<td>10839-26010</td>
</tr>
<tr>
<td>16</td>
<td>225-362</td>
<td>21</td>
<td>1862-3795</td>
<td>26</td>
<td>16838-42085</td>
</tr>
</tbody>
</table>

Remark 5.3. It is shown in [7, 1] that $\gamma(\Gamma_9) = 17$, $\gamma(\Gamma_{10}) = 25$, $54 \leq \gamma(\Gamma_{12}) \leq 61$ and $78 \leq \gamma(\Gamma_{13}) \leq 93$ (shown in [14]). Substituting these results in Proposition 3.1 we obtain the bounds for $n = 13$ in Table 2.

Similarly, it is shown in [7, 1] that $\gamma_t(\Gamma_9) = 20$, $\gamma_t(\Gamma_{10}) = 30$, $\gamma_t(\Gamma_{12}) = 65$ and $97 \leq \gamma_t(\Gamma_{13}) \leq 101$. Substituting these results in Proposition 3.2 we obtain the bounds for $n = 13$ in Table 3.

Note that our optimization problems obtained from up-down degree enumerator polynomial give $\gamma(\Lambda_{12}) \geq 39$, $\gamma_t(\Lambda_{12}) \geq 45$ and $\gamma(\Lambda_{13}) \geq 59$, $\gamma_t(\Lambda_{13}) \geq 68$. Furthermore, using (3.1) we obtain that $49 \leq \gamma(\Lambda_{12}) \leq 54$ and $\gamma_t(\Lambda_{12}) = 58$. For these reasons we put a * to the lower bounds for the cases $n = 12$ and $n = 13$ in Table 2 and Table 3.

References


