Matemática
Contemporânea

Vol. #, 1–9
http://doi.org/will_be_added_later

Finding graphs with exponential clique-growth using genetic algorithms

M.A. Pizaña[1] and **I.A. Robles**[1]

[1]Universidad Autónoma Metropolitana, Av. San Rafael Atlixco 186, Mexico City, 09340, Mexico

Dedicated to Professor Jayme Szwarcfiter
on the occasion of his 80th birthday

Abstract. The clique graph $K(G)$ is the intersection graph of the set of all the (maximal) cliques of G. The iterated clique graphs of G are defined inductively by $K^0(G) = G$ and $K^{n+1}(G) = K(K^n(G))$. An open problem is to determine whether there is a graph G, with exponential clique-growth rate, i.e. such that $|K^n(G)| = \Theta(t^n)$, for some $t > 1$. In this work we report the use of genetic algorithms to find a candidate for such a graph. The circulant $G = C_m(1, 3, 6, 7, 8)$ shows an experimental clique-growth rate of $|K^n(G)| = \Theta\left(\sqrt{3}^n\right)$. Further preliminary theoretical results (beyond the scope of this paper) also suggest that this graph has indeed the desired property, but the open problem still remains to be settled.

Keywords: genetic algorithms, clique graph, graph theory.

2020 Mathematics Subject Classification: 05C69, 05C76, 68W50.

iarobles271@gmail.com

1 Introduction

All our graphs are finite, simple and non-empty. The order of a graph G is denoted by $|G|$. The cycle graph of n vertices is denoted by C_n and the octahedral graph is denoted by O_3. The strong product of graphs G and H is denoted by $G \boxtimes H$. The circulant graph $C_n(a_1, a_2, \ldots, a_r)$ of n vertices, is the graph defined on Z_n where two vertices $x, y \in Z_n$ are adjacent if and only if $x - y \in \{\pm a_1, \pm a_2, \ldots, \pm a_r\}$.

The *clique-growth function* of a graph G is defined as $g_G(n) = |K^n(G)|$. There are known examples of graphs were $g_G(n)$ has linear growth [7], polynomial growth [8] and super-exponential growth [10], but so far, no graph with *exponential growth* (i.e. $g_G(n) = \Theta(t^n)$, with $t > 1$) is known.

Since the number of graphs of order n is huge even for small n (e.g. 6×10^{22} for $n = 16$ [11]) a brute force search is not feasible and it is necessary to use a different approach that can perform searching in a more directed way. For our purposes, genetic algorithms yielded satisfactory results. The method used is summarized in the next section.

2 The genetic algorithm

Genetic algorithms (GAs) are algorithms based on a metaheuristic inspired by the mechanisms of biological evolution [6]. GAs have been successfully used to solve search and optimization problems. If S is the search space of an optimization problem, the *fitness* of a candidate solution $c \in S$, is a positive real value that measures how optimal c is. The *fitness* is computed by a *fitness function* $f : S \to R^+$, that is defined according to the optimization problem. In our case S is the set of graphs of n vertices and the fitness function f is described in section 3.

A summary of the general steps that are executed sequentially for the GA used in this work is given below.

1. *Initialization*: create an initial population with at least 100 random graphs of n vertices.

2. *Reproduction*: randomly choose a pair of graphs G_1 and G_2 from the population. The probability of a graph G of being chosen is proportional to its *fitness* $f(G)$. The fitness is computed with parameters $N = 7$ and $t = 4$ (see Definition 3.1).

3. *Crossover*: let $\phi(X)$ denote a uniformly chosen random element of the set X. If M_1 and M_2 are the $n \times n$ adjacency matrices of G_1 and G_2, respectively, create a graph G_3 with $n \times n$ adjacency matrix M_3, that will be the *descendant* of G_1 and G_2. The entry $(M_3)_{ij}$ in row i and column j of the adjacency matrix M_3, is defined as follows:

$$(M_3)_{ij} = \begin{cases} \phi(\{(M_1)_{ij},\ (M_2)_{ij}\}), & \text{for } j > i, \\ (M_3)_{ji}, & \text{for } j < i, \\ 0, & \text{for } i = j. \end{cases}$$

4. *Mutation*: for each pair of different vertices $x, y \in G_3$ (the descendant), randomly toggle with probability $p_m = 0.001$, its adjacency relation (i.e. if x and y are adjacent, make them non-adjacent and vice versa).

5. *Predation*: randomly replace a graph of the population with the descendant. The probability of a graph G of being replaced is inversely proportional to its fitness $f(G)$.

6. *Verification*: if G is the graph with the maximum fitness of the population, check if $f(G) > 0.99$ (the maximum fitness is 1), if that's the case the algorithm stops, otherwise repeat from step *2*.

The code of the GA described above was implemented using the computer algebra system GAP (Groups, Algorithms and Programming [5]) with the package YAGS (Yet Another Graph System [2]). The next section describes in more detail the fitness function used for the GA.

3 The fitness function

Since we want the GA to search graphs G such that its growth function satisfies $g_G(n) = \Theta(t^n)$, with $t > 1$, our general approach to define the fitness function $f(G)$ is as a measure of how optimal $g_G(n)$, can be approximated by the function:

$$y_e(n) = A \cdot B^n,$$

where A and B are constants to be determined. Applying logarithm to the previous function we obtain:

$$Ln(y_e(n)) = Ln(A \cdot B^n) = Ln(A) + Ln(B) \cdot n.$$

By taking $y(n) = Ln(y_e(n))$, $a = Ln(A)$ and $b = Ln(B)$ in the above equation, we obtain the following lineal equation:

$$y(n) = a + bn. \tag{3.1}$$

By using the function $y(n)$ instead of $y_e(n)$, we can apply the least squares method [4], for computing the constants a and b. Under this approach the fitness function f can be defined as a measure of how optimally the linear regression model $y(n)$ approximates $Ln(g_G(n))$ (the precise definition is stated later in Definition 3.6). However, there are graphs with super-exponential clique-growth rate, which makes it impossible to compute $g_G(n)$ even for $n = 4$. For instance, in the case of the octahedron O_3, it is well known that $g_{O_3}(n) = \sqrt{2}^{\,g_{O_3}(n-1)}$ [10], and hence

$$g_{O_3}(4) = \sqrt{2}^{\sqrt{2}^{\sqrt{2}^{\sqrt{2}^{6}}}} \approx 3 \times 10^{38}.$$

We *can* compute this order because of the established recurrence for $g_{O_3}(n)$, but we can not hope to do so in general.

To avoid problems caused by $g_G(n)$ growing too fast, we use the following iterative algorithm: given $t > 1$ and $g_G(0) = |G|$, for computing $g_G(n)$ search for the cliques of the graph $K^{n-1}(G)$, keeping track of the total cliques found at any moment, if this number exceeds $t \cdot g_G(n-1)$, stop the search; otherwise continue until computing $g_G(n) = |K^n(G)|$. This ensures that we only compute values that satisfy $g_G(n) \leq |G| \cdot t^n$, which lead us to the following definition.

Definition 3.1 (Logarithmic growth vector of a graph)**.** Given a graph G, an integer $N \geq 1$ and $t > 1$, let $M \leq N$ be the maximum integer for which $g_G(n) \leq |G| \cdot t^n$, for all $n \leq M$. The logarithmic growth vector is defined as follows:

$$Y_{G,N,t} = (Ln(g_G(0)), Ln(g_G(1)), \ldots, Ln(g_G(M)))\,.$$

Remark 3.2. We will use the notation Y_G instead of $Y_{G,N,t}$, if its clear in the context what the values for N and t are.

Remark 3.3. Note that Definition 3.1, implies that $|Y_G| = N+1$ if $g_G(n) \leq |G| \cdot t^n$, for $0 \leq n \leq N$ and $|Y_G| < N+1$, otherwise.

Using Definition 3.1, we can restate our definition for the fitness function f as a measure of how optimal is the approximation of the vector Y_G using the lineal regression model $y(n) = a + bn$ in (3.1). To refine this definition, we will make use of the *correlation coefficient* [3], whose definition has been adapted for the context of this work and is described below.

Definition 3.4 (Correlation coefficient)**.** Given a graph G, an integer $N \geq 1$ and $t > 1$, let Y_G be the logarithmic growth vector of the graph G and define the vector $X_G = (0, 1, \ldots, |Y_G|)$. The correlation coefficient ρ_G of the graph G, is defined as follows:

$$\rho_G = \frac{Cov(X_G, Y_G)}{\sqrt{Var(X_G) \cdot Var(Y_G)}}.$$

Where Var and Cov are the variance and covariance respectively [3].

Remark 3.5. ρ_G is undefined, if $Var(X_G) = 0$ or $Var(Y_G) = 0$.

It is well known that ρ_G^2 has the property that its value tends to 1, the better a linear regression model approximates a set of points and tends to

0, otherwise [3]. In principle ρ_G^2 could be used as a fitness function, but we need to consider that by Remark 3.3, if $|Y_G| < N+1$, then the graph G has $N+1-|Y_G|$ iterated clique graphs that growth faster than $|G| \cdot t^n$. Therefore, the fitness function should give a higher value for those plots for which $|Y_G| = N+1$ and a lower value if $|Y_G| < N+1$. This lead us to state the fitness function as follows:

Definition 3.6 (Fitness function)**.** Given a graph G, an integer $N \geq 1$ and $t > 1$, let $Y_G = Y_{G,N,t}$ be its logarithmic growth vector (Definition3.1). Define the vector $X_G = (0, 1, \ldots, |Y_G|)$ and let ρ_G be the correlation coefficient of G (Definition 3.4). The fitness $f(G)$ is defined as follows:

$$f(G) = \begin{cases} \rho_G^2 \cdot \left(\frac{|Y_G|}{N+1}\right), & \text{if } Var(Y_G) \neq 0 \text{ and } Var(X_G) \neq 0, \\ 0, & \text{otherwise.} \end{cases}$$

Graph G	Growth $g_G(n)$	Fitness $f(G)$
C_{10}	$\Theta(1)$	0
$C_{13}(1,3,4)$	$\Theta(n)$	0.918
$C_{13}(1,3,4) \boxtimes C_{13}(1,3,4)$	$\Theta(n^2)$	0.941
O_3	$\Theta\left(\underbrace{\sqrt{2}^{\sqrt{2}^{\cdot^{\cdot^{\cdot^{\sqrt{2}^6}}}}}}_{n \text{ times}}\right)$	0.270
$C_{25}(1,3,6,7,8)$	$\Theta\left(3^{\frac{n}{2}}\right)$ (conjectured)	0.997

Table 3.1: Sample graphs with their clique-growth rate and fitness.

Table 3.1 shows the computed growth and fitness for some sample graphs. Note that the circulant graph $C_{25}(1,3,6,7,8)$, has the highest fitness.

4 Results and conjectures

After running several times the genetic algorithm described in Section 2, the genetic algorithm found the circulant graph $G = C_{25}(1, 3, 6, 7, 8)$, with fitness $f(G) = 0.997$. It is noteworthy that the genetic algorithm worked with graphs in general and not specifically with circulants. The growth function of this circulant behaves like this:

n	0	1	2	3	4	5	6	7	8	9	10	11	12
$g_G(n)$	25	50	100	175	325	550	1000	1675	3025	5050	9100	15175	27325

Table 4.1: Growth rate $g_G(n)$ for $G = C_{25}(1, 3, 6, 7, 8)$

As long as it can be computed, this sequence of numbers satisfy the following recurrence relation:

$$h(n) = \begin{cases} |G|, & \text{for } n = 0, \\ 2 \cdot h(0), & \text{for } n = 1, \\ 3 \cdot h(n-2) + h(0), & \text{for } n \geq 2. \end{cases} \tag{4.1}$$

Using *triangular covering maps* from [9], we know that

$$|K^n(C_m(1, 3, 6, 7, 8))| = \frac{m}{25} |K^n(C_{25}(1, 3, 6, 7, 8))|$$

for all $m \geq 25$ and $n \geq 0$ and hence, the same recurrence relation holds (at least for $n \leq 12$ as in Table 4.1) for all the circulants $\mathrm{C}_m(1, 3, 6, 7, 8)$ with $m \geq 25$.

It is straight forward to prove by induction that the previous recurrence relation is equivalent to the next one:

Lemma 4.1. *The recurrence relation for $h(n)$ in (4.1) can be rewritten* *as follow:*

$$h(n) = \begin{cases} |G|, & \text{for } n = 0, \\ 2 \cdot h(0), & \text{for } n = 1, \\ \frac{9 \cdot h(n-1) + 2 \cdot h(0)}{5}, & \text{for } n \text{ even, with } n \geq 2, \\ \frac{5 \cdot h(n-1) + h(0)}{3}, & \text{for } n \text{ odd, with } n \geq 3. \end{cases}$$

And using standard methods [1], we can obtain the solution to the recurrence:

Lemma 4.2. *The recurrence relation $h(n)$ has the following solution:*

$$h(n) = \begin{cases} \left(\frac{3^{\frac{n}{2}+1} - 1}{2}\right) h(0), & \text{if } n \text{ is even}, \\ \left(\frac{5 \cdot 3^{\frac{n-1}{2}} - 1}{2}\right) h(0), & \text{if } n \text{ is odd}. \end{cases}$$

In particular, $h(n) = \Theta(3^{\frac{n}{2}})$.

Besides the empirical recurrence in (4.1) which matches Table 4.1, we also have preliminary theoretical results (which are beyond this scope of this paper), that also suggest that $h(n)$ is indeed the same as the clique-growth function for $G = C_{25}(1, 3, 6, 7, 8)$. All of this motivates us to propose the following:

Conjecture 4.3 (Exponential growth conjecture)**.** *There are graphs with exponential clique-growth. Moreover, let $m \geq 25$ and $G = C_m(1, 3, 6, 7, 8)$. Then the growth function of G, $g_G(n)$, equals the function $h(n)$ defined by the recurrence relation (4.1). Therefore $g_G(n) = h(n) = \Theta(\sqrt{3}^n)$ and $C_m(1, 3, 6, 7, 8)$ grows exponentially.*

References

[1] Richard A. Brualdi. *Introductory combinatorics.* North-Holland New York, 1977.

[2] C. Cedillo, R. MacKinney-Romero, M. A. Pizaña, I. A. Robles, and R. Villarroel-Flores. *YAGS - Yet Another Graph System, Version 0.0.5*, 2019. http://xamanek.izt.uam.mx/yags/.

[3] Samprit Chatterjee and Ali S. Hadi. *Covariance and correlation coefficient*, pages 25–30. Wiley, 5th edition, 2012.

[4] F.M Dekking, C. Kraaikamp, Lopuhaa H.P, and L.E Meester. *The method of least squares*, pages 329–336. Springer, 2005.

[5] The GAP Group. *GAP – Groups, Algorithms, and Programming, Version 4.11.0*, 2020.

[6] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, 1989.

[7] F. Larrión and V. Neumann-Lara. A family of clique divergent graphs with linear growth. *Graphs and Combinatorics*, 13(3):263–266, Sep 1997.

[8] F. Larrión and V. Neumann-Lara. Clique divergent graphs with unbounded sequence of diameters. *Discrete Mathematics*, 197-198:491 – 501, 1999. 16th British Combinatorial Conference.

[9] F. Larrión and V. Neumann-Lara. Locally C_6 graphs are clique divergent. *Discrete Math.*, 215(1-3):159–170, 2000.

[10] Víctor Neumann-Lara. On clique-divergent graphs. In *Problèmes combinatoires et théorie des graphes (Colloq. Internat. CNRS, Univ. Orsay, Orsay, 1976)*, volume 260 of *Colloq. Internat. CNRS*, pages 313–315. CNRS, Paris, 1978.

[11] N.J.A. Sloane. Number of graphs on n unlabeled nodes: sequence A000088 in the On-line Encyclopedia of Integer Sequences. https://oeis.org/A000088.