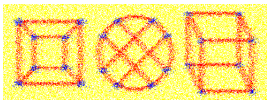
Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 27, no. 9, pp. 767–782 (2023)
DOI: 10.7155/jgaa.00643

Maximum Subgraph Problem and Minimum Linear Arrangement of Generalized Sierpinski Graphs

R. Sundara Rajan [1]
A. Berin Greeni [2] *P. Leo Joshwa* [2]

[1]Department of Mathematics,
Hindustan Institute of Technology and Science,
Chennai 603103, India
[2]School of Advanced Sciences,
Vellore Institute of Technology,
Chennai 600127, India

Submitted: June 2023 Reviewed: October 2023 Revised: October 2023
Reviewed: October 2023 Revised: October 2023 Accepted: October 2023
Final: November 2023 Published: November 2023
Article type: Regular paper Communicated by: M.S. Rahman

Abstract. The creation of scientific supercomputers is one of the most pressing issues confronting technology today. Experts in computer science anticipate that future supercomputers will be built on large-scale parallel processing. A system with multiple processors and memories will be used in such a computer. The interconnection network that allows communication between the system's processors and memories is a critical component of such systems. In the topic of interconnection networks for parallel computer architectures, graph embedding problems have grown in relevance. Network embedding has been recognized as a valuable method for developing efficient algorithms and simulating various architectures in parallel and distributed computing. In this paper, we obtain the maximum subgraph of the generalized Sierpinski graphs $S(n,m), n \geq 2, m \geq 3$, and calculate the minimum linear arrangement of generalized Sierpinski graphs by graph embeddings.

1 Introduction

The mathematical representation of the interconnection network's structure is a graph, where the edges represent the connections between the processors and the vertices represent the network's

E-mail addresses: vprsundar@gmail.com (R. Sundara Rajan, corresponding author) beringreeni@gmail.com (A. Berin Greeni) joseleowa18@gmail.com (P. Leo Joshwa)

structure. An interconnection network is a critical component in parallel computing systems. It consists of physical connections (such as wires or optical fibers) and communication protocols that enable the transfer of data between the different processing units. In a parallel computing environment, multiple computers or processing units work together to solve a problem or perform tasks concurrently. To make this collaboration efficient, a fundamental requirement is the ability to exchange data seamlessly between these parallel computing nodes. Graph embedding stands out as a highly efficient technique for simulating guest graphs into host graphs. Various domains, such as VLSI designs, representations, data structures, biological models of cloning, parallel computer network architecture, and structural engineering, all contribute to the mathematical framework of graph embedding. In addition, dilation, congestion, and wirelength are significant parameters to consider when assessing an embedding.

Edge isoperimetric problems (EIP) and Convex edge partition play a significant role in demonstrating embedding, which is about finding the maximum number of edges induced when particular numbers of vertices are chosen. The concept of edge isoperimetric problems is applied to the guest graph and has a broad range of applications in the fields of computer science and combinatorics [4]. The Convex edge partition is about finding the shortest path within the component when the graph is partitioned into two different components. This concept of convex edge partition is applied in the host graph [2, 7, 20, 23, 25, 22].

Concerning vertex degree, chromatic number, clique number, and network size, the generalized Sierpinski graphs $S(n, m)$ [24] are the most vital networks. Several network topologies based on the generalized Sierpinski graph have been developed and researched in the literature due to their ideal properties [16]. In the last 25 years, many studies have been done on the generalized Sierpinski graph. The study of connection networks, topology, physics, and research into a specific class of universal topological spaces all contribute to the interest in these graphs. Other sources of fascination include games like Chinese Rings or the Tower of Hanoi. Sierpinski graphs, which have applications in topology, the Tower of Hanoi, and computer science, are highly explored fractal graphs. $S(n, m)$ has been utilized in the development of computer and communications networks for decades due to its superior fault-tolerance and routing features. Additionally, it is employed in distributed computing and VLSI design. As a combinatorial optimization problem, finding the ideal linear arrangement of a network in terms of an objective cost function is known as the Minimum Linear Arrangement (MinLA) problem. Overall, the MinLA problem is an NP-complete problem [9].

The family of graphs, known as the generalized Sierpinski graph, has been studied as a model for interconnection networks of multiprocessor computers. Solution and investigation of combinatorial optimization problems on the generalized Sierpinski graph have implications for estimating the communication complexity of a layout, and it has practical importance in the area of VLSI circuit design, and network optimization for parallel computer architecture. Lexicographic ordering has been previously used as a tool for solving problems of the generalized Sierpinski graphs. The vertex set of $S(n, m)$ consists of all n-tuples of integers $1, 2, \ldots, m$ for positive integers n and m with the vertex set $V = \{0, 1, \ldots, m-1\}^n$. Each of the vertices can be numbered from 0 to $m^n - 1$, and each vertex can be given a label which is the m-ary representation of its number. It was shown by Harper [10] that initial segments of the lexicographic order are solutions of the edge-isoperimetric problem on the generalized Sierpinski graphs. A different example of using the lexicographic ordering of the vertices is the construction of a minimum connected hub set [19], where each vertex needs to examine its own label to determine whether it is in a hub set or not. In this paper, we show that initial segments of the lexicographic order are solutions to the maximum subgraph problem on $S(n, m)$. In addition, we solve the minimum linear arrangement on the generalized Sierpinski

graph by labeling the vertices of $S(n,m)$ in lexicographic order, and labeling the vertices of the path graph sequentially, starting from the leftmost vertex.

There have been extensive studies on various mathematical problems related to generalized Sierpinski graphs [13]. These include solving the edge-isoperimetric problem on the Sierpinski graph, and achieving a conclusive resolution by Harper [10]. In addition, Harper [12] investigated if $S(n,m)$ which is the generalized Sierpinski graph could be embedded in K_m^n which is the Hamming graph with the same vertex set and confirmed that the answer is affirmative. In 2015, Rajan et al. [22] conducted a study on the dilation problem of an embedding, and computed the exact dilation of embedding the circulant network into a triangular grid, Tower of Hanoi graph and Sierpinski gasket graph.

In the next section, the basic definitions and preliminaries related to embedding problems are provided. In Section 3 and 4, algorithms to solve the maximum subgraph problem and the minimum linear arrangement problem on generalized Sierpinski graphs $S(n,m), n \geq 2, m \geq 3$ are presented, respectively. For both algorithms, proof of correctness and the time complexity analysis are provided. A Sage code implementation to find the maximum number of edges induced by l vertices ($1 \leq l \leq m^n$) in $S(n,m)$, along with its output, is included in the appendix section. Readers can gain a better understanding of the algorithm or the underlying logic of the Sage program by consulting references [25] and [23]. In Sections 5 and 6, some real-world applications are discussed, and concluding remarks are provided, respectively.

2 Basic Definition

Definition 2.1 [5] Let X(Guest) and Y(Host) be any two graphs. The vertex sets of X and Y are represented by $V(X)$ and $V(Y)$ respectively. The edge sets of X and Y are represented by $E(X)$ and $E(Y)$ respectively. An one-to-one mapping $\mathcal{T} : V(X) \to V(Y)$ is called an *embedding* if each edge $(u,v) \in E(X)$ is mapped into a path in Y between $\mathcal{T}(u)$ and $\mathcal{T}(v)$.

Definition 2.2 [5] Let $\mathcal{T}$ be an embedding from X into Y. For $e \in E(Y)$, the number of path in $\{P_{\mathcal{T}}(u,v) : P_{\mathcal{T}}(u,v)$ is a path between $\mathcal{T}(u)$ and $\mathcal{T}(v)$ in Y for $(u,v) \in E(Y)$ that contains $e\}$ is called the *congestion on* e with respect to $\mathcal{T}$ and is defined by $C_{\mathcal{T}}(e)$. For $S \subseteq E(Y)$, we define $C_{\mathcal{T}}(S) = \sum_{e \in S} C_{\mathcal{T}}(e)$.

Definition 2.3 [5] Let $\mathcal{T} : X \to Y$ be an embedding. Then the *congestion* of $\mathcal{T} : X \to Y$ is given by $C_{\mathcal{T}}(X,Y) = max\ C_{\mathcal{T}}(e)$. The congestion problem is to determine $\min_{\mathcal{T}} C_{\mathcal{T}}(X,Y)$ where the minimum is taken along all embedding $\mathcal{T} : X \to Y$.

Definition 2.4 [20] The wirelength of an embedding $\mathcal{T} : X \to Y$ is given by

$$WL_{\mathcal{T}}(X,Y) = \sum_{e \in E(Y)} C_{\mathcal{T}}(e)$$

and $\min_{\mathcal{T}} WL_{\mathcal{T}}(X,Y)$ is called the *wirelength* of embedding X into Y and is denoted by $WL(X,Y)$ where the minimum is taken along all embeddings $\mathcal{T} : X \to Y$.

When Y is a path, the wirelength of embedding from X into Y is nothing but the minimum linear arrangement of X. The MinLA of a grid $M[3 \times 3]$ with respect to $\mathcal{T}$ is given in Figure 1. The wirelength problem is an NP-complete problem [9].

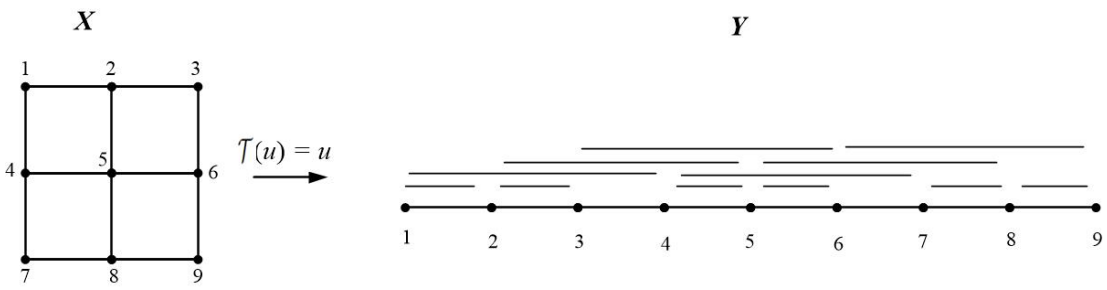

Figure 1: Wiring diagram of a grid X into a path Y with $\text{MinLA}_{\mathcal{T}}(X, Y) = 24$.

Definition 2.5 [4] If all of the shortest paths between any two vertices in a subgraph W of a graph X lies (entirely) only in W, then the subgraph is called *convex*. If $X \setminus S$ partitions into two constituents, say X_1 and X_2 each of which is convex, then the edge cut S of X is said to be a *convex cut*.

Definition 2.6 [4] For a given l, $1 \leq l \leq |V_X|$, find $W \subseteq V$ such that if

$$I_X(l) = \max_{W \subseteq V,\ |W|=l} |I_X(W)|$$

where $I_X(W) = \{(u, v) \in E : (u, v) \in W\}$. A subset of vertices $W \subseteq V$ is called optimal if $|W| = l$ and $|I_X(W)| = I_X(l)$. The problem of finding I_X is called the *maximum subgraph problem*.

Definition 2.7 [4] For a given l, $1 \leq l \leq |V_X|$, find $W \subseteq V$ such that if

$$\Theta_X(l) = \min_{W \subseteq V,\ |W|=l} |\Theta_X(W)|$$

where $\Theta_X(W) = \{(u, v) \in E : u \in W, v \notin W\}$. A subset of vertices $W \subseteq V$ is called optimal if $|W| = l$ and $\Theta_X(W)| = \Theta_X(l)$. The problem of finding Θ_X is called the *minimum cut problem*.

Lemma 2.8 [21] Let $\mathcal{T}$ be an embedding from X into Y. Let S be a convex cut of Y such that $Y \backslash S$ partitions into two constituents Y_1 and Y_2 and let X_1 and X_2 be subgraphs of X induced by $\mathcal{T}^{-1}(V(Y_1))$ and $\mathcal{T}^{-1}(V(Y_2))$ respectively. Suppose X_1 and X_2 are maximal induced subgraph on $|V(X_1)|$ and $|V(X_2)|$ vertices in X. Then

$$C_{\mathcal{T}}(S) = \sum_{v \in V(X_1)} deg_X(v) - 2|E(X_1)| = \sum_{v \in V(X_2)} deg_X(v) - 2|E(X_2)|.$$

Lemma 2.9 [3, 20] Let $\mathcal{T}$ be an embedding from X into Y. Let $E^r(Y)$ be the set of edges of Y repeated precisely r times. Let $\{S_j : 1 \leq j \leq n\}$ be a partition of $E^r(Y)$ such that each S_j is a convex cut of Y and fulfills the condition of the Lemma 2.8. Then,

$$WL(X, Y) = WL_{\mathcal{T}}(X, Y) = \frac{1}{r} \sum_{i=1}^{n} C_{\mathcal{T}}(S_j).$$

3 Maximum Subgraph Problem

The maximum subgraph problem for a given graph X, tries to find the maximum induced subgraph W on l vertices and involves determining the subgraph of X that contains the most edges among all possible subgraphs induced by l vertices. In the case of generalized Sierpinski graphs, we seek to find the maximum number of edges in a subgraph induced by l vertices. The edge-isoperimetric problem (EIP) for generalized Sierpinski graphs has been extensively studied [10]. These solutions provide valuable insights into estimating the edge congestion over the path embedding of the generalized Sierpinski graphs.

Structure of generalized Sierpinski graphs $S(n,m)$: In 1944, Scorer et al. introduced the graph $S(n,m)$, with $n \geq 1$ and $m \geq 2$ [24]. They demonstrated that $S(n,3)$ corresponds to the Tower of Hanoi with n discs. In 1997, Klavžar and Milutinoviá explicitly revealed that the representation of $S(n,m)$ was implicit in Scorer's work [15]. The order of $S(n,m)$ is m^n, indicating that it contains m^n vertices. In $S(n,m)$, all vertices, except for the corner vertices, have a degree of m, while the corner vertices have a degree of $m-1$. By summing the degrees of all vertices, we obtain $m(m-1)+(m^n-m)m$, which simplifies to $m^{n+1}-m$. Since each edge is incident to two vertices, the total number of edges in $S(n,m)$ is $(m^{n+1}-m)/2$.

The special cases include $S(n,1)$ consisting of a single vertex, and $S(n,2)$ forming a path with 2^n vertices. The endpoints of this path are the two corner vertices labeled as 0^n and 1^n. Each interior vertex in $S(n,2)$ is connected to two edges, one leading towards 0^n and the other towards 1^n. The Sierpinski graph $S(n,m)$ holds significance due to its bi-regular structure, short diameter, and solid connections. To enhance its effectiveness, various modifications have been made over time. The metric topological structure of the Sierpinski network $S(n,m)$ is particularly important in different Sierpinski graph variations, as it is considered more reliable and effective than the Sierpinski gasket graph $ST(n,m)$. Its fractal nature, self-similarity, and connectivity properties make it a versatile tool with wide-ranging applications in various fields, including networking, data processing, simulation, image and signal processing, and theoretical research [13].

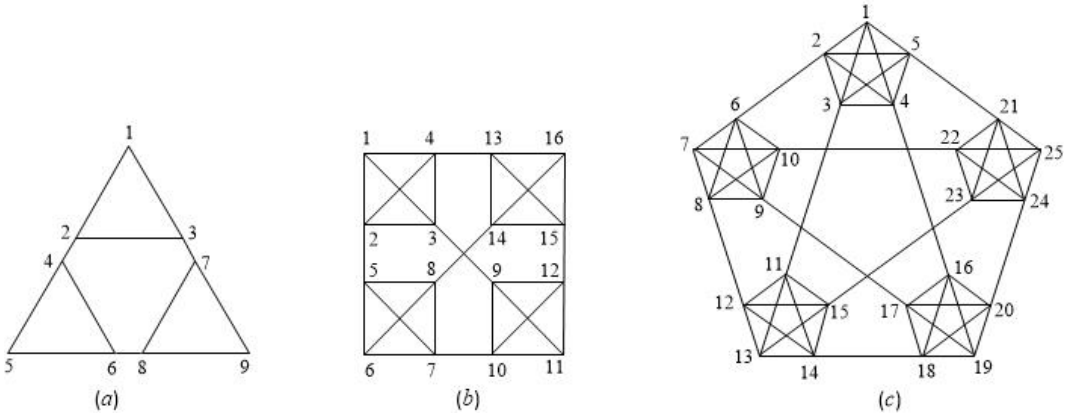

Figure 2: Generalized Sierpinski graphs (a) $S(2,3)$ (b) $S(2,4)$ (c) $S(2,5)$.

Definition 3.1 [14] For positive integers n and m, let $S(n,m)$ be the graph with vertex set $V = \{0,1,\ldots,m-1\}^n$, where two vertices $(u_1,\ldots,u_n)$ and $(v_1,\ldots,v_n)$ are adjacent if there exists an index $h \in [n]$ such that

1. $u_i = v_i$ for $i < h$,
2. $u_h \neq v_h$ and
3. $(u_i, v_i) = (v_h, u_h)$ for $i > h$.

For illustration, the generalized Sierpinski graphs $S(2,3)$, $S(2,4)$ and $S(2,5)$ are given in Figure 2. We can also number the vertices from 0 to $m^n - 1$, interpreting a vertex as the m-ary representation of its number, that is, vertex $(u_1,\ldots,u_n)$ has number

$$\sum_{i=0}^{n-1} u_{n-i} m^i.$$

Then the lexicographic order on V agrees with the usual order on the set $\{0,1,\ldots,m^n-1\}$. It has been proved in [10] that the edge-isoperimetric problem for $S(n,m)$ is solved by taking an initial interval.

Definition 3.2 (Lexicographic ordering [1]) A collection of n-tuples with integer elements is said to be in lexicographic order if $(u_1u_2\cdots u_n)$ is bigger than $(v_1v_2\cdots v_n)$ if and only if an index i, $1 \leq i \leq n$, exists with the property that $u_j{=}v_j$ for $1 \leq j \leq i$ and $u_i > v_i$.

This lexicographic ordering becomes an important tool for finding the maximum subgraph for the Sierpinski graph $S(n,m)$. Initial l-segments of $V(S(n,m))$ in lexicographic order, are solutions to maximize the number of edges for $S(n,m)$.

Theorem 3.3 [10] *Let $X = S(n,m)$ and let l be an integer with $1 \leq l \leq m^n$. Among the subsets $W \subseteq \{0,1,\ldots,m^n-1\}$ of order $W = l$, the size of $X[W]$ is maximized for $W = \{0,1,\ldots,l-1\}$.*

Algorithm to solve the maximum subgraph problem :

Input: The $S(n,m)$, $n \geq 2$, $m \geq 3$, on m^n vertices.

Algorithm: Label the vertices of $S(n,m)$ by lexicographic ordering from 1 to m^n.

Output: Maximum number of edges induced by l vertices in $S(n,m)$, $1 \leq l \leq m^n$.

Proof of correctness: We assume that the labels represent the vertices to which they are assigned. Let $l = \sum_{i=0}^{k} a_i m^i$ with $a_i \in \{0,1,\ldots,m-1\}$, we set, for $i = 1,2,\ldots,k$,

$$\alpha_i = \min\left\{ a_i,\ \left\lceil \frac{m-1}{m^i-1} \sum_{j=0}^{i-1} a_j m^j \right\rceil \right\}.$$

Now, we claim that

$$f_m(l) = \frac{1}{2}\sum_{i=0}^{k} a_i \left(m^{i+1} - m + a_i - 1\right) + \sum_{i=1}^{k} \alpha_i.$$

We proceed by induction on k. For $k = 0$, we have $l = a_0 < m$ and the vertices $0, 1, \ldots, l-1$ form a clique, hence

$$f_m(l) = \binom{l}{2} = \frac{1}{2}l(l-1) = \frac{1}{2}a_0\left(m^{0+1} - m + a_0 - 1\right),$$

as required. For $k \geq 1$, the graph on the first l vertices consists of the following parts:

- a_k copies of $S(k, m)$ and $\binom{a_k}{2}$ edges between these copies,
- a copy of the graph on the first l_k vertices of $S(k, m)$ and α_k edges between this incomplete $S(k, m)$ and the a_k complete copies.

Using the induction hypotheses for $f_m(l_k)$, we conclude

$$\begin{aligned} f_m(l) &= a_k E(S(k,m)) + \binom{a_k}{2} + f(l_k) + \alpha_k \\ &= a_k\frac{(m^k - m)}{2} + \frac{a_k(a_k-1)}{2} + \frac{1}{2}\sum_{i=0}^{k-1} a_i\left(m^{i+1} - m + a_i - 1\right) + \sum_{i=1}^{k-1}\alpha_i + \alpha_k \\ &= \frac{1}{2}\sum_{i=0}^{k} a_i\left(m^{i+1} - m + a_i - 1\right) + \sum_{i=1}^{k}\alpha_i. \end{aligned}$$

As a consequence of the above algorithm, we have the following results.

Theorem 3.4 *If $f_m(l)$ is optimal on l-vertices in $S(n, m)$, $1 \leq l \leq m^n$, then $f_m(m^n - l)$ is also optimal on $m^n - l$ vertices, by considering $M' = \{y = m^n - x + 1$ / for all $x \in f_m(m^n - l)\}$.*

Time Complexity Analysis of Maximum Subgraph Algorithm :

The maximum subgraph problem algorithm aims to compute the maximum number of edges induced by l vertices in $S(n, m)$, where $1 \leq l \leq m^n$, $n \geq 2$ and $m \geq 3$. The algorithm's time complexity can be analyzed as follows:

Input: The generalized Sierpinski graphs, $S(n, m),\ n \geq 2,\ m \geq 3$.

Algorithm: Maximum subgraph algorithm.

Output: Time taken to compute the maximum number of edges induced by l vertices in $S(n, m)$.

Method: Since the graph $S(n, m)$ has m^n vertices, to label them using lexicographic ordering, we need m^n time units, and to find the maximum number of edges for each l-segments, $1 \leq l \leq m^n$, we need another m^n time units. Therefore, the total time taken can be given as $2m^n$. Hence, the time taken to compute the maximum number of edges induced by l vertices in $S(n, m)$ is $O(m^n)$.

4 MinLA of Generalized Sierpinski Graphs

Finding the wirelength for embedding irregular graphs into any graph can be a difficult task. In this part, we made a significant advancement by using Generalised Sierpinski graphs as a guest graph and path as a host graph, and by doing this, we were able to determine its wirelength (also known as MinLA). The MinLA problem is an example of a combinatorial optimization problem, with the goal of obtaining a linear layout of an arrangement in such a manner that some objective cost function is maximized. In 1964, Harper first brought attention to this problem [11]. Harper aimed to create error-correcting codes with low average absolute errors for a certain category of graphs. Later, Mitchison and Durbin looked into this problem as an oversimplified model of certain cognitive nerve movements. Many different areas may benefit from MinLA, including those dealing with single machines, job scheduling, biology, graph drawing, storage-time product reduction issues, and other fields. Combinatorial optimization problem MinLA is stated mathematically as follows: For an undirected graph $X = (V, E)$, find a layout $\mathcal{T}$ that minimizes $\sum_{(u,v)\in E} |\mathcal{T}(u) - \mathcal{T}(v)|$. It is proved that MinLA is NP-complete [9]. This problem is also called optimum linear ordering, edge sum problem, and minimum-1-sum. Vertices represent modules and edges indicate interconnections in this abstract concept of VLSI layout arrangement. The total wirelength provides a measure of the setup cost. MinLA is used in single-machine task scheduling, storage-time product reduction, biological applications, graph drawing, reordering big space matrices, and other domains [8].

MinLA Algorithm :

Input: The generalized Sierpinski graphs, $S(n, m),\ n \geq 2,\ m \geq 3$, on m^n vertices and a path P_{m^n}.

Algorithm: Label the $S(n, m)$ vertices from 1 to m^n in lexicographic order. Label the vertices of P_{m^n} as $1, 2, \ldots, m^n$ from left to right.

Output: An embedding $\mathcal{T}$ of $S(n, m)$ into a path P_{m^n} given by $\mathcal{T}(u) = u$ inducing MinLA $(S(n, m))$.

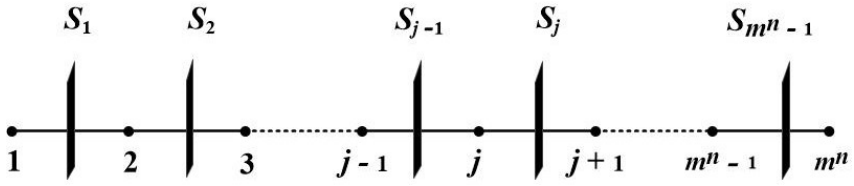

Figure 3: Convex cut S_j in P_{m^n}.

Proof of correctness: Each edge $(j,\ j+1),\ 1 \leq j \leq m^n - 1$ of P_{m^n} is a convex cut S_j and its removal disconnects P_{m^n} into two constitutes Y_{j1} and Y_{j2}, where $V(Y_{j1}) = \{1, 2, \ldots, j\}$ and $V(Y_{j2}) = \{j+1, \ldots, m^n\}$. Let X_{j1} and X_{j2} be the subgraph induced by $\mathcal{T}^{-1}(V(Y_{j1}))$ and $\mathcal{T}^{-1}(V(Y_{j2}))$ respectively. Since the labeling pattern of the $S(n, m)$ is lexicographic and by Theorem 3.3, it is clear that $E(X_{j1})$ is the maximum subgraph on $|V(X_{j1})| = j$ vertices in X and by Theorem 3.4, $E(X_{j2})$ is the maximum subgraph on $|V(X_{j2})| = m^n - j$ vertices in X. Thus, each S_j satisfies the condition of the Lemma 2.8. Therefore, $C_{\mathcal{T}}(S_j)$ is minimum. Lemma 2.9, implies that the linear arrangement is minimum.

Theorem 4.1 *Let X be a generalized Sierpinski graphs $S(n,m),\ n \geq 2,\ m \geq 3$. Then the minimum linear arrangement of X is given by*

$$MinLA(X) = \frac{m+1}{6(m-1)}\left[(m-1)(m-2)\times m^n \times n + m^{n+1} - m\right].$$

Proof: Label the vertices of $S(n,m)$ and P_{m^n} using MinLA Algorithm. Again by the MinLA Algorithm, the congestion on each S_j, $1 \leq j \leq m^n - 1$ is minimum and is given by

$$C_{\mathcal{T}}(S_j) = mi - j\ - 2I_X(i),\ 1 \leq j \leq m-1,\ (j-1)\left(\frac{m^n-1}{m-1}\right)+1 \leq i \leq j\left(\frac{m^n-1}{m-1}\right)$$

Then by Partition Lemma, we have

$$MinLA(X) = \sum_{j=1}^{m-1} \sum_{i=(j-1)\left(\frac{m^n-1}{m-1}\right)+1}^{j\left(\frac{m^n-1}{m-1}\right)} \left[mi - j\ - 2I_X(i)\right]$$

$$= \sum_{j=1}^{m-1} \sum_{i=(j-1)\left(\frac{m^n-1}{m-1}\right)+1}^{j\left(\frac{m^n-1}{m-1}\right)} \left(mi - j\right) - 2\sum_{j=1}^{m-1} \sum_{i=(j-1)\left(\frac{m^n-1}{m-1}\right)+1}^{j\left(\frac{m^n-1}{m-1}\right)} I_X\left(i\right)$$

$$= \frac{1}{2}m\left(m^n-1\right)^2 - 2\sum_{j=1}^{m-1} \sum_{i=(j-1)\left(\frac{m^n-1}{m-1}\right)+1}^{j\left(\frac{m^n-1}{m-1}\right)} I_X\left(i\right).$$

$$= \frac{m+1}{6(m-1)}\left[(m-1)(m-2)\times m^n \times n + m^{n+1} - m\right].$$

□

Time Complexity Analysis of the MinLA Algorithm :

The MinLA algorithm aims to compute the MinLA of generalized Sierpinski graphs $S(n,m)$, where $n \geq 2$ and $m \geq 3$. The algorithm's time complexity can be analyzed as follows:

Input: The generalized Sierpinski graphs, $S(n,m),\ n \geq 2,\ m \geq 3$ and P_{m^n}.

Algorithm: MinLA Algorithm.

Output: Time taken to compute the MinLA of generalized Sierpinski graph $S(n,m)$.

Method: The input graph $S(n,m)$ contains m^n vertices, so the total time for labeling all m^n vertices is m^n time units. The MinLA algorithm performs $m^n - 1$ edge cuts. Each cut operation

takes one unit of time. Additionally, the algorithm computes the edge congestion, which also takes one unit of time for each cut in the path. Thus, the total time for edge congestion calculations is $m^n - 1$ time units. Finally, by using Lemma 2.9, one unit of time is required to calculate the wirelength. Therefore, the total time taken can be given as $m^n + 2(m^n - 1) + 1$. Hence, to compute the exact MinLA of $S(n, m)$, the time taken is $O(m^n)$.

5 Real-World Applications

Graph theory is a fascinating and essential field of mathematics that deals with the study of networks, which are represented by mathematical models called graphs. Graph theory has numerous practical applications in various fields such as computer science, VLSI circuit design, and network optimization for parallel computer architecture. One of the essential tools for implementing parallel algorithms efficiently is graph embedding, which is calculated using cost criteria such as dilation, congestion, wirelength, load, and expansion. Graph embedding is crucial in minimizing wirelength, which is a critical factor in reducing the cost of implementing interconnected networks in VLSI circuit design. The minimum linear arrangement problem, also known as the minimum

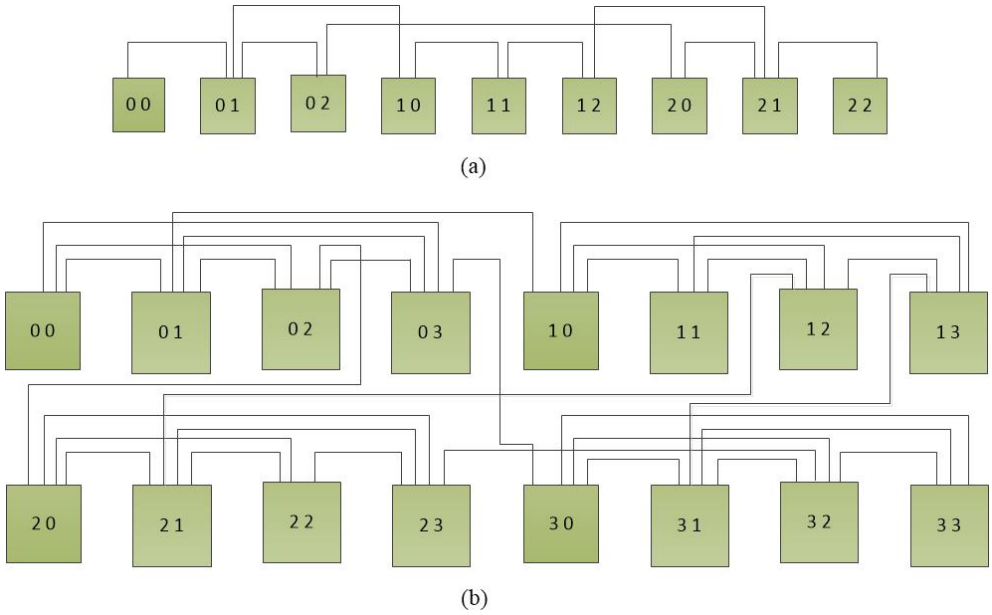

Figure 4: Layouts of (*a*) $S(2, 3)$, (*b*) $S(2, 4)$.

linear ordering problem, is a well-studied problem in graph theory and combinatorial optimization. It involves finding a linear ordering of the vertices of a graph that minimizes a certain objective function. This problem has applications in various fields [6, 18]. Here are a few examples:

- VLSI Design: In VLSI layout design, the MinLA problem is used to optimize the placement of components or modules on a chip. By finding an optimal linear ordering of the modules, the problem aims to minimize the total wirelength or signal delay in the integrated circuit. This, in turn, improves the overall performance and efficiency of the VLSI system.

- Network Routing: In communication networks, the MinLA problem is relevant for optimizing the routing paths between nodes. By finding an optimal linear ordering of the network nodes, the problem helps minimize the total transmission delay or congestion in the network. This is crucial in improving the efficiency and performance of data routing in various network architectures, including computer networks and telecommunication systems.

- Data Clustering: The MinLA problem has applications in data clustering and visualization. By arranging data points linearly based on their similarities or dissimilarities, the problem aids in identifying meaningful patterns or clusters in the data. This helps in data exploration, visualization, and understanding of complex datasets in fields such as machine learning, data mining, and pattern recognition.

Indeed, the problem of minimizing the total edge length and the problem of reducing the number of crossing edges are important in graph drawing and visualization. While they are distinct problems, there is a relationship between them that can be exploited to approximate the crossing number of non-planar graphs. In graph drawing, the MinLA problem focuses on finding a layout of the graph that minimizes the total length of the edges. By optimizing the edge lengths, we aim to improve the comprehensibility and readability of the graph representation. Various algorithms and techniques have been developed to tackle the MinLA problem, including force-directed algorithms, spectral methods, and hierarchical approaches [8].

On the other hand, the crossing number problem deals with determining the minimum number of edge crossings in a graph drawing. A crossing occurs when two edges intersect or cross over each other. Minimizing the number of crossings is desirable as it enhances the clarity and reduces visual clutter in the graph, See Figure 4. However, finding the exact crossing number of a graph is a challenging task and is known to be NP-hard [17]. One approach is to use the solutions of the MinLA problem as an approximation for the crossing number problem. By minimizing the total edge length in the graph drawing, we indirectly reduce the number of crossings, as shorter edges are less likely to intersect with others. While this approximation provides a practical way to estimate the crossing number, it is important to note that it may not always yield an optimal solution. There might be cases where a layout with minimal edge length does not necessarily correspond to the layout with the fewest crossings. However, exploiting the MinLA problem can still provide valuable insights and improvements in graph drawing, contributing to better visual representations of non-planar graphs.

6 Concluding Remarks

In this paper, we have obtained the maximum number of edges induced by any $l,\ 1 \leq l \leq m^n$ of the generalized Sierpinski graphs $S(n,m),\ n \geq 2,\ m \geq 3$. Further, we have obtained its minimum linear arrangement. Finding the wirelength of embedding generalized Sierpinski graphs into other good candidates for interconnection networks is under investigation.

Acknowledgements

The authors thank Prof. Thomas Kalinowski, University of New England (UNE) Australia, for his continuous support and encouragement throughout the preparation of this paper. His comments to improve the quality of this paper are greatly appreciated. Further, we would like to thank the referees for their comments and suggestions which were very helpful for improving the quality of this paper.

References

[1] R. Ahlswede and N. Cai. General edge-isoperimetric inequalities, part I: Information-theoretical methods. *European Journal of Combinatorics*, 18(4):355–372, 1997. doi:10.1006/eujc.1996.0105.

[2] M. Al-Jubeh, M. Hoffmann, M. Ishaque, D. L. Souvaine, and C. D. Tóth. Convex partitions with 2-edge connected dual graphs. *Journal of Combinatorial Optimization*, 22(3):409–425, 2011. doi:10.1007/s10878-010-9310-1.

[3] M. Arockiaraj, P. Manuel, I. Rajasingh, and B. Rajan. Wirelength of 1-fault hamiltonian graphs into wheels and fans. *Information Processing Letters*, 111(18):921–925, 2011. doi:10.1016/j.ipl.2011.06.011.

[4] S. L. Bezrukov. Edge isoperimetric problems on graphs. *Graph Theory and Combinatorial Biology*, 7:157–197, 1999.

[5] S. L. Bezrukov, J. D. Chavez, L. H. Harper, M. Röttger, and U. P. Schroeder. Embedding of hypercubes into grids. In *Mathematical Foundations of Computer Science 1998: 23rd International Symposium, MFCS'98 Brno, Czech Republic, August 24–28, 1998 Proceedings 23*, pages 693–701. Springer, 1998. doi:10.1007/BFb0055820.

[6] A. Caprara, M. Oswald, G. Reinelt, R. Schwarz, and E. Traversi. Optimal linear arrangements using betweenness variables. *Mathematical Programming Computation*, 3:261–280, 2011. doi:10.1007/s12532-011-0027-7.

[7] J. G. Carlsson, B. Armbruster, and Y. Ye. Finding equitable convex partitions of points in a polygon efficiently. *ACM Transactions on Algorithms (TALG)*, 6(4):1–19, 2010. doi:10.1145/1824777.1824792.

[8] J. Díaz, J. Petit, and M. Serna. A survey of graph layout problems. *ACM Computing Surveys (CSUR)*, 34(3):313–356, 2002. doi:10.1145/568522.568523.

[9] M. R. Garey. Computers and intractability: A guide to the theory of NP-completeness, freeman. *Fundamental*, 1997.

[10] L. Harper. The edge-isoperimetric problem on Sierpinski graphs: Final resolution. *arXiv preprint arXiv:1802.08355*, 2018.

[11] L. H. Harper. Optimal assignments of numbers to vertices. *Journal of the Society for Industrial and Applied Mathematics*, 12(1):131–135, 1964. doi:10.1137/0112012.

[12] L. H. Harper. Can the Sierpinski graph be embedded in the Hamming graph? *arXiv preprint arXiv:1609.06777*, 2016.

[13] A. M. Hinz, S. Klavžar, and S. S. Zemljič. A survey and classification of Sierpiński-type graphs. *Discrete Applied Mathematics*, 217:565–600, 2017. doi:10.1016/j.dam.2016.09.024.

[14] S. Klavzar. Coloring Sierpiński graphs and Sierpiński gasket graphs. *Taiwanese Journal of Mathematics*, 12(2):513–522, 2008. doi:10.11650/twjm/1500574171.

[15] S. Klavžar and U. Milutinović. Graphs $S(n, k)$ and a variant of the Tower of Hanoi problem. *Czechoslovak Mathematical Journal*, 47:95–104, 1997. doi:10.1023/A:1022444205860.

[16] S. Klavžar, U. Milutinović, and C. Petr. 1-perfect codes in Sierpiński graphs. *Bulletin of the Australian Mathematical Society*, 66(3):369–384, 2002. doi:10.1017/S0004972700040235.

[17] S. Klavžar and B. Mohar. Crossing numbers of Sierpiński-like graphs. *Journal of Graph Theory*, 50(3):186–198, 2005. doi:10.1002/jgt.20107.

[18] Y.-L. Lai and K. Williams. A survey of solved problems and applications on bandwidth, edgesum, and profile of graphs. *Journal of Graph Theory*, 31(2):75–94, 1999.

[19] C.-H. Lin, J.-J. Liu, Y.-L. Wang, and W. C.-K. Yen. The hub number of Sierpiński-like graphs. *Theory of Computing Systems*, 49(3):588–600, 2011. doi:10.1007/s00224-010-9286-3.

[20] P. Manuel, I. Rajasingh, B. Rajan, and H. Mercy. Exact wirelength of hypercubes on a grid. *Discrete Applied Mathematics*, 157(7):1486–1495, 2009. doi:10.1016/j.dam.2008.09.013.

[21] M. Miller, R. S. Rajan, N. Parthiban, and I. Rajasingh. Minimum linear arrangement of incomplete hypercubes. *The Computer Journal*, 58(2):331–337, 2015. doi:10.1093/comjnl/bxu031.

[22] R. S. Rajan, P. Manuel, I. Rajasingh, N. Parthiban, and M. Miller. A lower bound for dilation of an embedding. *The Computer Journal*, 58(12):3271–3278, 2015. doi:10.1093/comjnl/bxv021.

[23] R. S. Rajan, P. D. Manuel, and I. Rajasingh. Embeddings between hypercubes and hypertrees. *J. Graph Algorithms Appl.*, 19(1):361–373, 2015. doi:10.7155/jgaa.00363.

[24] R. Scorer, P. M. Grundy, and C. A. B. Smith. Some binary games. *The Mathematical Gazette*, 28(280):96–103, 1944. doi:10.2307/3606393.

[25] A. A. Shantrinal, S. Klavzar, T. Rajalaxmi, and R. S. Rajan. An algorithm for embedding Turán graphs into incomplete hypercubes with minimum wirelength. *J. Graph Algorithms Appl.*, 25(1):367–381, 2021. doi:10.7155/jgaa.00562.

Appendix

We now present a Sage code with its output to find the maximum number of edges induced by l vertices in $S(n,m),\ 1 \leq l \leq m^n,\ n \geq 2,\ m \geq 3$.

Program to find the maximal sizes for subgraphs on l vertices in $S(n,m)$

```
1. def m_ary(m,l) :
2. if l == 0 :
3. return []
4. x = m_ary(m,l//m)
5. x.append(l%m)
6. return x
7. def f(m,l) :
8. a = m_ary(m,l)
9. k=len(a) − 1
10. alpha=[0]
11. for i in range (1,k + 1) :
12. alpha.append (min(a[k − i], ceil ((m − 1)/(m^i − 1)∗sum(a[k − j] ∗ m^j for j in range (i)))))
13. return 1/2∗sum(a[k − i] ∗ (m^(i + 1) − m + a[k − i] − 1) for i in range (k + 1))+sum( alpha
    [i] for i in range (1,k + 1)
14. print f(m,l) for l in range (1,k + 1)
```

The maximal sizes on l vertices in $S(n, m)$ and its corresponding diagram is given below.

Output 1 ($f(3, l)$ when $n = 5$):

0, 1, 3, 4, 5, 7, 8, 10, 12, 13, 14, 16, 17, 18, 20, 21, 23, 25, 26, 27, 29, 30, 32, 34, 35, 37, 39, 40, 41, 43, 44, 45, 47, 48, 50, 52, 53, 54, 56, 57, 58, 60, 61, 63, 65, 66, 67, 69, 70, 72, 74, 75, 77, 79, 80, 81, 83, 84, 85, 87, 88, 90, 92, 93, 94, 96, 97, 99, 101, 102, 104, 106, 107, 108, 110, 111, 113, 115, 116, 118, 120, 121, 122, 124, 125, 126, 128, 129, 131, 133, 134, 135, 137, 138, 139, 141, 142, 144, 146, 147, 148, 150, 151, 153, 155, 156, 158, 160, 161, 162, 164, 165, 166, 168, 169, 171, 173, 174, 175, 177, 178, 179, 181, 182, 184, 186, 187, 188, 190, 191, 193, 195, 196, 198, 200, 201, 202, 204, 205, 206, 208, 209, 211, 213, 214, 215, 217, 218, 220, 222, 223, 225, 227, 228, 229, 231, 232, 234, 236, 237, 239, 241, 242, 243, 245, 246, 247, 249, 250, 252, 254, 255, 256, 258, 259, 260, 262, 263, 265, 267, 268, 269, 271, 272, 274, 276, 277, 279, 281, 282, 283, 285, 286, 287, 289, 290, 292, 294, 295, 296, 298, 299, 301, 303, 304, 306, 308, 309, 310, 312, 313, 315, 317, 318, 320, 322, 323, 324, 326, 327, 328, 330, 331, 333, 335, 336, 337, 339, 340, 342, 344, 345, 347, 349, 350, 351, 353, 354. 356. 358. 359. 361. 363.

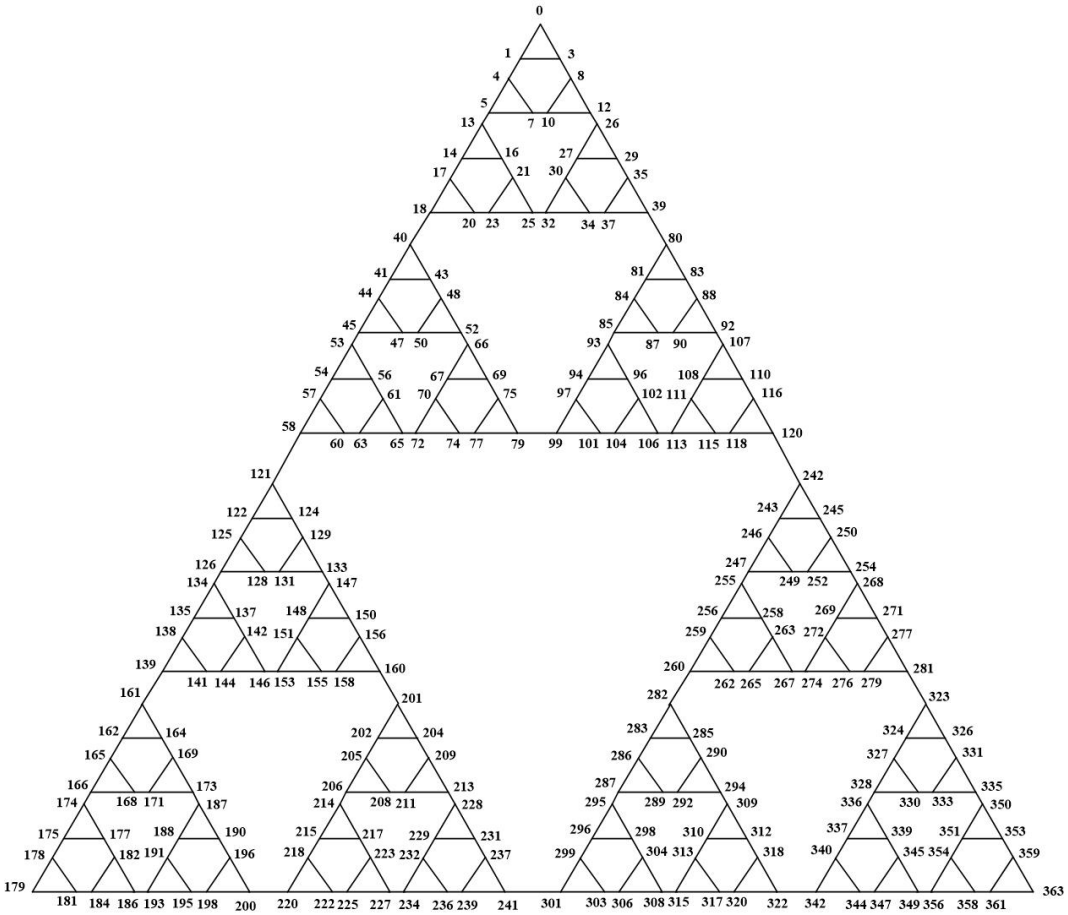

Figure 5: The number of edges induced by l vertices in $S(5, 3)$ using lexicographic labeling.

Output 2 ($f(6,l)$ when $n=3$):

0, 1, 3, 6, 10, 15, 16, 17, 19, 22, 26, 31, 32, 34, 36, 39, 43, 48, 49, 51, 54, 57, 61, 66, 67, 69, 72, 76, 80, 85, 86, 88, 91, 95, 100, 105, 106, 107, 109, 112, 116, 121, 122, 123, 125, 128, 132, 137, 138, 140, 142, 145, 149, 154, 155, 157, 160, 163, 167, 172, 173, 175, 178, 182, 186, 191, 192, 194, 197, 201, 206, 211, 212, 213, 215, 218, 222, 227, 228, 230, 232, 235, 239, 244, 245, 247, 249, 252, 256, 261, 262, 264, 267, 270, 274, 279, 280, 282, 285, 289, 293, 298, 299, 301, 304, 308, 313, 318, 319, 320, 322, 325, 329, 334, 335, 337, 339, 342, 346, 351, 352, 354, 357, 360, 364, 369, 370, 372, 375, 378, 382, 387, 388, 390, 393, 397, 401, 406, 407, 409, 412, 416, 421, 426, 427, 428, 430, 433, 437, 442, 443, 445, 447, 450, 454, 459, 460, 462, 465, 468, 472, 477, 478, 480, 483, 487, 491, 496, 497, 499, 502, 506, 510, 515, 516, 518, 521, 525, 530, 535, 536, 537, 539, 542, 546, 551, 552, 554, 556, 559, 563, 568, 569, 571, 574, 577, 581, 586, 587, 589, 592, 596, 600, 605, 606, 608, 611, 615, 620, 625, 626, 628, 631, 635, 640, 645.

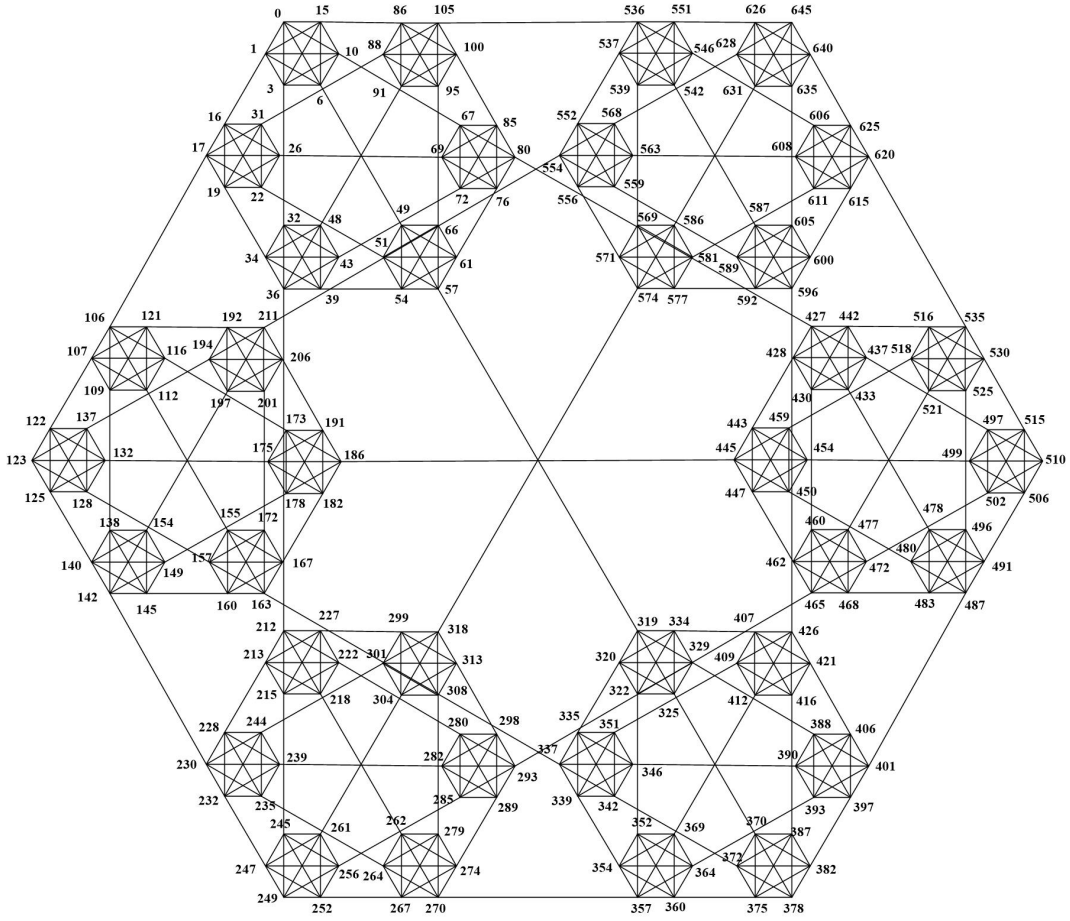

Figure 6: The number of edges induced by l vertices in $S(3,6)$ using lexicographic labeling.