Experimentally finding graphs that minimize Wiener-entropy

ALEKSANDAR PETROV, University of Twente, The Netherlands

Graph entropy is a way to measure the complexity of a graph. There are many different graph entropies, but there is little work done on finding their extremal values. It is especially intriguing to find the minimal values because of the lack of analytical methods to tackle this problem. This paper will focus only on the Wiener entropy and aims to find the trees that minimize the entropy, until the possible computation limit. After that, 2 classes of graphs, referred to as "brooms" and $G_{n,k,j}$, will be inspected and the goal is to find the optimum structure that minimizes the entropy. There is one existing research paper on minimizing Wiener entropy, and the results from this experiments aim to confirm or deny the predictions made there. The experiments will be performed by brute-forcing the graphs that minimize the entropy in the respective search space. In the end, it appears that the class of brooms minimizes the entropy for trees with number of vertices(n) $16 < n < 32$ and is expected to continue to minimize it for bigger trees. Furthermore, the results, for the 2 distinct classes, confirm the conjectures for brooms and discover an error in the predictions for the optimum structure of the $G_{n,k,j}$ class. The paper then defines new conjectures based on the results gathered.

Additional Key Words and Phrases: Wiener entropy, graph entropy, trees, brooms

1 INTRODUCTION

Shannon entropy, named after Claude Shannon, is a fundamental concept in information theory. It provides a measure of the uncertainty of a random variable and it is widely used in various fields, for example in Computer science, Mathematics, and others. Scientists have extended this notion of entropy to graphs. One such graph entropy is called Wiener entropy, which utilizes the transmission of the nodes in the graph, will be studied in this paper.

This paper aims to explore the minimization of Wiener entropy in graphs, focusing on trees but not only. Understanding the properties of graphs with minimum Wiener entropy can have implications in diverse fields where entropy is used. For instance, entropy is used in biology and chemistry [4, 13, 15], in ecological studies [16], and others.

The research on Wiener-entropy is small. Nevertheless, there are several results presented in the research of Yanni Dong & Stijn Cambie [7]. They found asymptotic lower bounds for the entropy of trees, experimentally found minimal trees, and predicted class, called brooms, to be the minimum trees for higher-sized graphs. However, the results are only for small graphs and this paper aims to build on top of those results by brute-forcing the minimum trees and confirming the prediction to a certain extent. Due to the fact, that the number of non-isomorphic trees grows exponentially, $T_n = 0.535(2.959)^n n^{\frac{2}{5}}$ according to [14], brooms are going to be explored by themselves, after confirming that they minimize the Wiener-entropy. The goal of studying them is to confirm the asymptotic bounds and show how the size of the tree is related to the structure of the broom. Furthermore, another class of graphs, derived from brooms but generalized for all graphs and referred to as $G_{n,k,j}$ later, is predicted to minimize the Wiener entropy in general. The paper will also explore them and again try to find the relation between the size and the structure of the graphs. The number of non-isomorphic graphs grows extremely fast with the size of the graph, $2^{\binom{n}{2}}/n!$ according to [8], and therefore only this specific class of graphs is going to be considered. In summary, the research questions, that this paper will answer, are:

- What is the tree T of n vertices that minimizes $I_w(T)$?
- What is the broom T of n vertices that minimizes $I_w(T)$?
- What is the graph $G_{n,k,j}$ that minimize $I_w(G_{n,k,j})$?

2 PRELIMINARIES

2.1 Background

Shannon entropy is a concept in information theory and it is used as a way to measure the uncertainty of a random variable [2, 7, 12]. Let X be a discrete random variable with possible outcomes $x_1, x_2, \ldots, x_n$, and let $P = p(x_1), p(x_2), \ldots, p(x_n)$ be the probability distribution on X. Shannon entropy is defined as:

$$-\sum_{i=1}^{n} p(x_i)\log_2(p(x_i))$$

In his paper [3] Dehmer proposes a general form of graph entropy. Let $G = (V, E)$ be a graph with $V = v_1, v_2, \ldots, v_n$ and E be the set of edges. Given any function $f\colon V \longrightarrow \mathbb{R}_{>0}$, the graph entropy with respect to f is defined as:

$$I_f = -\sum_{i=1}^{n} \frac{f(v_i)}{\sum_{j=1}^{n} f(v_j)} \log_2\left(\frac{f(v_i)}{\sum_{j=1}^{n} f(v_j)}\right)$$

There are many different ways one can define such informational functional but this paper will focus only on one. The distance between two vertices u and v, denoted by $d(u, v)$, is the length of the shortest path from u to v. Wiener-entropy, as defined in [7], uses $f(v_i) = \sigma(v_i)$ where $\sigma(v_i)$ is called *transmission* of a vertex and it is the sum of distances towards all other vertices: $\sigma(v) = \sum_{u\in V} d(v, u)$. The entropy is then defined as:

$$I_w = -\sum_{i=1}^{n} \frac{\sigma(v_i)}{\sum_{j=1}^{n} \sigma(v_j)} \log_2\left(\frac{\sigma(v_i)}{\sum_{j=1}^{n} \sigma(v_j)}\right)$$

2.2 Existing results on Wiener entropy

This section will go over the work that has been done related to Wiener-entropy in graphs.

The main research paper that has done work in the field is by Yanni Dong & Stijn Cambie [7]. In their work, they compare 2 popular natural distance-based graph entropies: eccentricity-based and Wiener-entropy. They derive the extremal behaviour of graphs based on the Wiener-entropy and conclude that Wiener-entropy is more spread than the eccentricity-based. In the derivation of the minimal behaviour of graphs, they prove the asymptotic lower bound for the Wiener-entropy which is:

TScIT 39, July 7, 2023, Enschede, The Netherlands

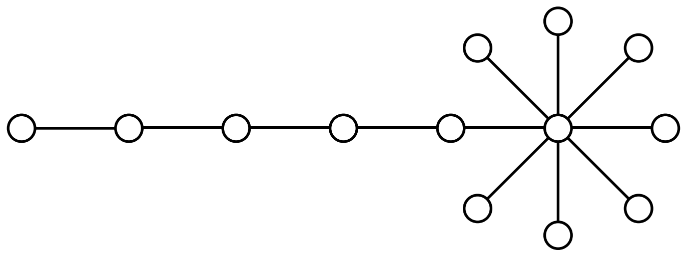

Fig. 1. Example of a broom for $n = 13$ and $k = 6$.

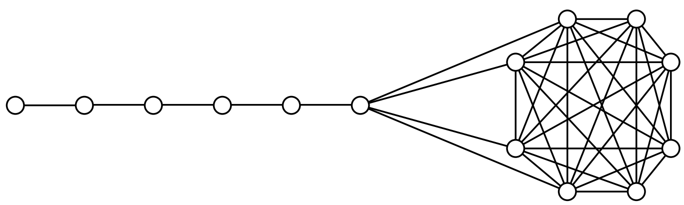

Fig. 2. Example of $G_{n,k,j}$ graph for $n = 14$, $k = 6$,$j = 4$.

$$I_w \geq \frac{3}{4}(1 + o(1)) \log_2(n))$$

Furthermore, in the proof they show that equality is attained by a specific class of trees that they define as "brooms". Therefore they expect the trees of order n, which minimize Wiener-entropy, to be the class of brooms. In this paper, they will be explored further. A broom is defined as a tree consisting of a path P_k with one of its end vertices c connected with $n - k$ pendent vertices. An example of a broom can be seen in Figure 1.

In their paper [7], the authors could not reach a precise answer on the graphs that minimize Wiener-entropy, therefore they only assume that with sufficiently large n, brooms will become those graphs. They brute-forced the graphs with $3 \leq n \leq 18$ and found out that only the graphs with $n = 17$ and $n = 18$ are brooms. Those results are not enough to conclude that brooms minimize the entropy for higher n. In addition, the optimum k is not known, but only asymptotically bounded and this research aims to give a concrete estimation of the optimal k.

The paper [7] also defines a graph $G_{n,k,j}$ which will be also explored in this paper. It comes from a broom but is generalized for all graphs. The conjecture is that this class of graphs would be minimizing the entropy for graphs in general. It is defined as a graph, consisting of a disjoint union of a path P_k and a clique K_{n-k} and one of the end vertices of P_k is connected with j vertices from K_{n-k}. This class of graphs is introduced due to the fact that it is impossible to compute the minimal graph out of all graphs. The number of graphs grows exponentially fast with the size n, shown to be $2^{\binom{n}{2}}/n!$ [8]. An example of $G_{n,k,j}$ can be seen in Figure 2.

3 PROBLEM STATEMENT

As it was mentioned previously, this paper aims to continue the research on trees and graphs that minimize the Wiener entropy. Due to the fact the number of trees grows exponentially with the size of the tree, the paper will also explore minimal brooms by themselves. In addition, $G_{n,k,j}$ graphs will also be explored because the number of graphs grows even faster than the trees and they are also predicted to minimize the Wiener entropy but for graphs in general. The paper [7] experimentally shows that $G_{n,k,1}$ is the minimal graph from that type. However, in their results, they have several assumptions about k and j which this paper aims to remove. The research questions, that are going to be answered in this paper, are:

(1) What is the tree T of n vertices that minimizes $I_w(T)$?
  (a) How can trees for a given n be generated?
  (b) What is the fastest way to calculate the Wiener entropy of a graph?
  (c) What is the fastest way to calculate the minimal Wiener entropy for a tree with n vertices?
(2) What is the broom T of n vertices that minimizes $I_w(T)$?
  (a) How is the length of the path (k) related to the size of the broom (n)?
  (b) How is the entropy ($I_w(T)$) related to the size of the broom(n)?
(3) What is the graph $G_{n,k,j}$ that minimize $I_w(G_{n,k,j})$?
  (a) How is the length of the path (k) related to the size of the graph (n)?
  (b) How is the number of connected nodes from the clique to the path (j) related to the size of the graph (n)?
  (c) How is the entropy ($I_w(G_{n,k,j})$) related to the size of the graph (n)?

4 METHODOLOGY

In this section, the tools and the steps for completing the research will be discussed. For each research question a program had to be implemented.

4.1 Tools

For performing the experiments initially Python was used, mainly because of the available library networkx [9], which makes working with graph structures easy. In addition, it has an implementation of an algorithm for generating all non-isomorphic trees for a given size. However, due to the fact that Python is not a very fast language and the library is written in Python as well, the experiment did not produce enough results. Therefore all the code was rewritten in Rust. Rust was chosen because it provides good performance and an easy way to parallelise the execution of a program. For working with graphs the crate(library) called petgraph was used. For producing the plots, python was still used with the libraries pandas [17], numpy [10] and matplotlib [11]. Most of the code for the experiments was run on the High Performance Cluster(HPC) of the University of Twente, due to the fact that it has modern and advanced hardware. With the use of the HPC, graphs with bigger sizes were able to be calculated and therefore more results were gathered.

4.2 Minimal tree for I_w

For finding the tree that minimizes I_w, 2 algorithms were needed: generating all non-isomorphic trees and calculating the I_w for a tree. After that, a program which combines both was written in order to calculate the minimal tree for a given order. The program was also configurable to provide top x minimal graphs.

4.2.1 Generating non-isomorphic trees. To begin with, the algorithm that was used for generating trees is Wright, Richmond, Odlyzko and McKay (WROM) algorithm [18]. The algorithm builds on top of Bayer and Hedetniemi rooted tree algorithm [1], which generates only rooted trees. Both algorithms use integer sequences to encode trees. The WROM algorithm sequentially generates non-isomorphic trees and that sequentiality restricted the optimizations that were possible in the combined program.

4.2.2 Calculating $I_w(T)$. For calculating the $I_w(T)$, we need the transmission ($\sigma(v)$) for all $v \in T$. The Dijkstra algorithm [5] was used on each vertex in order to obtain all of the shortest paths in the graph. The rewritten formula for I_w, derived in [7], was used in order to optimize the performance. It improved the speed of the calculation because it contains considerably fewer division operations and those operations are expensive in computation. It uses the Wiener index which is defined as:

$$W(G) = \sum_{v_i, v_j \in V} d(v_i, v_j)$$

The rewritten formula for $I_w(T)$:

$$I_w = \log_2(2W(G)) - \frac{1}{2W(G)} \sum_{i=1}^{n} \sigma(v_i) \log_2 \sigma(v_i)$$

4.2.3 Combining both parts to calculate the minimum tree. Making a program for calculating the minimum tree was fairly simple. However, 2 modifications were made. First, a producer-consumer pattern was used in order to improve performance. Producer-consumer pattern [6] is a concurrency design pattern, where there are multiple threads(producers) which produce objects and put them into a shared queue, from where then multiple threads(consumers) can get the objects and perform a task with them. It appeared that the generation of trees was faster than the calculation of $I_w(T)$. Therefore, it was beneficial to have multiple CPUs, computing in parallel the calculation of the entropy, while other CPUs are generating trees. There were only 2 producers because the WROM algorithm is sequential and therefore not possible to parallelize. However, it was observed that with 2 producers the program was significantly faster. The second modification was to add a configuration that outputs not only the minimal tree. Instead of keeping track only of the minimal graph, the program takes an argument which defines how many graphs to save.

4.3 Minimal broom for I_w

For calculating the minimal broom, the search space is reduced to only broom graphs. The method, in which the search for the minimum broom was performed, was looping through all the brooms with n vertices and changing the number of vertices that are on the path part of the broom (changing the k). Because of the fact that the structure of the graph is known, it was possible to calculate the entropy without using graph objects in the code but by implementing a clever calculation. This was considerably faster than creating a graph object, then running the Dijkstra algorithm on each node and then calculating the entropy. A pseudo-code of that calculation can be seen below at Algorithm 1. Additionally, a version which utilizes parallelization was implemented in order to compute brooms up to $n = 125\,000$. The parallelization was not in the calculation, but making different CPUs to calculate minimal broom with different n.

Algorithm 1 Calculate wiener entropy of a broom

function CALCULATE_ENTROPY(n, k)
  $transmission_{star} \leftarrow k(k+1)/2 + 2(n-k-1)$
  $transmissions_{path} \leftarrow 0$
  $entropy_{path} \leftarrow 0$
  for i *in* $1 \ldots k$ **do**
    $t \leftarrow ((k-i)(k-i+1)/2 + i(i-1)/2 + i(n-k))$
    $transmissions_{path}$ += t
    $entropy_{path}$ += $t * \log_2(t)$
  end for
  $WienerIndex \leftarrow (n-k)transmission_{star} + transmissions_{path}$
  $entropy \leftarrow \log_2(WienerIndex) - (1/WienerIndex)((n-k)transmission_{star} * \log_2(transmission_{star}) + entropy_{path})$
  return *entropy*
end function

4.4 Minimal $G_{n,k,j}$ graph for I_w

A similar approach to calculating the minimal broom was applied here. The difference was that in the search, the program had to loop through all pairs of k and j for a given n. Unlike the broom solution, here the number of pairs (k, j) was significantly more and therefore parallelization in the calculation was utilized. A different thread was spawned for each k and each thread calculated the entropy for all possible j values. This technique appeared to improve the performance. The calculation of I_w is different but the approach stayed the same because again the structure of the graph is known beforehand, therefore a clever calculation was possible and the object-orientated approach could be skipped. The algorithm can be seen below at Algorithm 2.

Algorithm 2 Calculating wiener entropy of a $G_{n,k,j}$ graph

function CALCULATE_ENTROPY(n,k,j)
  $transmision_{not\_con} \leftarrow ((n-k-1) + (k(k+1))/2 + k)$
  $transmission_{con} \leftarrow ((n-k-1) + (k(k+1))/2)$
  $transmissions_{path} \leftarrow 0$
  $entropy_{path} \leftarrow 0$
  for i *in* $1 \ldots k+1$ **do**
    $t = ((k-i)(k-i+1)/2 + i(i-1)/2 + ij + (n-k-j)(i+1))$
    $transmissions_{path}$ += t
    $eentropy_{path}$ += $t * \log_2(t)$
  end for
  $WienerIndex \leftarrow (n-k-j) * transmission_{not\_con} + j * trnsmission_{con} + transmissions_{path}$
  $entropy \leftarrow \log_2(WienerIndex) - (1/WienerIndex)((n-k-j) * transmission_{not\_con} * \log_2(transmission_{not\_con})) + j * transmission_{con} * \log_2(transmission_{con}) + entropy_{path})$
  return *entropy*
end function

5 RESULTS

In this section, the results for each research question will be presented and discussed. For brooms and $G_{n,k,j}$ graph, logarithmic linear regression was applied to find a function that approximates k well. The way it was done was using the data gathered, finding a and b such that $\log_2(k) = a\log_2(n) + b$ and then using them to approximate k with the equation $k = 2^b n^a$. The code for generating the trees and the plots can be found at https://gitlab.utwente.nl/s2615576/research-project.

5.1 Minimal trees

Brute-forcing the tree that minimizes $I_w(T)$ took exponentially more time as the number of vertices of the tree grew. There are 2 reasons for that. First, the number of trees exponentially increases with n. In the past [14] proved the asymptotic estimate for the total number of (unlabeled) trees to be: $T_n = 0.535(2.959)^n n^{\frac{2}{5}}$. Secondly, the calculation of the entropy for each tree is slower for bigger trees. In the end, in this experiment minimal trees up until $n = 31$ were able to be computed, the last one taking several days. The numbers of non-isomorphic trees for $n = 30$ is $1.4 \cdot 10^{10}$, $n = 31$ is $4 \cdot 10^{11}$ and for $n = 32$ is $1.1 \cdot 10^{12}$ for which it would take nearly a week to compute the minimum graph.
From the experiment, all of the minimal trees with $n > 16$ turned out to be brooms. Therefore Conjecture 1 was made. Furthermore, Conjecture 2 was made because for trees with $n \geq 19$, the second minimal tree is also a broom and for trees with $n \geq 25$ 3 out of the top 4 minimal trees are brooms. More results, supporting Conjecture 2 can be seen in Table 1.

Conjecture 1. *Among all trees where $n > 16$, the Wiener entropy is minimized by a broom.*

Conjecture 2. *For a given m, there exists an n_0 such that for all $n > n_0$ the top m minimal trees are brooms.*

m	n_0
1	17, 18
2	19, 20, 21, 22, 23, 24, 26, 27, 28, 31
3	25, 29, 30

Table 1. Showing results supporting Conjecture 2

The relation between the length of the path of the broom(k) and the size of the broom(n) will be discussed in the next section.

5.2 Minimal brooms

Restricting the search space only to brooms led to significantly more results. In the end, the minimal brooms with size $16 \leq n < 125\ 000$ were calculated. The relation between n and k seems to confirm the proposition by [7] and the results can be seen in Figure 3. Using logarithmic linear regression, the optimal choice for k seems to be: $k \approx 1.814n^{0.568}$.
However, the prediction for the entropy in [7] seems to not be perfectly matching. First of all, there is a mistake in the formula

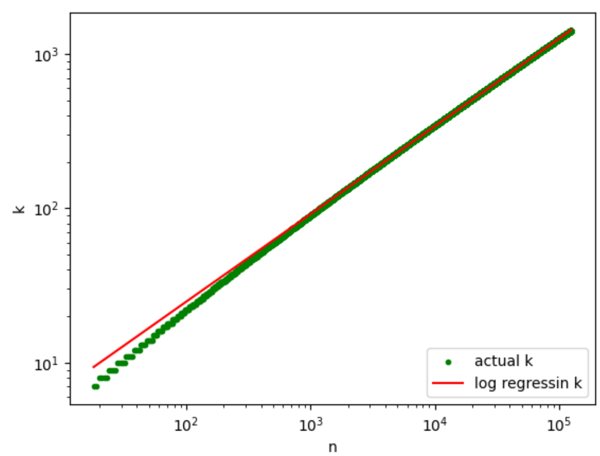

Fig. 3. Relation between broom size(n) and tail length(k).

of their Proposition 29. With the correction, for $0 < \epsilon < \frac{1}{3}$ the asymptotic estimation for the entropy is:

$$I_w(G) = \frac{3 + 2\epsilon}{4} \log_2(n)$$

With that correction, the lower and upper bounds were more accurate and the actual results were in between those bounds. The lower and upper bounds were calculated by taking 0 and $\frac{1}{3}$ for ϵ respectively. The prediction was made by calculating the ϵ from the formula for the optimal choice for k. The value that was substituted was $\epsilon = 0.168$.

Although the entropy was not matching perfectly, with more results gathered we can see that the entropy slowly getting closer to the predicted value. The prediction in the paper [7] is an asymptotic prediction with $n \to \infty$, therefore with enough results, it might reach the prediction. A plot can be seen in Figure 4.

5.3 Minimal $G_{n,k,j}$

Conjecture 3. *There exists a value n_0 such that for all $n \geq n_0$, among graphs of order n, the Wiener-entropy is minimized by $G_{n,k,1}$ and/or $G_{n,k,n-k}$.*

For the graphs $G_{n,k,j}$ the paper [7] predicted that after a certain $n > n_0$, $j = 1$ to be the minimum graph. However, the paper [7] missed the possibility of having 2 different graphs having equal entropy. From the results gathered, graphs with higher order tend to have 2 minimal graphs, where $j = 1$ with a certain k_0 is one of them and the second one is with $k = k_0 + 1$ and $j = n - k$. It should be noted that $j = n - k$ is the maximum value that j can attain. Because of that, this paper proposes slightly changed conjecture seen at 3. The n_0 appears to be the same as in the paper [7], being $n_0 = 1270$.

Another interesting result is that for bigger graphs, the fraction of graphs with 2 minimal entropies tends to be increasing. However, it should be noted that there are still graphs for which there is only 1 minimum. This observation can be seen in Figure 6.

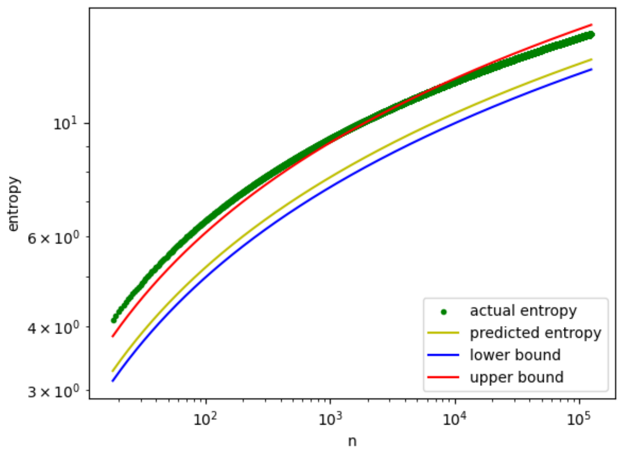

Fig. 4. Relation between the size of the broom(n) and the entropy of the broom.

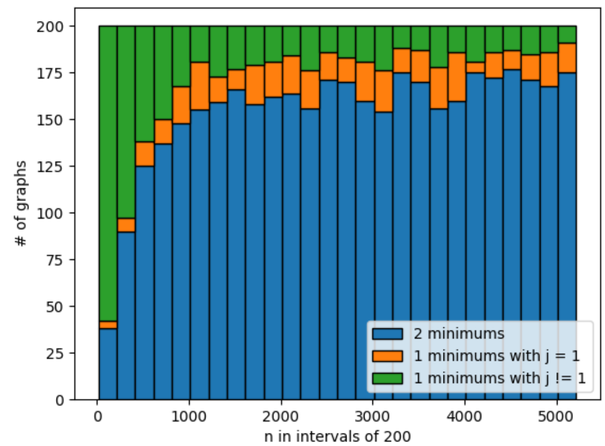

Fig. 6. The distribution of 1 and 2 minimums in $G_{n,k,j}$

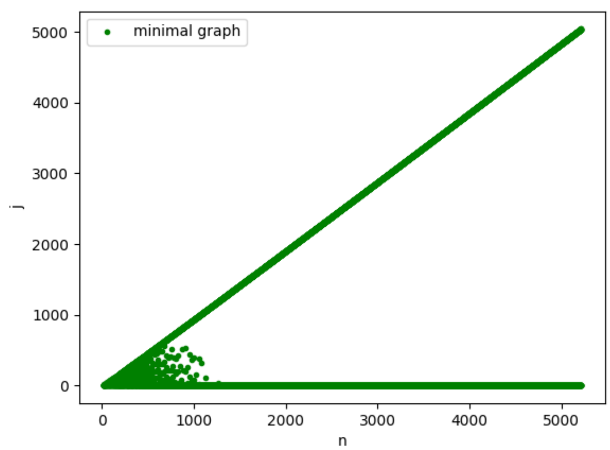

Fig. 5. Relation between the number of connected nodes of the clique to the path(j) and the size of the graph(n).

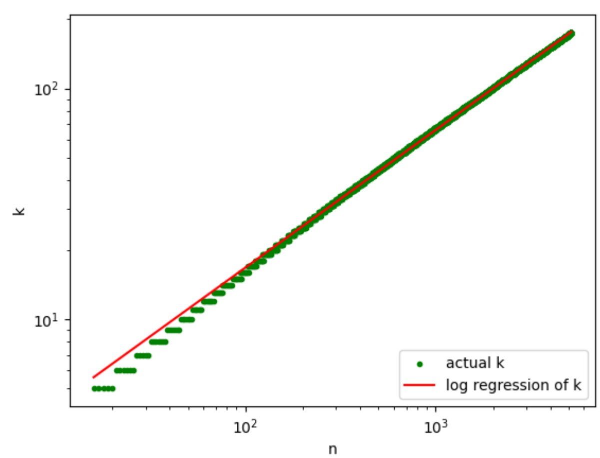

Fig. 7. Relation between graph size(n) and path length(k) for $G_{n,k,j}$.

In Figure 6 can be also seen that for bigger graphs which had 1 minimum, the number of graphs that have $j = 1$ and $j \neq 1$ is almost the same.

The optimal choice for k, considering all minimal graphs, again seems to be linear on a logarithmic scale therefore again logarithmic linear regression was applied. The exponent seems to be very close to the one for brooms but the coefficient differs significantly. The result can be seen in Figure 7 and the formula is:

$$k \approx 1.07n^{0.597}$$

6 CONCLUSION

This research has analyzed the trees and graphs that minimize Wiener entropy. It found that for trees with number of vertices $16 < n < 32$, the minimal graphs are brooms and this trend is expected to continue. Furthermore, the relation between the length of the path(k) and the size of the broom(n) for the minimal broom was examined and was found to be log-linear. In addition, a certain graph class, $G_{n,k,j}$, was researched because it is expected to minimize the Wiener entropy for all graphs. The relation between the length of the path(k) and the size of the graph(n) was again examined. An interesting result that was found is that for bigger sizes, 2 minimal graphs were more common. In addition, j was taking the values of either 1 or the maximum possible value with the optimum k.

7 FUTURE WORK

For future research, trees with more than 31 nodes can be explored. As the number of trees becomes enormous, different generation of trees can be a point of research. There might be a way to not look through every possible tree, but to find a structure of trees that are definitely not minimal and can be skipped.

Another possible future research can be to extend the class of $G_{n,k,j}$ graphs by allowing for multiple paths connecting to the clique and not forcing the clique to be at the end of a path. The extended class should be still computable and it would support or contradict the prediction that $G_{n,k,j}$ are minimizing the entropy for graphs.

ACKNOWLEDGMENTS

I want to express my deep gratitude to my supervisor - Milan Lopuhaä-Zwakenberg for their invaluable guidance, support, and feedback throughout this Bachelor Thesis. In addition, I want to thank Yanni Dong for her propositions for the focus of this research and for providing me with literature on graph entropy.

REFERENCES

[1] Terry Beyer and Sandra Mitchell Hedetniemi. 1980. Constant Time Generation of Rooted Trees. *SIAM J. Comput.* 9, 4 (1980), 706–712. https://doi.org/10.1137/0209055 arXiv:https://doi.org/10.1137/0209055

[2] Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience.

[3] Matthias Dehmer. 2008. Information processing in complex networks: Graph entropy and information functionals. *Appl. Math. Comput.* 201, 1 (2008), 82–94. https://doi.org/10.1016/j.amc.2007.12.010

[4] Matthias Dehmer and Abbe Mowshowitz. 2011. A history of graph entropy measures. *Information Sciences* 181, 1 (2011), 57–78. https://doi.org/10.1016/j.ins.2010.08.041

[5] Edsger W Dijkstra. 1959. A note on two problems in connexion with graphs. *Numerische mathematik* 1, 1 (1959), 269–271.

[6] Edsger W. Dijkstra. 2002. *Cooperating Sequential Processes*. Springer New York, New York, NY, 65–138. https://doi.org/10.1007/978-1-4757-3472-0_2

[7] Yanni Dong and Stijn Cambie. [n. d.]. On the main distance-based entropies: the eccentricity- and Wiener-entropy. ([n. d.]). arXiv:2208.12209 [math.CO] preprint on webpage at https://arxiv.org/pdf/2208.12209.pdf.

[8] G W Ford and G E Uhlenbeck. 1957. COMBINATORIAL PROBLEMS IN THE THEORY OF GRAPHS. IV. *Proc Natl Acad Sci U S A* 43, 1 (Jan. 1957), 163–167.

[9] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. 2008. Exploring Network Structure, Dynamics, and Function using NetworkX. In *Proceedings of the 7th Python in Science Conference*, Gaël Varoquaux, Travis Vaught, and Jarrod Millman (Eds.). Pasadena, CA USA, 11 – 15.

[10] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature* 585, 7825 (Sept. 2020), 357–362. https://doi.org/10.1038/s41586-020-2649-2

[11] J. D. Hunter. 2007. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering* 9, 3 (2007), 90–95. https://doi.org/10.1109/MCSE.2007.55

[12] David J. C. MacKay. 2003. *Information Theory, Inference, and Learning Algorithms*. Copyright Cambridge University Press.

[13] Harold J. Morowitz. 1955. Some order-disorder considerations in living systems. *The bulletin of mathematical biophysics* 17, 2 (01 Jun 1955), 81–86. https://doi.org/10.1007/BF02477985

[14] Richard Otter. 1948. The Number of Trees. *Annals of Mathematics* 49, 3 (1948), 583–599. http://www.jstor.org/stable/1969046

[15] Nicolas P. Rashevsky. 1955. Life, information theory, and topology. *Bulletin of Mathematical Biology* 17 (1955), 229–235.

[16] William B Sherwin and Narcis Prat i Fornells. 2019. The Introduction of Entropy and Information Methods to Ecology by Ramon Margalef. *Entropy* 21, 8 (2019). https://doi.org/10.3390/e21080794

[17] Wes McKinney. 2010. Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference*, Stéfan van der Walt and Jarrod Millman (Eds.). 56 – 61. https://doi.org/10.25080/Majora-92bf1922-00a

[18] Robert Alan Wright, Bruce Richmond, Andrew Odlyzko, and Brendan D. McKay. 1986. Constant time generation of free trees. *SIAM J. Comput.* 15, 2 (1986), 540–548. https://doi.org/10.1137/0215039