Exploring Novel Graphs with Diverse Properties for Real-World Applications

A. Javan[1] and A. Moeini[2*]

[1,2] Department of Algorithms and Computation, School of Engineering Sciences, College of Engineering, University of Tehran, 16 Azar, Tehran, 1417614411, Tehran, Iran

Corresponding author: A. Moeini, moeini@ut.ac.ir, +989122190659

Contributing author: A. Javan, ajavan@ut.ac.ir, +989120147231

Abstract: Graphs are versatile mathematical models used to solve a wide range of practical and mathematical problems. However, meeting the specific requirements of applications, such as wireless sensor network key pre-distribution, small-world complex network modeling, and Ramsey theorems, necessitates graph designs with tailored characteristics and a combination of desirable properties. These properties include minimum vulnerability, minimum diameter, regularity, Hamiltonian connectivity, and minimal complete subgraph representation. Incorporating multiple such features into a single graph presents a challenge due to conflicting properties and the exponential difficulty of achieving a desired combination. To address this challenge and provide more realistic representations, we propose an algorithm for generating a novel class of graphs called Tight graphs. We conduct a thorough investigation into the properties of these graphs and demonstrate their potential applicability in diverse domains, including wireless sensor network key pre-distribution, small-world complex network models, and improved lower bounds for Ramsey numbers. Our algorithm enables the design of graphs that strike a balance between desirable properties, catering to the specific requirements of various applications.

Keywords: Network Vulnerability, Wireless Sensor Networks, Small-world Complex Networks, Ramsey Numbers, Tight Graphs

1. Introduction

Graphs have become invaluable tools for modeling communication systems between entities, finding applications in various domains such as social networks, map coloring, radio frequency assignment, VLSI, communication networks, bioinformatics, and data flow analysis [1-6]. The diverse nature of these applications has led to the development of several named graph types, including Harary [7], Erdős–Rényi [8], Barabási–Albert [9], Tutte [10], Petersen [11], and others. Harary graphs were introduced to minimize vulnerability (connectivity) [7].

In the pursuit of achieving improved maximum-connected graphs, Jafarpour et al. proposed a new graph type [12]. However, constructing graphs with diverse combinatorial properties poses a significant challenge due to inherent conflicts among these properties. For instance, the number of edges in a graph plays a crucial role in reducing its vulnerability. While graphs like Harary graphs aim to achieve maximum connectivity using the fewest possible edges, they often exhibit an abundance of complete subgraphs, making them impractical for certain real-world applications. Consequently, as we strive to enhance these properties, the complexity of generating such desirable graphs increases exponentially. In this study, we further enhance this graph type by incorporating the following properties:

- Minimum diameter: We minimize the graph's diameter, which is closely associated with transmission speed in networks. This property proves crucial in addressing various graph-related problems and applications.
- Maximum regularity: We design the graphs to be as regular as possible, promoting load balancing on network vertices, which is highly desirable in numerous applications.
- Reduced vulnerability (on average): Vulnerability stands as a key concern in network design. Our graphs aim to achieve lower vulnerability on average, thereby enhancing network robustness.
- Minimum complete subgraphs: By distributing edges more evenly, we prevent the occurrence of excessive complete subgraphs, reducing graph vulnerability.

Graphs possessing the aforementioned properties provide realistic models for a range of problems, including small-world complex network models, key pre-distribution in wireless sensor networks (WSNs), and finding Ramsey numbers. Any combination of these properties can be utilized in other specific applications that require them.

The remainder of this paper is organized as follows: Section 2 discusses specific classes of graphs, referred to as named graphs, that exhibit low vulnerability, regularity, or low diameter. Subsequently, Section 3 presents the concept of constructing a new graph type called Tight graphs, along with their characteristics. These graphs effectively demonstrate their efficiency in providing small-world complex network models and constructing low-vulnerability networks. Moreover, Tight graphs find applications as base graphs in establishing lower bounds for Ramsey numbers. Section 4 explores several applications of Tight graphs. Section 5 will present an analysis of the time and memory complexity of the algorithms introduced in this research, and finally, Section 6 concludes the discussion.

2. Analysis of Specific Graph Classes: Regularity, Diameter, and Vulnerability

This section focuses on exploring notable graph classes known as named graphs, including circulant graphs, Harary graphs, and generalized Petersen graphs. These graph classes are examined in terms of their regularity, diameter, and vulnerability, shedding light on their unique properties and potential applications.

A circulant graph, denoted by $C(n, L \subseteq \pm 1, \pm 2, ..., \pm j)$, where $1 \le j \le \lfloor n/2 \rfloor$ and $n \ge 3$, is a graph with a vertex set $V = \{0,1,...,n-1\}$ and an edge set $E = \{(i, j) : |i-j| \overset{n}{\equiv} s\}$ and $s \in L \subseteq \{\pm 1, \pm 2, ..., \pm j\}$ [13,14]. For example, $C(n, \pm 1)$ represents a cycle graph, while $C\left(n, \pm 1, \pm 2, ..., \pm \lfloor n/2 \rfloor\right)$ corresponds to the complete graph K_n. Circulant graphs exhibit a regular structure in which each vertex has an equal number of neighbors. However, the diameter of circulant graphs is proportional to the number of vertices in the graph [15]. Figure 1 illustrates three different examples of circulant graphs.

Harary graphs are derived from circulant graphs and are renowned for achieving expected connectivity with the minimum number of edges. A Harary graph, denoted as $H(n,k)$, represents a specific instance of a k-connected graph with n vertices and the minimum possible number of edges. The Harary graph $H(n,k)$ is characterized by $\lceil kn/2 \rceil$ edges [16-19]. Since the diameter of circulant graphs is linear, it follows that Harary graphs also have a linear diameter [15,20].

Generalized Petersen graphs (GPGs) are constructed by combining two circulant graphs with an equal number of vertices, where edges exist between vertices with matching indices in the two edge graphs [11,21]. The outer graph forms a cycle, while the inner graph represents a circulant graph with a list L of size k. Consequently, generalized Petersen graphs can be defined as $GPG(n,k)$, with the vertex set $V\left(GPG(n,k)\right) = \{u_0, u_1, ..., u_{n-1}, v_0, v_1, ..., v_{n-1}\}$ and the edge set $E\left(GPG(n,k)\right) = \{u_i u_{i+1}, u_i v_i, v_i v_{i+k} \mid 0 \le i \le n-1\}$. Figure 2 depicts two examples of generalized Petersen graphs with different internal circulant graphs. When constructing multilevel GPG structures, such as $GPG(n,1,k_2,...,k_l)$, where l denotes the number of nested levels in the graph, they exhibit a structure reminiscent of spider webs.

Circulant and generalized Petersen graphs are commonly employed graph types for establishing lower bounds on Ramsey numbers [22-25]. On the other hand, Harary graphs are well-known for their maximum connectivity [7,26] and find applications in the construction of small-world complex networks [27-31].

3. Tight Graphs

In this section, we present a class of graphs that exhibit regularity, lower vulnerability, and smaller diameter compared to the graphs discussed in Section 2. Moreover, these graphs offer improved utility in determining more accurate lower bounds for Ramsey numbers. To construct the proposed graphs, we consider two individuals facing each other and iteratively add edges to an initially empty graph with n vertices. In each step, the first person has the sole authority to choose the source vertex, while the second person connects this vertex to the farthest vertex in the graph. The farthest vertex from any given vertex u is defined as the vertex with the longest shortest path to u. For disconnected vertices, the path length is assumed to be infinite.

Consequently, the resulting graph structure solely depends on the decisions made by the first person. Figure 3 illustrates the Star and Crown graphs, both comprising seven vertices, which can be generated using this algorithm. In both cases, the first person consistently designates the vertex with the highest degree as the source vertex for the second person.

These graphs, created through this methodology, exhibit a propensity to resist the formation of complete subgraphs. For example, in Figure 4a, the first person selects the vertices b, b, b, and c consecutively, and the second person adds the edges (b,a), (b,c), (b,d), and (c,d) to the graph. However, in Figure 4b, due to the substantial distance between vertex a and vertices d, e, and f, one of the edges (d,e), (d,f), or (e,f) should not be present in the graph. In other words, the formation of a complete subgraph in such graphs is not possible when the graph's diameter is large.

In the represented algorithm, the second person is unable to determine the destination vertex and instead connects the source vertex chosen by the first person to the farthest vertex in the graph. By granting decision-making power to the second person, we can introduce various types of graphs.

In addition, this paper considers regularity and low vulnerability as additional properties for the target graph. To achieve regularity, it is sufficient for the first and second person to connect two vertices with the lowest degree. Regarding low vulnerability, it is desirable for the paths between two vertices to be as long as possible. In other words, when choosing between several vertices with the same degree and located at the same distance from the source vertex, the vertex with the fewest number of shortest paths from the source vertex is selected as the destination vertex. The priority for selecting destination vertices is determined based on the lowest degree, followed by the longest distance from the source vertex, and finally, the fewest number of shortest paths from the source vertex.

Furthermore, the algorithm aims to minimize connections within a community by preferring vertices with a larger number of paths between them, as it is anticipated that vertices with more paths between them are more likely to belong to the same community. Conversely, the algorithm strives to create diverse paths from each vertex to other vertices, ensuring that even if one or more different paths are lost between two vertices, there remains a feasible connection between them. This characteristic indicates that the generated graphs are likely to exhibit a higher Tightness parameter compared to graphs with an equal number of vertices and edges [32]. Therefore, these graphs are referred to as Tight graphs. The procedure for generating Tight graphs is outlined in Algorithm 1, and three examples of Tight graphs are illustrated in Figure 5. This paper presents an algorithm based on the aforementioned approach and investigates specific properties of the resulting graphs. In this section, we will delve into the structure of Tight graphs and explore their inherent characteristics.

3.1. *Some Properties of Tight graphs*

By constructing Tight graphs (based on Algorithm 1), some properties can be synthesized. These graphs can be considered \$m\$-regular under certain circumstances. The only case where the resulting graphs will not be regular is when both the number of vertices and the desired degree of the vertices are odd. This will cause one of the vertices to have a higher degree than the others because the sum of the degrees of the vertices must be even. Therefore, we can give the Property 3.1 for Tight graphs.

Property 3.1: If expected degree (m) is even or the number of vertices (n) is even, the resulting graph will be m-regular.

One of most subjects of interest in named graphs is related to the Hamiltonian circuits. During the construction of hard graphs, the process begins by selecting vertices with minimum degree as the source vertex. The farthest vertex from this source can be any other vertex in the graph. In the early stages, the vertex with the lowest degree is chosen as the destination vertex, resulting in the creation of complete or path subgraphs with two vertices ($K_2 = P_2$). As the process progresses, these subgraphs evolve into larger path subgraphs. Eventually, once all the paths are interconnected, they form a path graph (P_n). In subsequent steps, the vertices at the beginning and end of the path graph are selected as the source and destination vertices, following the conditions outlined by the algorithm. This process leads to the formation of a cycle graph (C_n) in the resulting after adding n edges. Thus, Tight graphs do have Hamiltonian circuits for which property 3.2 and lemma 3.3 are about.

Property 3.2: The first n iterations of Algorithm 1 creates a Hamiltonian circuit where n is the number of vertices.

Lemma 3.3: The graph is Hamiltonian if Algorithm 1 does not reach the Final Step (line 11) of the Algorithm 1.

Proof: Based on Property 3.2, the first n iterations of Algorithm 1 creates a Hamiltonian circuit. The only step in the algorithm that might destroy the created Hamiltonian circuit is the Final Step (line 11) of the Algorithm 1.

Theorem 3.4: According to Final Step of Algorithm 1, there will always be vertices *p* and *q* such that $(v, p) \notin E(G)$, $(u, q) \notin E(G)$, and $(p, q) \in E(G)$.

Proof: Proof by contradiction. Suppose that for all edges $(p, q) \in E(G), (v, p), (u, p) \in E(G)$ or $(v, q), (u, q) \in E(G)$. Therefore, for each edge in the graph, except for the edges on which the vertices of *u* and *v* are, the degree of *u* and *v* must be added at least by one. We know that $\deg(u) = \deg(v) = m - 1$ and $\forall a \in V(G), a \neq u, a \neq v : \deg(a) = m$. Therefore, the number of edges that are not on the vertices of *u* and *v* are equal to $\big((mn/2) - 1\big) - \big(2(m-2) + 1\big)$, and for each of these edges, the degree of *u* and *v* must added at least by one. Thus,

Algorithm 1: Tight Graph Construction Algorithm

Input: Graph order: $n = |V(G)|$
Input: Expected degree: m
Output: Tight graph with n vertices and connectivity m

if ($n \le 0$ or $m < 0$) **Then**
  return (**null**)
end if
$V(G) \leftarrow 1...n$
$E(G) \leftarrow \phi$
if ($n = 1$ or $m = 0$) **Then**
  return (G)
end if
while (The minimum degree of vertices < m) **do**
  if (The number of vertices with degree less than m is 2) **then**
    Final Step:
    Find u and v with minimum degree
    if ($(u,v) \in E(G)$) **then**
      Find p and q such that:
        $(v,p) \notin E(G)$, $(u,q) \notin E(G)$, and $(p,q) \in E(G)$
      $E(G) \leftarrow E(G) \cup \{(u,q),(v,p)\}$
      $E(G) \leftarrow E(G) - (p,q)$
    else
      $E(G) \leftarrow E(G) \cup (u,v)$
    end if
    return (G)
  end if
  $u \leftarrow$ vertex with the minimum degree
  $v \leftarrow$ first vertex in $V(G)$ ordered by the following conditions:
    1. $v \ne u$,
    2. Minimum degree,
    3. Maximum distance from u ,
    4. Less minimum paths from u .
  $E(G) \leftarrow E(G) \cup (u,v)$
end while
return (G)

$$\deg(u) = \deg(v) = m - 1 \geq \frac{mn}{2} - 1 - 2(m-2) - 1 \Rightarrow 0 \geq \frac{mn}{2} - 3m + 3,$$

that is a contradiction, because there is no such $n > m > 0$ that applies to this inequality assuming $\deg(u) = \deg(v) = m - 1$.

Therefore, three main cases remain to check the Hamiltonian circuit in the graphs resulting from Algorithm 1. All these cases arise when the edges (p,q) are on the initial Hamiltonian circuit. These cases are:

- $(u,p),(u,q),(v,p),(v,q) \notin E(G)$
- $(u,q),(v,p),(v,q) \notin E(G)$ and $(u,p) \in E(G)$
- $(u,q),(v,p) \notin E(G)$ and $(u,p),(v,q) \in E(G)$

Figure 6 depicts these cases.

Theorem 3.5: If, in the Final Step of the Algorithm 1, the selected edge (p,q) is part of the initial Hamiltonian circuit, and edges (u,p) and (v,q) are in E(G), then the resulting graph will be Hamiltonian.

Proof: Based on Property 3.2, the circuit is formed in the first n iterations of the algorithm implementation, and this circuit will remain until the end. Consequently, the resulting graph will be Hamiltonian. Alternatively, in Final Step of Algorithm 1, one edge will be removed, and alternative edges will be added to the graph. In this scenario, if the initial Hamiltonian circuit is $qa_1a_2...a_kuvb_1b_2..b_s pq$, then the alternative circuit will be $qa_1a_2...a_kupb_s..b_2b_1vq$, which is still a Hamiltonian graph. Figure 7 illustrates this Hamiltonian circuit.

With simulations that have been done with up to 500 vertices and different vertex degree greater than 1 (m>1), the resulting graph was Hamiltonian because in case of destruction of the initial Hamiltonian circuit, there was always some alternative Hamiltonian circuits. According to the properties, corollaries, and theorems discussed above, we come to a conjecture which may need further investigation. It seems that the Tight graphs are always Hamiltonian.

Conjecture 3.6: The graph resulting from the Tight graph construction algorithm (Algorithm 1) is always Hamiltonian.

3.2. The logarithmic diameter of Tight graphs

Due to the complexity of the algorithm used to construct Tight graphs, determining their diameter through mathematical equations would be highly intricate. As per the algorithm, it is expected that after forming the cycle, the process of adding edges to the graph will occur in a crumpling fashion, resembling the order depicted in Figure 8. Although the reliability of the order shown in Figure 8 is uncertain, it is anticipated that the graph diameter will decrease at an approximate logarithmic rate. In contrast, Harary graphs, including their more general form, circulant graphs, exhibit a linear diameter [15,20].

Consequently, Tight graphs can be compared to circulant graphs in terms of their diameter. As illustrated in Figures 9 and 10, the rate of change in the graph diameter for a fixed m in Tight graphs is logarithmic, increasing by ln(n) (where n represents the number of graph vertices). In contrast, the diameter of circulant graphs grows linearly. Figure 11 displays the diameter of circulant and Tight graphs for different vertex counts (N) ranging from 10 to 120 and varying degrees (M). Observations and simulations indicate that the diameter of Tight graphs appears to be bounded by a logarithmic limit.

Based on simulations, the diameter of the Tight graph with m=3 remains limited to 2ln(n)-1 for up to a thousand vertices, and for larger m values, the graph diameter is even smaller. However, this result is derived from simulations and lacks a mathematical proof, thus remaining an open problem. Figures 12 and 13 provide examples of Tight and circulant graphs, respectively. The intricate interconnections in Tight graphs contribute to their logarithmic diameter, while the regular arrangement of edges in circulant graphs leads to linear diameter growth.

Conjecture 3.7: For m>2, the upper bound of the diameter of Tight graphs is 2ln(n)-1.

4. Applications of Tight graphs

Tight graphs possess several properties, including logarithmic diameter, low vulnerability, regularity, and a tendency to avoid forming complete subgraphs whenever possible. These characteristics give rise to various applications for this type of graph. In this section, we will explore some of these applications.

4.1. Key Pre-distribution in Wireless Sensor Networks (WSNs)

Wireless sensor networks find applications in diverse fields such as environmental monitoring, transportation, and meteorology [33,34]. Energy and memory consumption are crucial considerations in these networks due to the limited size, low memory, and battery capacity of the sensors (or agents) involved. Therefore, optimizing communication between these agents is essential to minimize energy consumption. Additionally, in certain applications, ensuring the security of transmitted information is of utmost importance. This necessitates the pre-distribution of encryption and decryption keys among the agents in the wireless sensor network [35]. A high number of keys would increase memory usage and information processing, ultimately leading to elevated energy consumption within the network. Another challenge in these networks arises from the spatial arrangement of the agents, which adds complexity to network-related problems. Each agent's antenna has a limited range, restricting its communication to other agents within this range. The resulting graph that represents these connections is known as a location graph, which dynamically changes with agent movements and relocations. Figure 14 presents an example of agent locations (denoted as a to f) and their location graph.

Within these networks, once the keys are stored in the memory of each agent, agents possessing a common key can exchange information with one another. As a result, a graph can be constructed where each edge represents the sharing of keys between two agents. This graph, generated from the key distribution process, is referred to as a key graph. The combination of the location graph and the key graph provides insight into the communication capabilities between agents within the wireless sensor network [36]. Specifically, two agents can communicate if they share a common edge in both the location graph and the key graph. The conjunction of edges from these two graphs forms the connection graph. Figure 15 presents an example of location and key graphs, along with the communication graph for this network.

The location graph is determined solely based on the positions of sensors or agents and cannot be controlled. However, the key graph, through appropriate pre-distribution, can be connected to the communication graph as extensively as possible. Moreover, the diameter of the communication graph plays a crucial role in packet transmission within the network. A large diameter results in repeated packet transmissions between agents, leading to high energy consumption across the network. Hence, designing a key graph with a small diameter, low vulnerability, and regularity is essential to ensure maximum connectivity of the communication graph, minimize energy consumption, and balance memory requirements in each agent [37,38].

Tight graphs possess all the desired properties mentioned above, making them highly efficient as fundamental graphs for key pre-distribution. To pre-distribute keys using Tight graphs, a Tight graph with the desired number of agents is constructed. Then, for the edges of each clique in this graph, a key can be assigned and placed in the memory of the corresponding agent vertices.

4.2. Ramsey Numbers

The Ramsey theorem is a significant result in the field of combinatorics. In the context of graph theory, it states that as the number of vertices in a graph increases, if we color the edges of the complete graph with these vertices using red and blue, there will always exist groups of vertices where all the edges between them are of the same color, either red or blue [39]. The Ramsey numbers $R(r,s) \in \mathbb{N}$ represent the minimum value of n_0 such that for any coloring of a complete graph with $n \geq n_0$ vertices using two colors (red and blue) in an arbitrary manner, there will be a set of r vertices where all the edges within that set are red, or a set of s vertices where all the edges within that set are blue [40-43]. Although the definition of Ramsey numbers is straightforward, determining these numbers is a highly challenging task. Thus far, apart from the numbers $R(r,1)$ and $R(r,2)$, which can be obtained using equations 1 to 3, only 9 Ramsey numbers have been discovered, highlighting the difficulty of this problem [41,44-53]

$$R(r,s) = R(s,r) \tag{1}$$

$$R(r,1) = R(1,r) = 1 \tag{2}$$

$$R(r,2) = R(2,r) = r \tag{3}$$

Equations 4 to 9 represent attempts to determine upper and lower bounds for Ramsey numbers. Obtaining equations to calculate smaller bounds can be a challenging task. Is there an alternative approach to determine the values of Ramsey numbers without relying on complex mathematical equations? One possible solution is to employ algorithms instead of attempting to narrow existing bounds. By using algorithms, we can explore computational methods to find the exact values of Ramsey numbers, bypassing the need for intricate mathematical equations.

Algorithm 2: Modified Tight Graph Construction Algorithm

Input: Graph order: $n = |V(G)|$
Input: Clique size limit: **r**
Output: Pseudo-Tight graph with Max-Clique smaller than **r**

if ($n \leq 0$ or $m < 0$) **Then**
  return (**null**)
end if
$V(G) \leftarrow 1...n$
$E(G) \leftarrow \phi$
if ($n = 1$ or $r = 1$) **Then**
  return (G)
end if
$\forall 0 < i, j \leq n : distance(i, j) \leftarrow \infty$
while ($\exists i, j : distance(i, j) > 1$) **do**
  $u \leftarrow$ vertex with the minimum degree
  $v \leftarrow$ first vertex in $V(G)$ ordered by the following conditions:
    1. $v \neq u$,
    2. Minimum degree,
    3. Maximum distance from u,
    4. Less minimum paths from u,
    5. Adding edge (u, v) does not create a clique with size **r.**
  $E(G) \leftarrow E(G) \cup (u, v)$
  Update distance matrix
  $distance(u, v) \leftarrow 1$
  $distance(v, u) \leftarrow 1$
end while
return (G)

$$R(r,s) \le R(r-1,s) + R(r,s-1) \tag{4}$$

$$R(s,s) \le 4R(s,2s) + 2 \tag{5}$$

$$R(s,s) \ge \frac{s\,2^{s/2}}{e\sqrt{2}} \tag{6}$$

$$c_1 \frac{s^2}{\ln s} \le R(3,s) \le c_2 \frac{s^2}{\ln s} \tag{7}$$

$$R(r,s) < \binom{r+s-2}{s-1} \tag{8}$$

$$\left[1+o(1)\right]\frac{s}{e\sqrt{2}}2^{s/2} \le R(s,s) \le \left[1+o(1)\right]\frac{4^{s-1}}{\sqrt{\pi s}} \tag{9}$$

The algorithm devised in this study is built upon the fundamental definition of Ramsey numbers. According to the definition of Ramsey numbers $R(r,s) \in \mathbb{N}$, our goal is to determine the value of n_0 such that if we color the edges of any complete graph with $n \ge n_0$ using red and blue, there will always exist either a complete subgraph with r vertices colored in red or a complete subgraph with s vertices colored in blue. Hence, if we can identify a complete graph with *k* vertices that does not contain either a complete subgraph with r red vertices or a subgraph with s blue vertices, we can conclude that $R(r,s) > k$.

The proposed algorithm in this study aims to discover this value of *k* in order to establish a lower bound for the Ramsey numbers. For this purpose, the algorithm for generating Tight graphs has been modified (Algorithm 2) to add as many edges as possible to the initial graph based on the value of r, until the addition of each new edge results in a clique of size r. The remaining edges in the resulting graph are considered as blue edges. Consequently, the desired counterexample graph should not contain any cliques of size r or independent sets of size s.

In Algorithm 3, Algorithm 2 initially generates a pseudo-Tight graph and subsequently employs a random search strategy to identify a potential counterexample for a Ramsey number. This random approach involves adding or removing edges from the graph. The selection of edges to add or remove is performed in a straightforward manner, constituting the simplest approach implemented in Algorithm 3.

To assess the effectiveness of this algorithm, we sought counterexamples for various Ramsey numbers. Specifically, we considered the Ramsey numbers listed in Table 1 as our target examples. The resulting counterexample graphs, along with their complements, obtained through the Algorithm 3, are depicted in Figure 16. From Figure 16, we can observe that graphs 16a and 16b exhibit a circulant structure, whereas graphs 16c and 16d do not. This indicates that Tight graphs, in conjunction with the proposed algorithm, yield more favorable outcomes compared to circulant graphs in these particular cases. As the values of r and s increase, it is expected that circulant graphs will become less efficient in establishing lower bounds for Ramsey numbers.

Algorithm 3: Ramsey Counterexample Search Algorithm

Input: Graph order: $n = |V(G)|$
Input: Ramsey parameters: **r** and **s**
Input: Iterations: **I**
Output: A counterexample graph with *n* vertices for R(**r**, **s**)

```
G ← Pseudo-Tight Graph (n, r)
for i ← 1 to I do
    if (there is a clique C with size r in G) then
        Randomly select one edge of C and remove it from G
    else
        if (there is an independent set X with size s in G) then
            Randomly select one edge of X and add it to the G
        else
            return (G)
        end if
    end if
end for
return (null)
```

Table 1: Some known Ramsey numbers

r	s	R(r, s)	Counterexample size
3	3	6	5
4	3	9	8
5	3	14	13
6	3	18	17

4.3. Small-world complex networks

In the realm of real-world networks, there exist large-scale networks that exhibit distinct properties and behaviors, setting them apart from typical random graphs. These networks are commonly referred to as complex networks [54]. Two broad categories of complex networks are small-world networks and scale-free networks [9,30,31,55-57]. In small-world networks, the network diameter demonstrates a logarithmic relationship with the number of network vertices, while scale-free networks exhibit a power-law distribution independent of the network's scale. To investigate the characteristics and parameters of such networks, models are often proposed. These models are subsequently evaluated to glean insights into the behavior and parameters of similar networks [58].

Notably, the Price and Barabasi-Albert models [9], developed by Derek Sola Price, Albert Laszlo Barabasi, and Reka Albert, serve to capture the features of scale-free networks, while the Watts-Strogatz model, introduced by Duncan J. Watts and Steven Strogatz, is designed to simulate small-world networks. Algorithm 4 outlines the procedure for constructing complex networks using the Watts-Strogatz model.

Algorithm 4: Watts-Strogatz Complex Network Construction Algorithm

Input: Graph order: $n = |V(G)|$

Input: Connectivity: m

Input: Rewire probability: β

Output: Small-world complex network based on Watts-Strogatz model

1: Construct a regular ring lattice, a graph $\boldsymbol{G}$ with $\boldsymbol{n}$ nodes each connected to $\boldsymbol{m}$ neighbors $\boldsymbol{m}/2$ of each side

2: Rewire each edge $(u,v) \in E(G)$ with (u,t) and probability β where t is chosen uniformly at random from all possible nodes while avoiding self-loops $u \neq t$ and link duplication $(u,t) \notin E(G)$

3: **return** (**G**)

In the initial step of Algorithm 4, Watts and Strogatz aim to construct a highly connected graph, followed by randomly rewiring some of its edges. The graph or network generated in the first step of this algorithm is known as a Harary graph. One of the reasons for utilizing this type of graph in the model is its inherent high connectivity, which helps preserve the overall structure of the network during the edge rewiring process. Watts and Strogatz expected this algorithm to produce a small-world network. However, our investigations in Sections 2 and 3, along with the findings presented in Figures 10 and 11a, demonstrate that the diameter of Harary graphs, and even circulant graphs, does not exhibit a logarithmic relationship with the number of vertices. Consequently, in order to achieve a logarithmic diameter ratio for the networks created using this model, we need to rely on the edge rewiring step. Considering the conjecture presented in Conjecture 3.7 and the observations from Figures 9 and 11b, it is expected that the Tight graphs possess a logarithmic diameter. Therefore, it is possible to replace the Watts-Strogatz complex network model with a novel model that employs Tight graphs instead of Harary graphs in the initial step of the algorithm. Algorithm 5 outlines the process for constructing a complex network using Tight graphs. As discussed in [12], graphs such as Tight graphs exhibit greater robustness. Additionally, through establishing various connections between pairs of vertices in the Tight graph construction algorithm, we anticipate the presence of alternative shortcuts even if a short path is lost. Consequently, by rewiring the edges of these graphs, the likelihood of compromising the overall network structure becomes significantly reduced, ensuring that the diameter of the graph remains logarithmic to the greatest extent possible [30,31]. Replacing Harary graphs with Tight graphs in the construction of a small-world complex network model enables the network's diameter to scale logarithmically with the number of vertices. This ensures that the network created using Algorithm 5 exhibits the defining characteristics of a small-world complex network such as logarithmic diameter and connectivity.

Algorithm 5: Complex Network Construction Algorithm Using Tight Graph

Input: Graph order: $n = |V(G)|$
Input: Connectivity: m
Input: Rewire probability: β
Output: Small-world complex network using Tight graph

1: Construct a Tight graph $\boldsymbol{G}$ with $\boldsymbol{n}$ nodes each having degree $\boldsymbol{m}$

2: Rewire each edge $(u,v) \in E(G)$ with (u,t) and probability β where t is chosen uniformly at random from all possible nodes while avoiding self-loops $u \neq t$ and link duplication $(u,t) \notin E(G)$

3: **return** (**G**)

While many methods for representing complex network models rely on basic graphs, another approach involves the concept of link prediction [59,60]. Link prediction focuses on estimating the likelihood of the formation or removal of links (edges) within complex networks over time. This approach serves as both a predictive tool and a framework for constructing and evolving complex network models. In real-world networks, the uncertainty in the existence of links over time results in a lack of fixed structures. For instance, in social networks, links may be formed or removed at any moment, leading to dynamic changes in the network's structure. This rapid evolution introduces uncertainty, which is reflected in the random rewiring of links in Algorithms 4 and 5. With this perspective, in addition to the traditional Watts-Strogatz model, a probabilistic model can be devised using Tight graphs as a foundation. In Tight graphs, the focus is on connecting the source vertex to the farthest possible vertex, aiming to minimize the network's diameter and achieve logarithmic scaling with the number of vertices. From a link prediction standpoint, the probability of establishing a link between two vertices can be modeled as a function of their distance from the source vertex, with closer vertices having a lower likelihood of connection.

5. Time and Memory Complexity Analysis

In this section, we analyze the memory and time complexity of the algorithms presented in this research, focusing on the worst-case scenarios. For Algorithm 1, which constructs Tight graphs, the operations in lines 1 to 8 execute in $O(1)$ time. The while loops from lines 9 to 30 iterates as many times as the number of edges to be added to the graph, which equals $nm/2$. Evaluating the degrees of the graph's vertices takes $O(n)$ time in the worst case, although this can be optimized to lower complexity using data structures like linked lists or heap trees. However, for this analysis, the worst-case scenario of $O(n)$ is considered.

The if condition in line 10 can also be evaluated in $O(n)$ time, and its instructions are executed at most once in the final step of graph construction. The operation in line 12 similarly takes

$O(n)$ time in the worst case. Checking the condition in line 13 requires $O(1)$, while executing line 14 involves evaluating all pairs of vertices, resulting in a complexity of $O(n^2)$. Lines 16, 17, and 19 are constant-time operations, each with a complexity of $O(1)$. Executing line 21 requires $O(n)$ time in the worst case. For the conditions in lines 24 to 28, determining the destination vertex takes $O(n+\varepsilon)=O(n+nm/2)$ where ε represents the number of edges, as condition 26 evaluates in $O(n)$, and conditions 27 and 28 are assessed using the BFS algorithm. Finally, the operations in lines 21, 29, and 31 also execute in $O(1)$ time. Thus, the overall time complexity of Algorithm 1 for constructing Tight graphs can be expressed as equation 10 where c_i are constants.

$$T_{A\text{lg}orithm_1}(n,m)=c_1nm(5n+nm)+c_2n^2+c_3n+c_4\in O(n^2m^2) \tag{10}$$

As observed, the bottleneck of this algorithm lies in identifying the vertex with the greatest distance and the minimum number of shortest distances from the source vertex, a process performed using the BFS algorithm. Optimizing this step could significantly reduce the overall time complexity of the algorithm.

To implement this algorithm, a graph adjacency matrix and arrays to store data such as the degrees of vertices are required. These structures collectively consume $O(n^2)$ space. Consequently, the memory complexity of Algorithm 1 can be expressed as equation 11.

$$M_{A\text{lg}orithm_1}(n,m)\in O(n^2) \tag{11}$$

Algorithm 2 has a structure similar to Algorithm 1, with a key difference: at each step of adding an edge to the graph, it first ensures that a clique of size r is not formed. Additionally, after adding an edge, the distance matrix is updated to improve the efficiency of finding the farthest vertex. The time required to check for a clique of size r is directly proportional to r and becomes exponential in the general case. Thus, Algorithm 3 is expected to exhibit exponential time complexity in the worst case. However, for small values of r and s, examining cliques and independent sets by considering all possible combinations of vertices can be performed with time complexity $O(n^r)$ and $O(n^s)$, respectively. Therefore, the overall time complexity of Algorithm 3 in the worst case can be expressed as equation 12.

$$T_{A\text{lg}orithm_3}(r,s,I)\in O(I\times n^{r+s}) \tag{12}$$

The memory required for Algorithm 3 is also $O(n^2)$, as it involves storing the graph's adjacency matrix, similar to Algorithm 1.

The time and memory complexity of Algorithm 5 are equivalent to those of Algorithm 1. While Algorithm 5 incorporates a rewiring step, the number of rewiring operations must remain smaller than the total number of edges in the original graph. Exceeding this limit could significantly alter the structure of the final network, diverging it from the initial network's characteristics.

6. Conclusion

In this study and for future research, we have introduced a novel graph type known as Tight graphs. As the name suggests, these graphs are expected to enhance vulnerability parameters such as Connectivity, Toughness, and Tightness. We compared Tight graphs with circulant graphs and found that our proposed graphs outperform Harary graphs and circulant graphs in general, particularly in terms of minimizing the graph diameter. This advantage arises from the avoidance of complete subgraph construction in Tight graphs. Consequently, Tight graphs offer potential for small-world models in complex network analysis.

Furthermore, we explored other graph properties of Tight graphs, including regularity and Hamiltonicity. The regularity of Tight graphs, combined with their small diameter and low vulnerability, makes them valuable in applications such as wireless sensor networks. Additionally, similar to circulant graphs, Tight graphs can serve as options for establishing lower bounds for Ramsey numbers. The construction algorithm of Tight graphs aims to prevent the formation of complete subgraphs by connecting each vertex to the farthest vertex, thereby extending this property to the complementary graph as well. Additionally, structures in nature that appear unordered at first glance, such as silkworm cocoons, might possess very low vulnerability and have construction algorithms similar to Tight graphs. Exploring the algorithms employed by nature in constructing networks with low vulnerability could yield valuable insights.

This study aims to evaluate the Hamiltonicity of Tight graphs and explore its special cases. Additionally, the logarithmic diameter of this class of graphs was analyzed through simulations. These findings are presented as conjectures, whose validity and formal proof require further mathematical investigation, offering potential directions for future research.

References

[1]. Bondy J.A, Murty U.S.R. "Graph theory with applications". *Macmillan London*. (1976). ISBN: 978-0444194510.

[2]. Calamoneri T., Massini, A., Vrt'o I. "Antibandwidth of complete k-ary trees". *Electronic Notes in Discrete Mathematics*. 24, pp 259-66, (2006). DOI: 10.1016/j.disc.2008.10.019.

[3]. Chen J., Saad Y. "Dense subgraph extraction with application to community detection". *IEEE Transactions on knowledge and data engineering*. 24, pp 1216-30, (2010). DOI: 10.1109/TKDE.2010.271.

[4]. Gross J.L., Yellen J., Anderson M. "Graph theory and its applications". *Chapman and Hall/CRC.* (2018). ISBN: 978-1584885054.

[5]. Nishizeki T., Rahman M.S. "Planar graph drawing". *World Scientific*. (2004). ISBN: 978-9812560339.

[6]. Sung W.K. "Algorithms in bioinformatics: A practical introduction". *Chapman and Hall/CRC*. (2009). ISBN: 978-1420070330.

[7]. Harary F. "The maximum connectivity of a graph". *Proceedings of the National Academy of Sciences of the United States of America*. 48, pp 1142-46, (1962). URL: www.jstor.org/stable/71730.

[8]. Gao Y. "Large deviations for mean field model in Erdős–Rényi graph". *Statistics and Probability Letters*. (2024), 205, 109953. DOI: 10.1016/j.spl.2023.109953.

[9]. Barabási A.L., Bonabeau E. “Scale-free networks”. *Scientific American*. 288, pp 60-9, (2003). DOI: 10.1038/scientificamerican0503-60.

[10]. Tutte W.T. “On Hamiltonian Circuits”. *Journal of the London Mathematical Society*. 21, pp 98-101, (1946). DOI: 10.1112/jlms/s1-21.2.98.

[11]. Holton D.A., Sheehan J. “The Petersen Graph”. *Cambridge University Press*. (1993). DOI: 10.1017/CBO9780511662058.

[12]. Jafarpour M., Shekaramiz M, Javan A, Moeini A. “Building Graphs with Maximum Connectivity”. *Intermountain Engineering, Technology and Computing (IETC)*. (2020). DOI: 10.1109/IETC47856.2020.9249130.

[13]. Balakrishnan R., Sethuraman G., Wilson R.J. “Graph theory and its applications”. *Alpha Science Int'l Ltd*. (2004). ISBN: 978-8185015552.

[14]. Basak M., Saha L., Das G.K., Tiwary K. “Fault-tolerant metric dimension of circulant graphs Cn (1, 2, 3)”. *Theoretical Computer Science*. 817, pp 66-79, (2020). DOI: 10.1016/j.tcs.2009.01.041.

[15]. Meijer P.T. “Connectivities and diameters of circulant graphs”. *Simon Fraser University*. (1987). URL: api.semanticscholar.org/CorpusID:28325039.

[16]. Baldoni R., Bonomi S., Querzoni L., Piergiovanni S.T. “Investigating the existence and the regularity of logarithmic harary graphs”. *Theoretical computer science*. 410, pp 21-23, (2009). DOI: 10.1016/j.tcs.2009.01.041.

[17]. Dillencourt M.B. “Polyhedra of small order and their Hamiltonian properties”. *Journal of combinatorial theory, Series B*. 66, pp 87-122, (1996). DOI: 10.1006/jctb.1996.0008.

[18]. Harary F. “Graph Theory”. *Massachusetts: Addison-Wesley Publication Company. Inc.* (1972). ISBN: 978-0201027877.

[19]. Liu Q., Huang X., Zhang Z. “Optimally restricted edge connected elementary Harary graphs”. *Theoretical Computer Science*. 497, pp 131-8, (2013). DOI: 10.1016/j.tcs.2011.12.015.

[20]. Bhabak P., Harutyunyan H.A, Tanna S. “Broadcasting in Harary-Like Graphs”. *IEEE 17th International Conference on Computational Science and Engineering*. 14887381, (2014). DOI: 10.1109/CSE.2014.244.

[21]. Coxeter H.S. “Self-dual configurations and regular graphs”. *Bulletin of the American Mathematical Society*. 56, pp 413-55, (1950). DOI: 10.2307/2033189.

[22]. Dudek D., Frankl P., Rödl V. “Some recent results on Ramsey-type numbers”. *Discrete Applied Mathematics*. 161, pp 1197-1202, (2013). DOI: 10.1016/j.dam.2010.09.005.

[23]. Goedgebeur J., Van Overberghe S. “New bounds for Ramsey numbers R(K_{k-e}, K_{l-e})". *Discrete Applied Mathematics*. 307, 212-221, (2022). DOI: 10.1016/j.dam.2021.10.023.

[24]. Liu H., Lu M., Zhang S. “Anti-Ramsey numbers for cycles in the generalized Petersen graphs”. *Applied Mathematics and Computation*. 430, 127277, (2022). DOI: 10.1016/j.amc.2022.127277.

[25]. Wu K., Su W., Luo H., Xu X. “New lower bounds for seven classical Ramsey numbers R(3,q)”. *Applied Mathematics Letters*. 22, pp 365-368, (2009). DOI: 10.1016/j.aml.2008.04.010.

[26]. Moazzami D. “Tenacity of a graph with maximum connectivity”. *Discrete Applied Mathematics*. 159, pp 367-380, (2011). DOI: 10.1016/j.dam.2010.11.008.

[27]. Merenda J.V.B.S, Bruno O.M. “Using deterministic self-avoiding walks as a small-world metric on Watts–Strogatz networks”. *Physica A: Statistical Mechanics and its Applications*. 621, 128713, (2023). DOI: 10.1016/j.physa.2023.128713.

[28]. Wang E.S., Hong C., Zhang X.H., He N. “Cascading failures with coupled map lattices on Watts–Strogatz networks”. *Physica A: Statistical Mechanics and its Applications*. 525, pp 1038-1045, (2019). DOI: 10.1016/j.physa.2019.04.031.

[29]. Wang S., Mei G., Cuomo S. “A simple and generic paradigm for creating complex networks using the strategy of vertex selecting-and-pairing”. *Future Generation Computer Systems*. 100, pp 994-1004, (2019). DOI: 10.1016/j.future.2019.05.071.

[30]. Watts D.J., Strogatz S.H. “Collective dynamics of small-world networks”. *Nature*. (1998), 393, 440-2. DOI: 10.1038/30918.

[31]. Xia Y., Fan J., Hill D. “Cascading failure in Watts–Strogatz small-world networks”. *Physica A: Statistical Mechanics and its Applications*. 389, pp 1281-5, (2010). DOI: 10.1016/j.physa.2009.11.037.

[32]. Javan A., Javan M., Jafarpour M., Moazzami D., Moieni A.. “On the J-Tightness of Graphs”. *Journal of Algorithms and Computation*. 53, pp 57-74, (2021). DOI: 10.22059/jac.2021.85198.

[33]. Khah S.A., Barati A., Barati H. “A dynamic and multi-level key management method in wireless sensor networks (WSNs)”. *Computer Networks*. 109997, (2023). DOI: 10.1016/j.comnet.2023.109997.

[34]. Nithya B. “Cluster based key management schemes in wireless sensor networks: a survey”. *Procedia Computer Science*. 171, pp 2684-93, (2020). DOI: 10.1016/j.procs.2020.04.291.

[35]. Sowmyadevi D., Shanmugapriya I. “Unsupervised machine learning based key management in wireless sensor networks”. *Measurement: Sensors*. 100847, (2023). DOI: 10.1016/j.measen.2023.100847.

[36]. Jia C., Ding H., Zhang C., Zhang X. “Design of a dynamic key management plan for intelligent building energy management system based on wireless sensor network and blockchain technology”. *Alexandria Engineering Journal*. 60, pp 337-46, (2021). DOI: 10.1016/j.aej.2020.08.019.

[37]. Kalghatgi H., Dhawle M., Raut U. “Defense techniques against spoofing attacks in wireless sensor networks”. *Materials Today: Proceedings*. (2023). DOI: 10.1016/j.matpr.2023.03.357.

[38]. Misbha D.S., Ezhil Roja P. “Lightweight key distribution for secured and energy efficient communication in wireless sensor network: An optimization assisted model”. *High-Confidence Computing*. 3, 100126, (2023). DOI: 10.1016/j.hcc.2023.100126.

[39]. Tran T. “Two problems in graph Ramsey theory”. *European Journal of Combinatorics*. 104, 103552, (2022). DOI: 10.1016/j.ejc.2022.103552.

[40]. Borwein J.M., Bailey D.H., Bailey D. “Mathematics by experiment: Plausible reasoning in the 21st century”. *AK Peters Natick, MA*. pp 259-291, (2004). DOI: 10.1007/978-3-319-32377-0-17.

[41]. Graham R.L., Rothschild B.L., Spencer J.H. “Ramsey Theory”. *Jon Wiley, Sons. Inc., New York*. (1990). ISBN: 978-0-471-50046-9.

[42]. Spencer J. “Large numbers and unprovable theorems”. *The American Mathematical Monthly*. 90, pp 669-75, (1983). DOI: 10.1080/00029890.1983.11971310.

[43]. Wu F., Zhang S., Li B., Xiao J. “Anti-Ramsey numbers for vertex-disjoint triangles”. *Discrete Mathematics*. 346, 113123, (2023). DOI: 10.1016/j.disc.2022.113123.

[44]. Bal D., DeBiasio L., Lo A. “A lower bound on the multicolor Size-Ramsey numbers of paths in hypergraphs”. *European Journal of Combinatorics*. 120, 103969, (2024). DOI: 10.1016/j.ejc.2024.103969.

[45]. Boyadzhiyska S., Lesgourgues T. “On the use of senders for asymmetric tuples of cliques in Ramsey theory”. *Journal of Combinatorial Theory*. 169, pp 63-95, (2024). DOI: 10.1016/j.jctb.2024.05.006.

[46]. Edmonds J. “Removing Ramsey theory: lower bounds with smaller domain size”. *Theoretical computer science*. 172, pp 1-41, (1997). DOI: 10.1016/S0304-3975(96)00020-5.

[47]. Jin Z., Zhou W., Yu T., Sun Y. “Anti-Ramsey number for perfect matchings in 3-regular bipartite graphs”. *Discrete Mathematics*. 347, 114011, (2024). DOI: 10.1016/j.disc.2024.114011.

[48]. Maherani L., Shahsiah M. “On Ramsey numbers of 3-uniform Berge cycles”. *Discrete Mathematics*. 347, 113877, (2024). DOI: 10.1016/j.disc.2024.113877.

[49]. McKay B.D., Radziszowski S.P. “Linear programming in some Ramsey problems”. *Journal of Combinatorial Theory, Series B*. 61, pp 125-32, (1994). DOI: 10.1006/jctb.1994.1038.

[50]. McKay B.D., Radziszowski S.P. “R(4, 5) = 25". *Journal of Graph Theory*. 19, pp 309-22, (1995). DOI: 10.1002/jgt.3190190304.

[51]. Radziszowski S. “Small Ramsey numbers”. *The electronic journal of combinatorics*. (2011). DOI: 10.37236/21.

[52]. Ruotolo J., Song Z.X. “Multicolor Ramsey number for double stars”. *Discrete Mathematics*. 347, 114034, (2024). DOI: 10.1016/j.disc.2024.114034.

[53]. Wang Y., Song Y., Li Y., Liu M. “Multi-color Ramsey numbers of two bipartite graphs”. *Discrete Mathematics*. 347, 114117, (2024). DOI: 10.1016/j.disc.2024.114117.

[54]. Newman M. “Networks: an introduction”. *Oxford University Press*. (2010). ISBN: 978-0199206650.

[55]. Golosovsky M. “Citation analysis and dynamics of citation networks”. *Springer*. (2019). DOI: 10.1007/978-3-030-28169-4.

[56]. Wang H., Moore J.M., Small M., Wang J., Yang H., Gu C. “Epidemic dynamics on higher-dimensional small world networks”. *Applied Mathematics and Computation*. 421, 126911, (2022). DOI: 10.1016/j.amc.2021.126911.

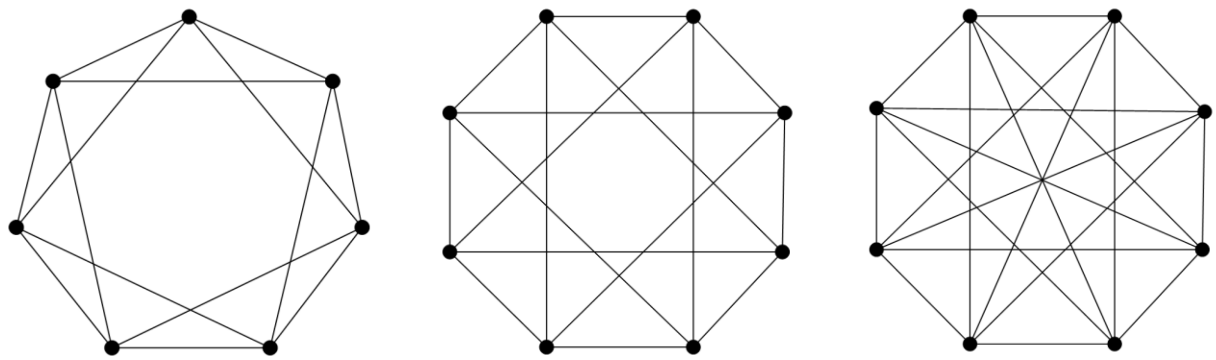
[57]. Wasserman S., Faust K. “Social network analysis: Methods and applications”. *Cambridge University Press*. (1994). ISBN: 978-0521387071.

[58]. Attar N., Aliakbary S., Nezhad Z.H. “Automatic generation of adaptive network models based on similarity to the desired complex network”. *Physica A: Statistical Mechanics and its Applications*. 545, 123353, (2020). DOI: 10.1016/j.physa.2019.123353.

[59]. Li Y., Feng M., Kurths J. “Evolving Network Modeling Driven by the Degree Increase and Decrease Mechanism”. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 53(9), pp 5369-5380, (2023). DOI: 10.1109/TSMC.2023.3268372.

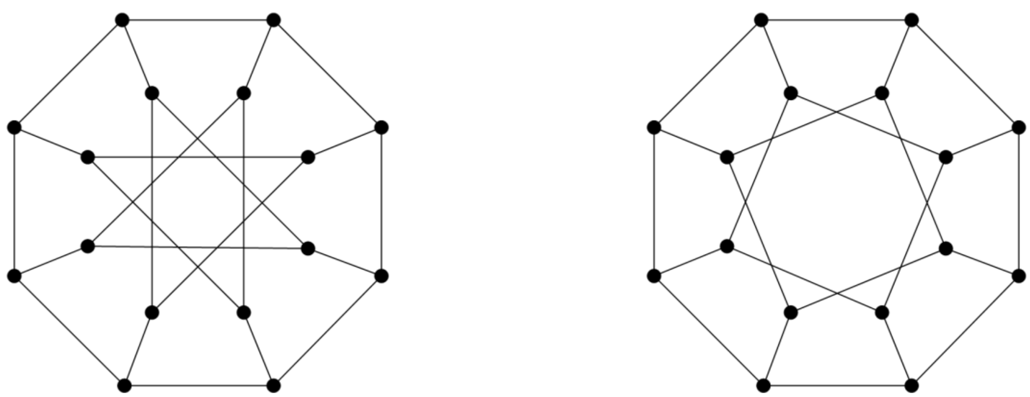
[60]. Zeng Z., Feng M., Kurths J. “Temporal network modeling with online and hidden vertices based on the birth and death process”. *Applied Mathematical Modelling*. 122, pp 151-166, (2023). DOI: 10.1016/j.apm.2023.05.034.

Appendix: Figures



(a) C(7,{1,2}) (b) C(8,{1,3}) (c) C(8,{1,3,4})

Fig. 1: Three different examples of circulant graphs. A graph C(n,{L}) is defined as a circulant graph consisting of n vertices, where each vertex with index i is connected to the vertices with indices i+{L} (modulo n).

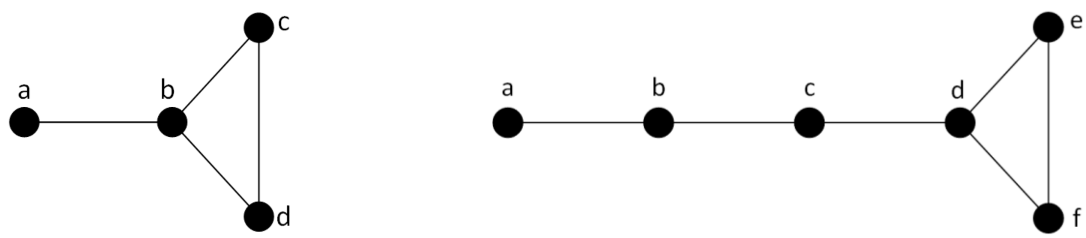


(a) GPG(8,3) (b) GPG(8,2)

Fig. 2: Two different examples of generalized Petersen graphs with different internal circulant graphs. A graph GPG(n, l) is defined as a generalized Petersen graph consisting of 2n vertices, where each vertex of internal set with index i is connected to the other internal vertices with indices i+l (modulo n).

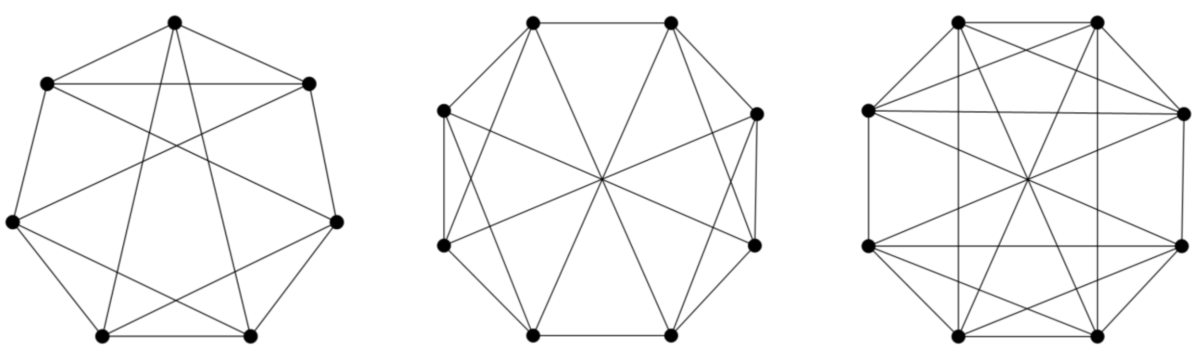
(a) Crown graph (b) Star graph

Fig. 3: Two graphs that generated based on the logic of selecting the vertex with the maximum degree as the source vertex and the farthest vertex from the source as the destination vertex.



(a) (b)

Fig. 4: Graph 4a can be created using the logic of connecting the source vertex to the farthest vertex. However, Graph 4b cannot be generated using this logic.



(a) T(7,4) (b) T(8,4) (c) T(8,5)

Fig. 5: Three different Tight graphs. A graph T(n, m) is defined as an almost m-regular Tight graph with n vertices.

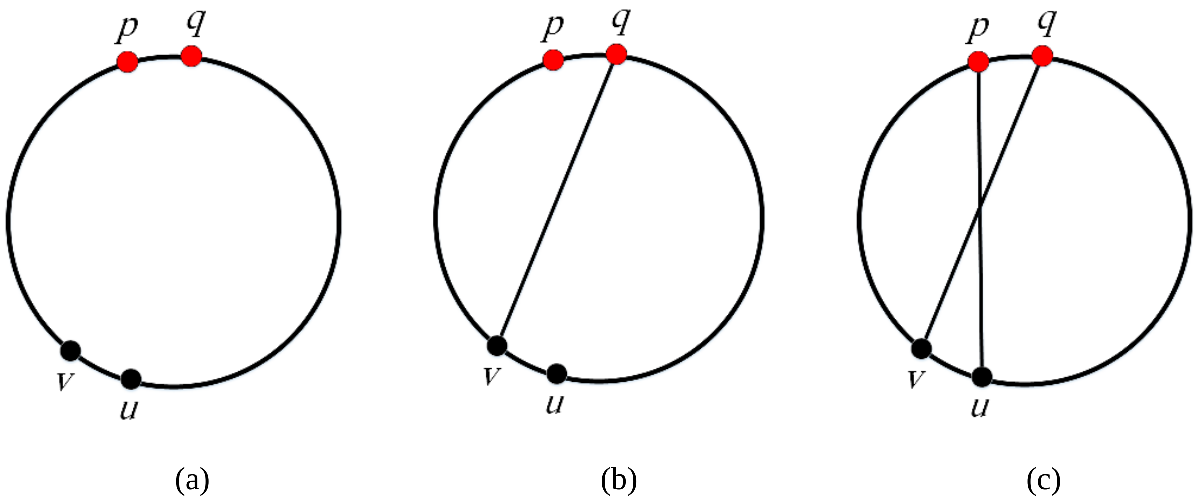


Fig. 6: Three cases that may occur in the final step of Tight graph construction algorithm. In the final step of constructing Tight graphs, the edge (p, q) is removed, and two alternative edges ((v, p) and (u, q)) are added to the graph. This adjustment ensures that the resulting graph achieves an m-regular structure.

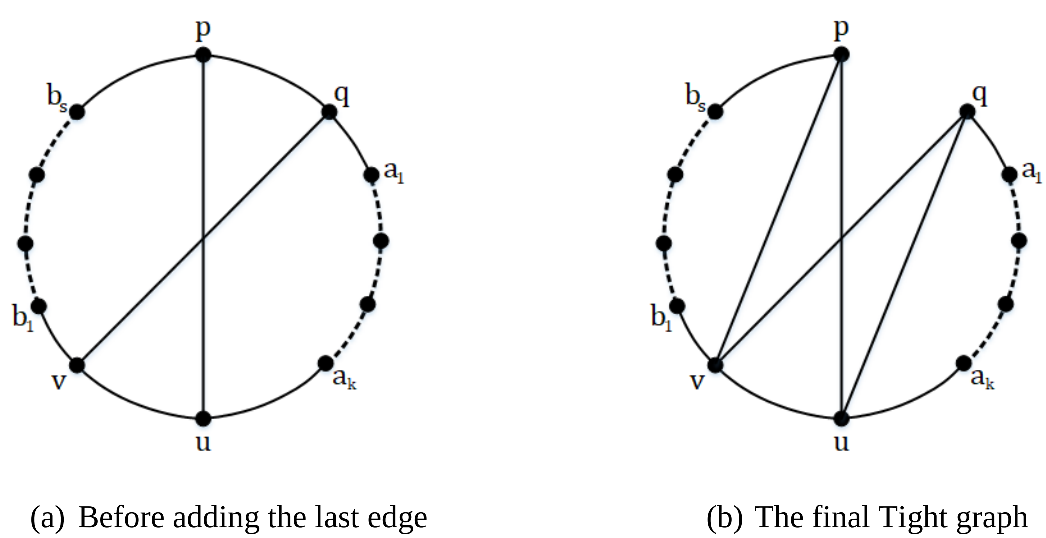
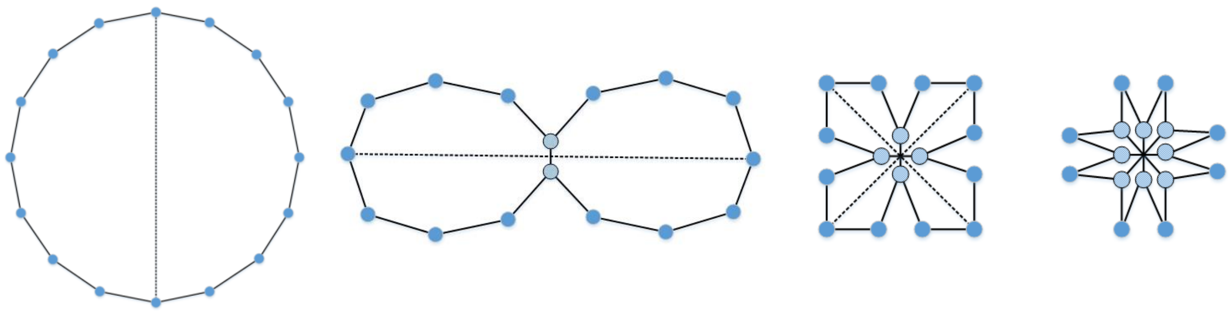
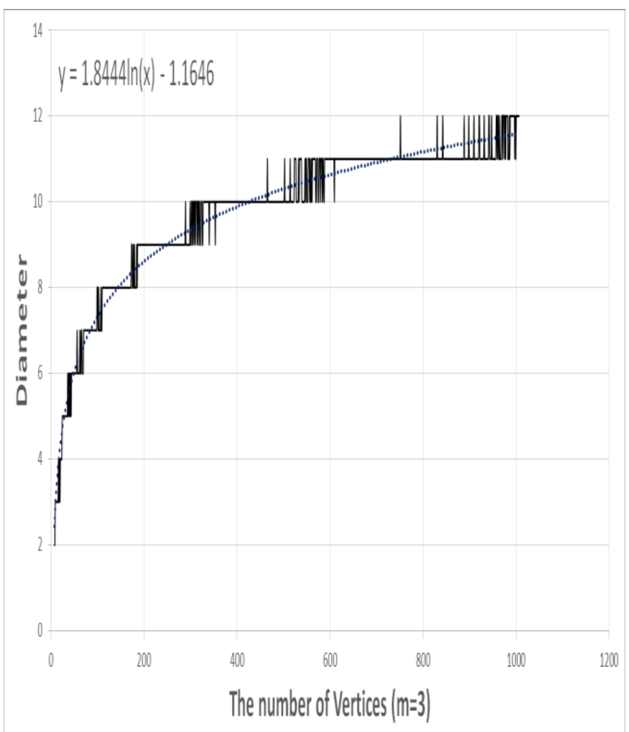


Fig. 7: A special case of the penultimate and final step of Tight graph construction. In this case, the constructed Tight graph will include an alternative Hamiltonian circuit $qa_1a_2\ldots a_kupb_s\ldots b_2b_1vq$.

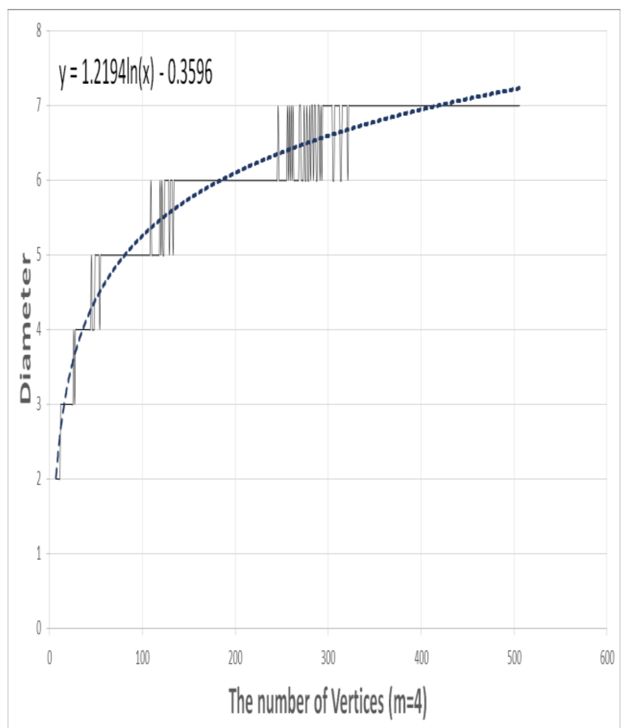


(a) (b) (c) (d)

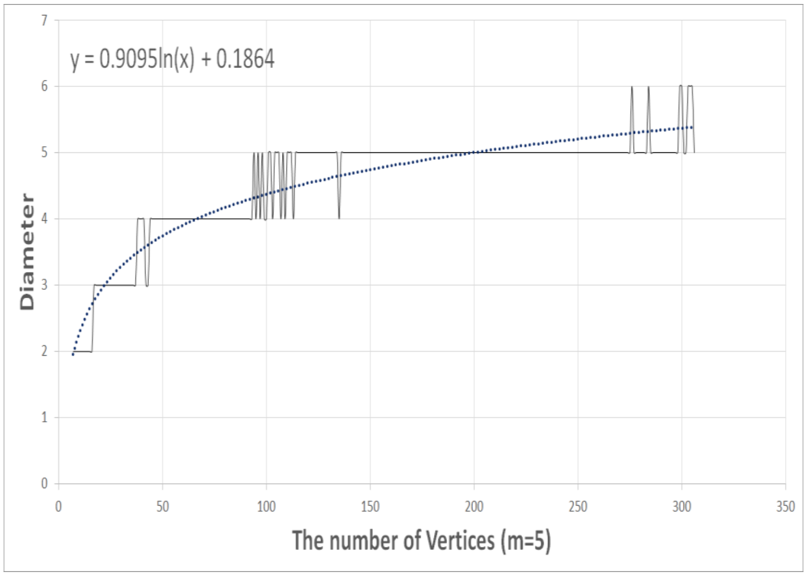
Fig. 8: Graph crumpling steps by adding new edges. By connecting the two vertices that are the furthest distance apart, the graph is expected to become more compact, reducing its overall diameter. This adjustment results in the graph's diameter being logarithmically proportional to the number of vertices, a characteristic commonly associated with efficient small-world networks.



(a) Diameter of Tight graphs for variant n and m=3

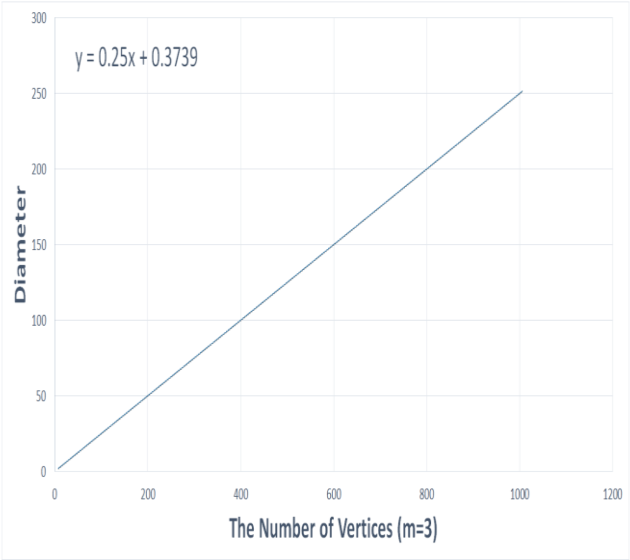


(b) Diameter of Tight graphs for variant n and m=4

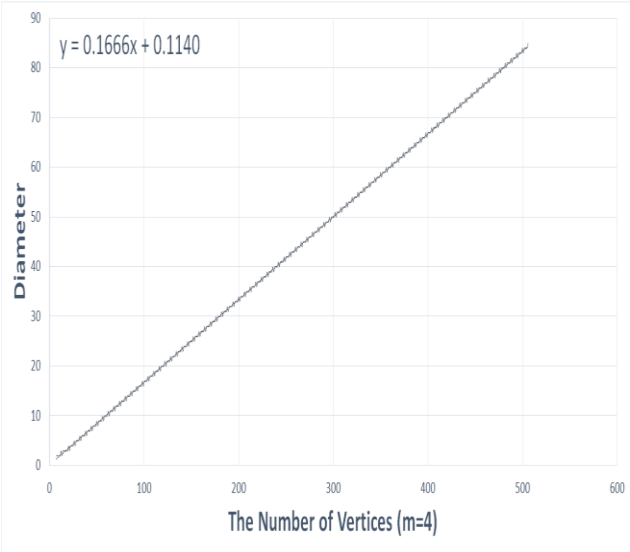


(c) Diameter of Tight graphs for variant n and m=5

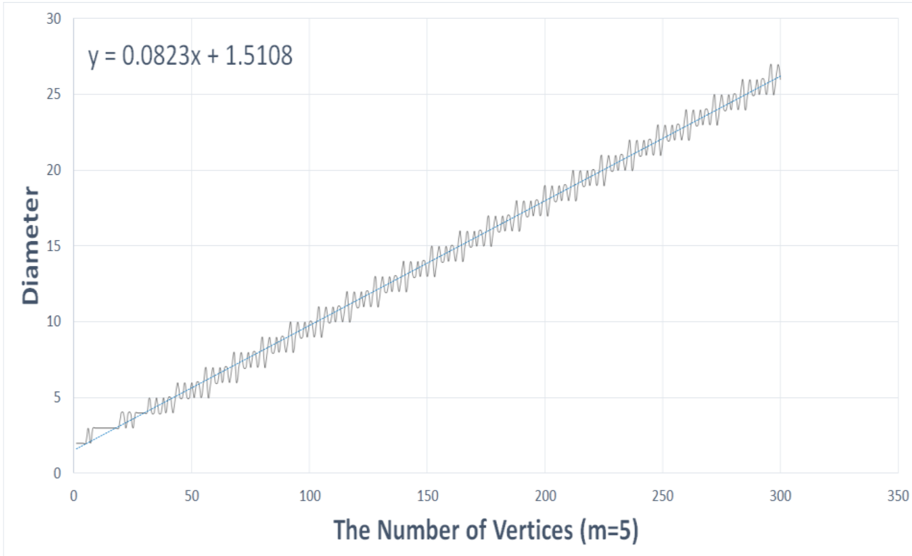
Fig. 9: Diameter change rate vs the number of vertices in Tight graphs. The dotted curve represents the trend line of the main chart.



(a) Diameter of circulant graphs for variant n and m=3

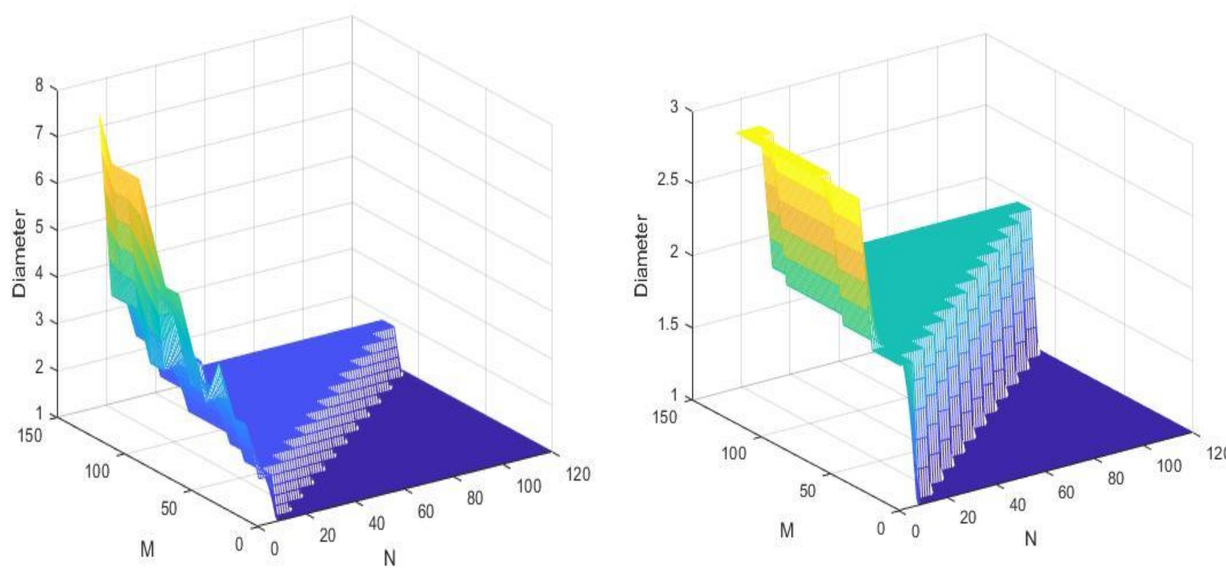


(b) Diameter of circulant graphs for variant n and m=4



(c) Diameter of circulant graphs for variant n and m=5

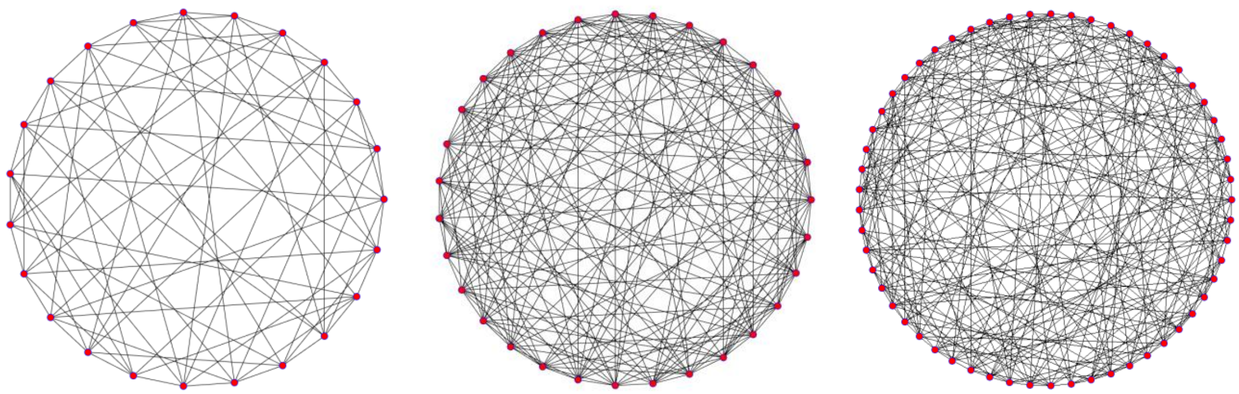
Fig. 10: Diameter change rate vs the number of vertices in circulant graphs. In circulant graphs, the diameter of the graph is directly proportional to the number of vertices, demonstrating a linear relationship.



(a) Diameter of circulant graphs.

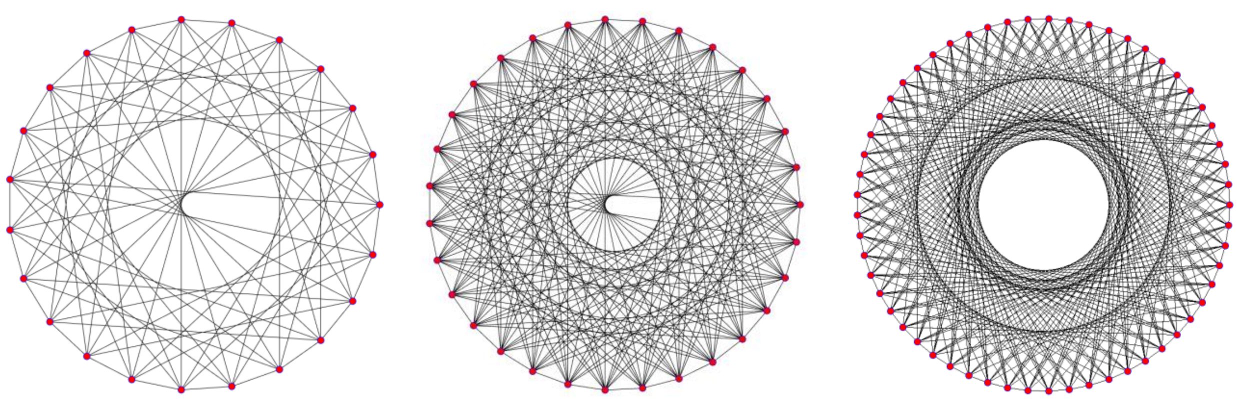
(b) Diameter of Tight graphs.

Fig. 11: Diameter of circulant and Tight graphs with different N and M.



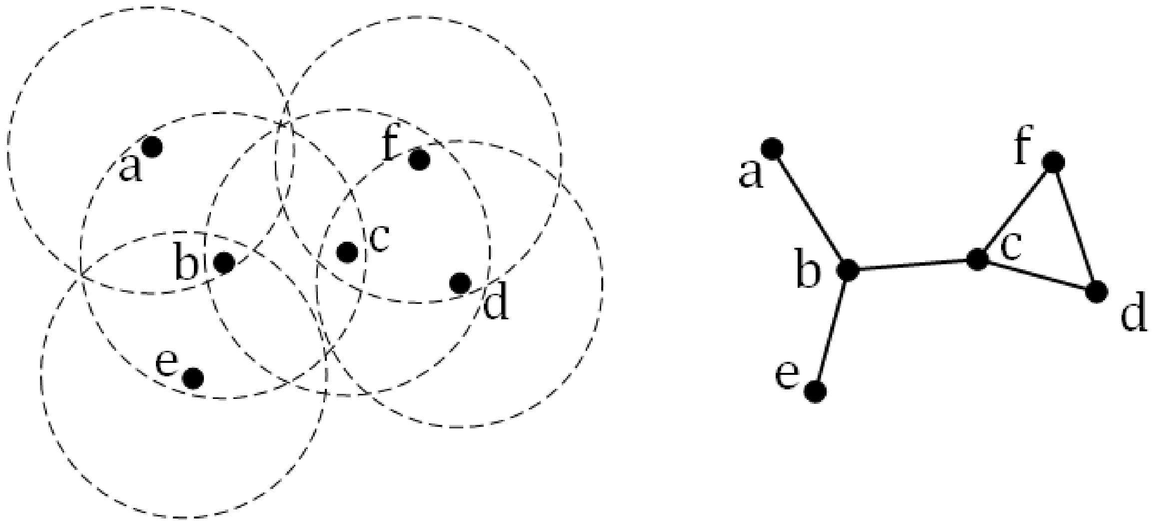
(a) T(23,7) (b) T(31,13) (c) T(57,10)

Fig. 12: Three Tight graphs (T(n, m)) with different n and m, where n is the number of vertices and m represent the expected connectivity. These graphs are approximately m-regular, exhibit a logarithmic diameter relative to the number of vertices n, and possess low vulnerability.



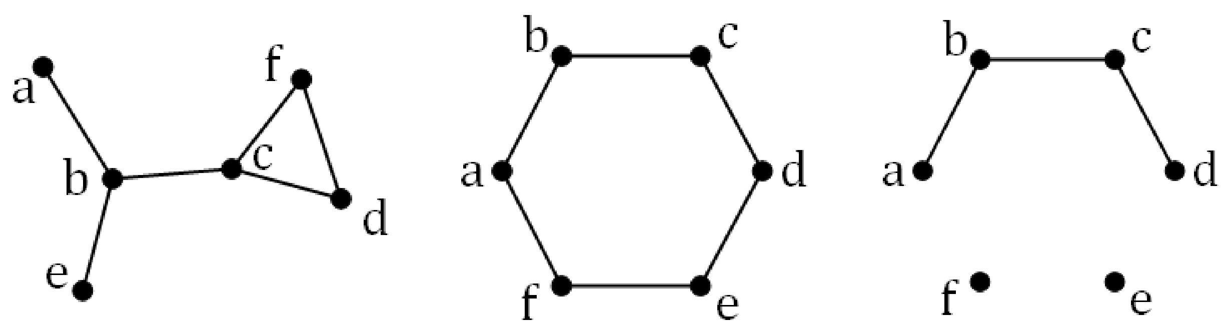
(a) C(23,{7}) (b) C(31,{13}) (c) C(57,{10})

Fig. 13: Three circulant graphs with different n and m, where n is the number of vertices and m represent the expected connectivity. The diameter of this type of graphs is linearly proportional to the number of vertices.



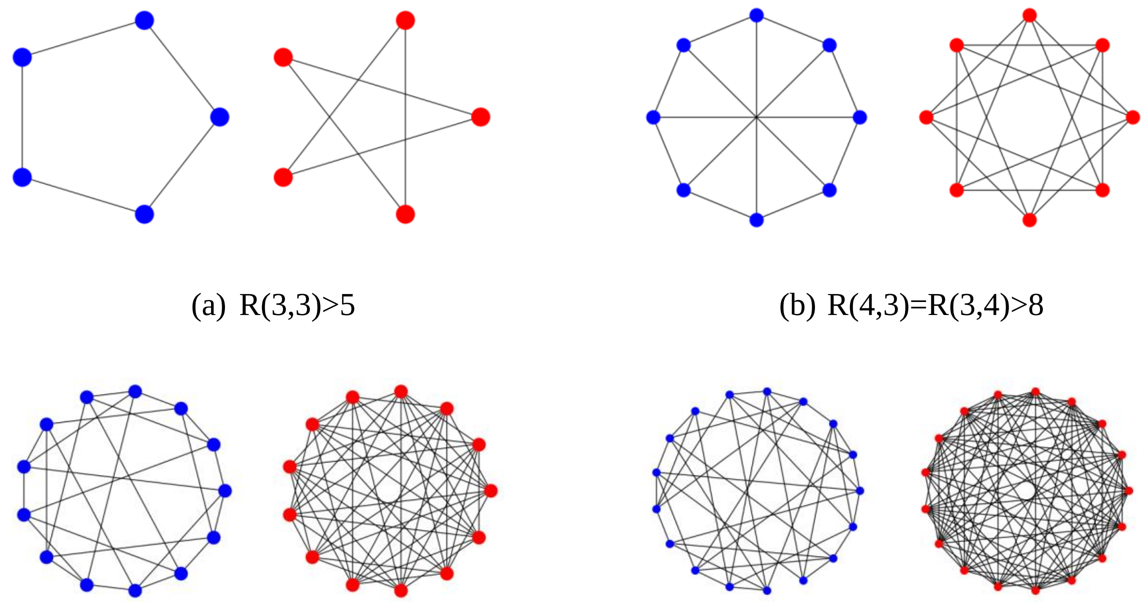
(a) Location of Agents and signal range. (b) Location graph.

Fig. 14: An example of a wireless sensor network (WSN) and its related location graph. Sensors that are within each other's signal range are considered to be directly connected, forming edges in the location graph.



(a) Location graph. (b) Key graph. (c) Communication graph.

Fig. 15: Location and Key graphs of a WSN along with its Communication graph. Sensors can communicate with each other if they are both within each other's signaling range and share a common encryption and decryption key. These conditions define the edges in the communication graph.



(a) R(3,3)>5

(b) R(4,3)=R(3,4)>8

(c) R(5,3)=R(3,5)>13

(d) R(6,3)=R(3,6)>17

Fig. 16: Counterexamples for known Ramsey numbers in Table 1. For example, a counterexample for the Ramsey number R(6,3) involves a pair where the graph on the right lacks a 6-vertex complete subgraph, while its complement, shown on the left, lacks a 3-vertex complete subgraph. The has 17 vertices, demonstrating that the Ramsey number R(6,3) must be greater than 17.

Biographies

Abolfazl Javan: He received his PhD in Algorithms and Computation from the University of Tehran, Iran, in 2024. He is currently an Instructor at the University of Tehran. His research interests include Algorithm Design (Randomized, Approximation, Parameterized), Computation Theory, Cellular Automata, Quantum Computing, Graph Theory, and Complex Networks.

Ali Moeini: He received his PhD in Nonlinear Systems from the University of Sussex, UK, in 1997. He is currently a Full Professor of Electrical & Computer Engineering at the School of Engineering Sciences, College of Engineering, University of Tehran, His research interests include Formal Methods in Software Engineering, Randomized Algorithms, Online Algorithms, Graph Theory, and Complex Networks.