MAUN Fen Bil. Dergi., 13, 1, 46-56 Araştırma Makalesi/ Research Article
MAUN J. of Sci., 13, 1, 46-56 DOI : 10.18586/msufbd.1643589

Analytical Validation of the Malatya Dominating Set Algorithm: Constructing Optimal Dominating Sets Without Redundant Nodes

Şeyda KARCI[1], Fatih OKUMUŞ[2]
[1] Department of Computer Engineering, Faculty of Technology, Marmara University, Istanbul, Türkiye
[2] Department of Software Engineering, Faculty of Engineering, Inonu University, Malatya, Türkiye

: fatih.okumus@inonu.edu.tr [1] 0000-0001-8489-7828 [2] 0000-0003-3046-9558

Received (Geliş): 20.02.2025 Revision (Düzeltme):19.03.2025 Accepted (Kabul): 14.04.2025

ABSTRACT

We revisit the Malatya Dominating Set Algorithm (MDSA) to examine its structure from a theoretical standpoint. Although earlier applications—combining centrality with greedy and dynamic programming—produced promising results, those outcomes lacked formal analysis. In this study, we show that MDSA yields near-optimal solutions on several graph types, including paths, cycles, stars, and bipartite graphs. We explore how MDSA selects nodes and the role centrality plays in that process. In practice, the algorithm often skips over nodes with low relevance, which helps produce smaller and more efficient sets. This observation supports earlier empirical findings, and it also helps explain the reasoning behind the algorithm's behavior. Our interest is not only in confirming its performance but also in gaining a clearer view of how and why it works. The results show that MDSA compares well with other methods for structured graphs where identifying minimal dominating sets is essential.

Keywords: Dominating Sets, Malatya Centrality Value, Malatya Dominating Set Algorithm, Greedy Algorithms, Dynamic Programming, Theoretical Proof

Malatya Hakim Küme Algoritmasının Analitik Doğrulaması: Artık Düğümler Olmadan Optimal Hakim Kümelerin Oluşturulması

ÖZ

Bu çalışmada, Malatya Hâkim Küme Algoritması (MDSA) teorik açıdan yeniden ele alınmaktadır. Merkezilik temelli yaklaşımları açgözlü ve dinamik programlama yöntemleriyle birleştiren algoritma, önceki çalışmalarda çeşitli veri kümeleri üzerinde başarılı sonuçlar üretmiş olsa da bu başarıların ardında güçlü bir teorik temel bulunmamaktadır. Bu kapsamda, MDSA'nın yol, döngü, yıldız ve iki taraflı çizgeler gibi belirli çizge türlerinde optimale yakın çözümler üretebildiği gösterilmiştir. Algoritmanın düğüm seçim süreci ve merkezilik hesaplamalarının bu sürece etkisi ayrıntılı biçimde incelenmiştir. Uygulama sonuçları, önemsiz düğümlerin elenmesiyle daha küçük ve verimli hâkim kümeler elde edildiğini göstermektedir. Bu bulgular, önceki deneysel sonuçlarla örtüşmekte ve algoritmanın karar mekanizmasını açıklamaya yardımcı olmaktadır. Bu çalışma yalnızca algoritmanın başarımını doğrulamakla kalmayıp, aynı zamanda bu başarımın arkasındaki temel ilkeleri de ortaya koymaktadır. Sonuçlar, MDSA'nın yapılandırılmış çizgelerde hâkim küme belirleme problemleri için etkili bir seçenek olduğunu göstermektedir.

Anahtar Kelimeler: Hâkim Kümeler, Malatya Merkeziyet Değeri, Malatya Hâkim Küme Algoritması, Açgözlü Algoritmalar, Dinamik Programlama, Teorik İspat

INTRODUCTION

Mathematical modeling to solve real-world problems is a fundamental approach in many engineering and scientific fields. Graph theory is a structure used in such modeling techniques for many optimization problems [1], [2]. There are many graph application areas such as social networks, computer networks, biological interaction networks, transportation, wireless networks, text mining, text summarization, network flows, circuit connections, molecular simulations, connection networks, etc. [3], [4], [5], [6]. As an NP-hard problem, the Minimum Dominating Set (MDS) problem holds a significant position in graph theory. This problem aims to find the smallest subset of nodes in a graph such that every node is connected to at least one dominant node, provided that the number of elements in the dominating set is minimal [7]. The MDS problem plays an important role in optimizing network structures by increasing efficiency in various areas. In communication networks, the use of MDS enables the selection of key relay nodes and the optimization of transmission load [8], [9]. In social network analysis, there are studies that apply MDS in determining effective nodes that control information dissemination [10], [11], [12], [13]. In biological networks, MDS is one of the methods used in modeling basic protein interactions and disease spread [14], [15], [16]. For the problem of node placement optimization in sensor networks, in cases requiring full coverage, the minimum number of sensor nodes can be determined using MDS [17], [18], [19]. However, the fact that the solution of this problem is NP-hard necessitates the development of efficient algorithms in large-scale graphs. In the literature, various methods such as approximation algorithms, greedy algorithms, heuristic and meta-heuristic methods, and linear programming-based solutions (integer linear programming - ILP) have been proposed. However, the vast majority of these methods either failed to reach the exact optimum solution or had high computational costs [8], [20].

Malatya Dominating Set Algorithm (MDSA), which we introduced in our previous study [21], offers a new method that optimizes the MDS determination process using the MC Values. In 2023, Yakut and colleagues introduced Malatya Centralization, which was subsequently applied to solve maximal independent set problems for simple graphs [22]. For the best matching in bipartite graphs, Öztemiz employed the MC algorithm [23]. The MDSA algorithm takes this centrality system to a further stage and introduces the second centrality concept. In our practices, we have seen that the solution of the minimum dominating set problem using classical centrality values creates limited effects. In the minimum dominating set problem, the "Second Layer Centrality Influence Area" philosophy was created based on the fact that the influence area of a node should be calculated together with the secondary neighborhoods. This philosophy led to the emergence of the Second Malatya Centrality concept. We show empirically in our previous study that MDSA performs effectively on large-scale graphs [21]. Nevertheless, this method requires theoretical validation and analytical demonstration of its best solutions for specific graph types.

The aim of this study is to strengthen the theoretical foundations of MDSA and to mathematically prove that it produces optimal/near-optimal solutions for some special graph structures. The study seeks answers to the following questions:

i. How can it be proven that the MDSA algorithm is optimal for certain types of graphs?
ii. What is the mathematical effect of the MC Value in the minimum dominating set selection process?
iii. How can the theoretical correctness of the MDSA process of eliminating unnecessary nodes be demonstrated?

By answering the above questions, this article aims to support the experimental successes of MDSA with analytical evidence and to demonstrate that this method has a theoretically solid basis.

MALATYA DOMINATIN SET ALGORITHM

The Malatya centrality (MC) value was defined by Yakut and his friends [22] for the first time, and this centrality value was called as the “First Malatya Centrality” (FMC) value. This value can be obtained by using Algorithm 1. The main principle to define the FMC value: “The relative strength of an entity within a community is determined by the strength of its neighbors”. Yakut and his friends to solve many graph theory problems utilized this idea.

Definition 1 (First Malatya Centrality): *Let $G = (V, E)$ be a simple graph with $\forall u \in V$, where N(u) is the collection of u's neighbor nodes other than itself. The summation of node degree of u over its neighbors' node degrees is called the “First Malatya Centrality Value”, and it is denoted as $\Psi_1(\dots)$ (Eq. 1).*

The FMC value computed by using the ratio of corresponding node degree to its neighbors' degrees as shown in Eq. 1.

$$\Psi_1(u) = \left(\sum_{\forall v_j \in N(u)} \frac{d(u)}{d(v_j)}\right) \frac{d_{active}(u)}{d(u)} = \frac{d_{active}(u)}{d(u)} \sum_{\forall v_j \in N(u)} \frac{d(u)}{d(v_j)} \quad (1)$$

Algorithm 1. Computing of the FMC Value

Input: A is adjacency matrix of G = (V, E)
Output: Ψ_1
1 Initialize Ψ_1 as an empty array of size $|V|$.
2 for each vertex $v_i \in V$:
3 if $d(v_i) \neq 0$ compute:
4 $\Psi_1(u) = \frac{d_{active}(u)}{d(u)} \sum_{\forall v_j \in N(u)} \frac{d(u)}{d(v_j)}$
5 return Ψ_1

In Algorithm 1, input A is adjacency matrix of Graph. The elements of adjacency matrix are initialized as: $[p(x)] = 1$ if $p(x)$ is true, otherwise $[p(x)] = 0$. $d(v_i)$ is degree of the node (how many neighbors it has). $d_{active}(v_i)$ is active node degree (whether the connections are active or not depending on certain conditions). $N(v_i)$ is neighborhood set (other nodes that the node is directly connected to). Finally, $\Psi_1(v_i)$ is the MC Value (a decisive metric for MDS).

We defined the Second Malatya Centrality (SMC) value for the first time and they used it to obtain the minimum dominating set for the given graph [21]. The idea/paradigm used in this study is “*The relative strength of an entity within a community is determined by the relative strength of its neighbors*”. The second MC value definition is given in Definition 2 as defined in [21].

Definition 2 (Second Malatya Centrality): *Assume that $G = (V, E)$ is a simple graph, and $\forall u \in V$, where $N(u)$ is the set of neighbor nodes of u except itself. The summation of node relative strength (FMC value) of u over its neighbors' node relative strength (FMC values) is called the “Second Malatya Centrality Value” (SMC).*

This value is denoted as $\Psi_1(\dots)$ and can be calculated with Eq.2.

$$\Psi_2(v_i) = \left(\sum_{\forall v_j \in N(v_i)} \frac{\Psi_1(v_i)}{\Psi_1(v_j)} \right) \frac{1}{d(v_i)} \frac{d_{active}(v_i)}{d(v_i)} = \frac{1}{d(v_i)} \frac{d_{active}(v_i)}{d(v_i)} \sum_{\forall v_j \in N(v_i)} \frac{\Psi_1(v_i)}{\Psi_1(v_j)} \quad (2)$$

The SMC value is quite suitable for the nature of the dominant set problem since the coverage area can extend to the tertiary neighborhood (another dominant node). The algorithm of the SMC values starts with getting the FMC values, and continue with new centrality values from FMC values. This computation steps shown in Algorithm 2.

Algorithm 2. Computing of the SMC Value

Inputs:

A ← Adjacency matrix of graph $G = (V, E)$

Ψ_1 ← FMC Value

Output: Ψ_2

```
Compute Ψ1 using "Algorithm 1".
Initialize Ψ2 as an empty array of size |V|.
for each vertex v_i ∈ V:
  if d(v_i) ≠ 0 compute:
      Ψ2(u) = (1/d(v_i)) (d_active(v_i)/d(v_i)) Σ_{∀v_j∈N(v_i)} Ψ1(v_i)/Ψ1(v_j)
return Ψ2
```

Node Selection Strategy: All of the nodes in the provided graph are initially designated as white nodes. The node with the highest centrality value is chosen following the computation of the graph's SMC values. All of its neighbors are designated as gray nodes, and this node is added to the minimum dominating set. Centrality values are then computed once more. The node with the highest centrality value among the white nodes is identified. The node with the highest value is identified among the grey nodes. The selection of the grey node occurs if its centrality value exceeds the white node's maximum centrality value and if its number of neighbors is at least two times bigger than the white node's. The white node is chosen in the alternative scenario. This process is repeated until there are no white nodes left in the graph. Thus, it is ensured that all nodes are covered under a minimum dominating set. This situation is demonstrated by Algorithm 3.

Algorithm 3. Overall Minimum Dominating Set Algorithm

Inputs:

A ← Adjacency matrix of graph $G = (V, E)$

Output: $V_D \subseteq V$, V_D is optimal minimum dominating set

```
Initialize all nodes v_i with color 0 (white), indicating
they are unprocessed.
while at least one white node remains:
  Compute Ψ2 using "Algorithm 2"
   u = argmax{Ψ2(v)}
  Mark u as a dominating node (color = 2, black).
   V_D = V_D ∪ u
  Mark all neighbors of u as covered (color = 1, grey).
  Update adjacency matrix A to remove connections
  involving u:
    A(u,k) ← 0 and A(k,u) ← 0 for all k in N(u).
    If any neighbor r of u has degree = 1, remove its
    connections: A(r,k) ← 0 and A(k,r) ← 0
Eliminate redundant nodes
Return V_D
```

Principle 1: *An entity's strength ought to be measured in relation to the strength of its neighbors within its community (Fig.1).*

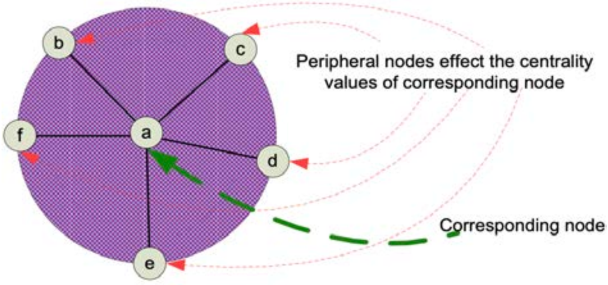


Figure 1. The central is the node whose centrality value is subject to computation. The peripheral nodes are used to compute the first centrality values.

The power of a node is proportional to the powers of that node relative to its neighbors. The MC value of node ‘a’ is computed by using its degree and its neighbors’ degrees. We called this concept as First Layer Centrality Effect Region. In literature, Jiang and Zheng defined a new graph for the given graph, and they called it as 2-hop graph [24]. Assume that, $G = (V, E)$ is a graph and the corresponding 2-hop graph is:

$$G^2 = (V, E_2) \quad (3)$$

$$E_2 = \{(u, v) | (u, v) \in E \text{ or } N(u) \cap N(v) \neq \emptyset\} \quad (4)$$

Two nodes u and v are adjacent in G^2 if and only if u and v are 2-hop adjacent in G. G^2 is called the 2-hop graph of G. The similar logic or philosophy was used in Malatya Dominating Set Algorithm, and due to this case, this algorithm is an effective algorithm. In Fig.1, the aim is to compute the FMC value of node “a”. So that the degree of node “a” is over the degrees of nodes “b”, “c”, “d”, “e”, “f”. The summation of these ratios give the FMC value of node “a” (Fig.1).

Principle 2: *An entity's relative strength can be defined as its strength in relation to its neighbors' relative strengths within its society (Fig.2).*

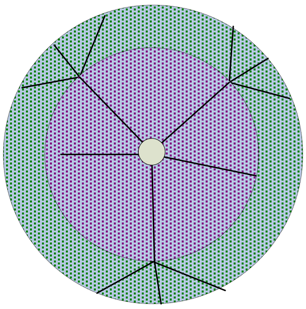


Figure 2. The central is the node whose centrality value is subject to computation. The peripheral nodes are used to compute the second centrality values.

The SMC value for node "a" can be computed similar to the FMC value. The FMC values are used to determine the independent set, vertex-cover sets. It is not suitable to determine the MDS. In order to determine the MDS, the SMC values are used. This case is called as "Second Layer Centrality Effect Region" (Fig.2). When a node is selected for MDS such as node "a", the next node selected for MDS may be neighbor of one of the nodes is set {g, h, i, j, k, m, n, p, q}. This philosophy should be used for all MDS algorithms.

Definition 3: *Assume that $G = (V, E)$ is a graph and V_D is the minimum dominating set for G obtained by using Malatya Dominating Set Algorithm. $\exists u \in V_D, \forall v_i \in N(u)$, all v_i have got more than one black node, then node u is called* ***redundant*** *node.*

The Redundant Node Elimination algorithm changes the state of redundant node in V_D (the minimum dominating set). The Malatya Dominating Set finalized with node colors as grey and black. If the grey neighbors of the nodes in the MDS have more than one black neighbor, two or more black neighbors of the grey are added to the grey nodes list and the grey node is added to the black nodes list.

VERIFYING OPTIMALITY OF MALATYA DOMINATING SET FOR SPECIAL GRAPHS

The Malatya Dominating Set algorithm can be verified that it is optimum for some important graphs.

Theorem 1*: Malatya Dominating Set algorithm is optimum for path graph P_n.*

Proof*: Assume that $G = (V, E)$ is a path graph of size n, Fig. 3 illustrates the first step of determining the first element of V_D.*

Algorithm 3 should be used to the graph shown in Fig. 3 in order to determine the best dominating set. Fig. 3 displays the algorithm's results. The SMC values are calculated using Algorithm 2 for the initial step of Algorithm 3 following the computation of the FMC values. Fig. 3 (a) displays the MC values for the original graph, whereas Fig. 3 (b) displays the values for the revised graph. The first element chosen for V_D is node "2," while the second element chosen for V_D is node "5". This is how the process continues. Table 1 lists the cardinalities of dominating set for various P_n sizes.

Table 1. The cardinalities of V_D obtained by the Malatya Dominating Set algorithm for different sizes of P_n.

n	3	4	5	6	7	8	9	10	11	12	13	14	15	16	…
$\lvert V_D \rvert$	1	2	2	2	3	3	3	4	4	4	5	5	5	6	…

Figure 3. MC values for the original Path graph are shown in (a), while the values for the revised graph are shown in (b).

The optimum cardinalities of VD can be computed by Eq.5.

$$|V_D| = \begin{cases} \left\lfloor \frac{n}{3} \right\rfloor + 1, if\ n\ mod 3 = 1 \\ \left\lfloor \frac{n}{3} \right\rfloor + 1, if\ n\ mod 3 = 2 \\ \left\lfloor \frac{n}{3} \right\rfloor, \quad if\ n\ mod 3 = 0 \end{cases} \quad (5)$$

In other word, it is seen that the formula for the cardinality of V_D is $|V_D| = \lceil \frac{n}{3} \rceil$ where $\lceil \dots \rceil$ is a ceiling function ■

Theorem 2: *MDSA is optimum for cycle graph C_n.*

Proof: Assume that $G = (V, E)$ is a cycle graph of size n and Fig. 4 illustrates a cycle graph. The FMC values are all 2 for all nodes and the SMC values for all nodes are 1. In this case, any node is randomly selected and selected node and its neighbors are colored as grey. This process concludes in path graph. The optimum case of Malatya Dominating Set algorithm for path graph was verified in Theorem 2. The node '1' is selected randomly, then selected node is removed from graph and its neighbors (1, 2, n) are colored as grey C_n, and this case concludes in path graph P_{n-3}. In the case of cycle graph, the neighbors of selected node are also removed from given graph, since all grey nodes have active degrees as 1. The Malatya Dominating Set algorithm obtains the following minimum dominating set cardinalities with respect to cycle sizes (Table 2 illustrates these cases). The selected node and its neighbors are removed from the given graph, and this case is $P_{(n-3)} = C_n - \{1,2,n\}$. The obtained result is a path and the Malatya Dominating Set algorithm is optimum for path graph.

Table 1. The cardinalities of V_D obtained by the Malatya Dominating Set algorithm for different sizes of C_n.

n	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**…**
$\lvert V_D \rvert$	1	2	2	2	3	3	3	4	4	4	5	5	5	6	…

The cardinality of V_D is given in Eq.5. In other way, it is seen that the formula for the cardinality of V_D is $|V_D| = \lceil \frac{n}{3} \rceil$ where $\lceil \dots \rceil$ is a ceiling function ■

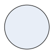


Figure 4. Cycle graph and at the first step node '1' is selected.

Theorem 3: *Malatya Dominating Set algorithm is optimum for star graph S_n.*

Proof: Assume that $S_n = (V, E)$ is a star graph as seen in Fig.5. The node degrees $d(2) = d(3) = \cdots = d(n) = 1$ and $d(1) = n - 1$. So, the FMC value for all nodes is:

$$\Psi_1(2) = \Psi_1(3) = \Psi_1(4) = \cdots = \Psi_1(n) = \frac{1}{n-1} \text{ and}$$
$$\Psi_1(1) = \frac{\frac{n-1}{1} + \frac{n-1}{1} + \cdots + \frac{n-1}{1}}{n-1} = (n-1)^2 \tag{6}$$

$$\Psi_2(2) = \Psi_2(3) = \Psi_2(4) = \cdots = \Psi_2(n) = \frac{\frac{1}{n-1}}{(n-1)^2} = \frac{1}{(n-1)^3}$$
$$\text{and } \Psi_2(1) = \left(\left(\frac{(n-1)^2}{\frac{1}{n-1}}\right)/(n-1)\right)\frac{1}{(e^{n-1})} \tag{7}$$

The structural visualization of star graph is shown in Fig. 5.

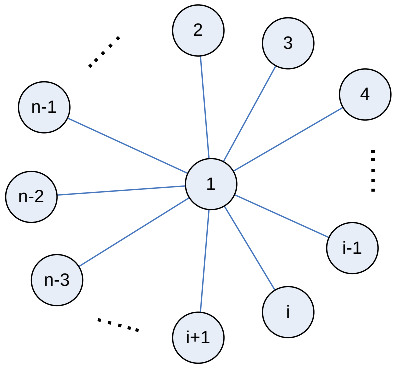


Figure 5. Star graph of size n.

The node '1' has maximum SMC value and it is subject to select for minimum dominating set. The remaining nodes are all neighbors of node '1', and they all are added to set V_N. This is the optimum result for star graph ■

Theorem 4*: Malatya Dominating Set algorithm does not include redundant nodes for bipartite graph $G = (V, E)$ where $V = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$, $|V_1| = m$, $|V_2| = n$, $m \geq 2$, $n \geq 2$.*

Proof: The proof can be handled for bipartite graphs in two cases.

Case a: Assume that $K_{m,n} = (V, E)$ is a grid graph where $|V| = m + n$, and $V = V_1 \cup U$, $V_1 \cap U = \emptyset$. The optimality of the Malatya Dominating Set algorithm for given graph (Fig.6).

Figure 6. Complete bipartite graph.

Without losing generality, assume that $m > n$. The SMC values are as follow:

$$\forall v_i \in V_1 \Rightarrow \Psi_1(v_i) = \frac{n^2}{m}$$
$$\text{and } \forall u_i \in U \Rightarrow \Psi_1(u_i) = \frac{m^2}{n} \tag{8}$$

$$\forall v_i \in V_1 \Rightarrow \Psi_2(v_i) = \frac{\frac{n^2}{m}}{\frac{m^2}{n}} \cdot \frac{1}{n} = \frac{n^2}{m^3} \tag{9}$$

$$\text{and } \forall u_i \in U \Rightarrow \Psi_2(u_i) = \frac{\frac{m^2}{n}}{\frac{n^2}{m}} \cdot \frac{1}{m} = \frac{m^2}{n^3}$$

$\frac{n^2}{m^3} < 1$ and $\frac{m^2}{n^3} > 1$, then one of the nodes in U is selected for the minimum dominating set. After this step, the nodes in V_1 are marked as grey and all nodes in U have active degrees as zero. Due to this case, all the remaining nodes in U are added to the minimum dominating set. The $n > m$ case is similar to this case.

Case b: Assume that G is not a complete bipartite graph and $G = (V, E)$ is a bipartite graph where $V = V1 \cup V_2$, $V_1 \cap V_2 = \emptyset$, $|V_1| = n$ and $|V_2| = m$. Thus, the situation in Eq.10 will occur.

$$\begin{aligned} &\forall u_i \in V_1,\ 1 \le i \le n,\ 1 \le d(u_i) \le m \\ &\text{and } \forall v_j \in V_2\ 1 \le j \le m,\ 1 \le d(v_j) \le n \end{aligned} \tag{10}$$

Malatya Dominating Set algorithm consists of four steps: Computation of the FMC values, the SMC values, selecting maximum arguments and the redundant node eliminations. Without losing generality, assume that $\forall u_i \in V_1,\ 1 \le i \le n,\ d(u_i) = r$ and $\forall v_j \in V_2\ 1 \le j \le m,\ d(v_j) = k$. This situation can be seen in Eq. 11.

$$\begin{aligned} &\Psi_1(u_i) = \frac{r}{k} r = \frac{r^2}{k} \text{ and } \Psi_1(v_j) = \frac{k}{r} k = \frac{k^2}{r} \\ &\text{and } \Psi_2(u_i) = \frac{\frac{r^2}{k}}{\frac{k^2}{r}} \frac{1}{r} = \frac{r^2}{k^3} \text{ and } \Psi_2(v_j) = \frac{\frac{k^2}{r}}{\frac{r^2}{k}} \frac{1}{k} = \frac{k^2}{r^3} \end{aligned} \tag{11}$$

If $r > k$ then $\Psi_2(u_i) > \Psi_2(v_j) \Rightarrow \frac{r^2}{k} > \frac{k^2}{r^3}$. A node in V_1 can be selected for the minimum dominating set. After selection of a node to minimum dominating set makes decreasing in degrees of r nodes in V_2 by 1 (as seen in Fig.7). The FMC values for nodes in V_2 are seen in Fig.7.

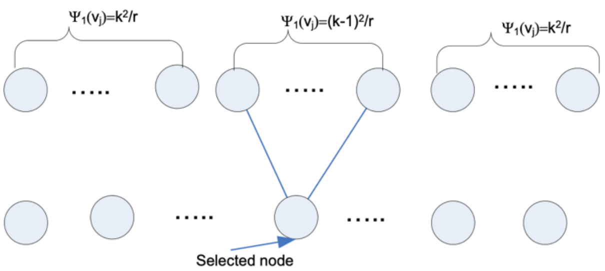


Figure 7. The node selection process changes degrees of some nodes.

The new state of graph is as follow: $|V_1| = m - 1$, $|V_2| = n$, and there are r grey nodes, one black node. The remaining nodes are white nodes. The remaining nodes in V_1 have the following the FMC values (the nodes in V_1 can be grouped into two ways with respect to node degrees).

The nodes (assume that these nodes are denoted as u_i have degrees as k-1. Thus, $\Psi_1(u_i)$ is:

$$\Psi_1(u_i) = \left(\frac{r}{k-1} + \frac{r(r-1)}{k}\right)\frac{r-1}{r} = \frac{(r-1)(rk-r+1)}{k(k-1)} \tag{12}$$

and the remaining nodes have the following the FMC values:

$$\Psi_1(u_i) = \frac{r}{k} r = \frac{r^2}{k} \tag{13}$$

All nodes in given graph are grouped into four groups.

a) The nodes in $v_j \in V_2$ have degrees as k-1, and so their SMC values are:

$$\begin{aligned} \Psi_2(v_j) &= \left[\frac{\frac{(k-1)^2}{r}}{\frac{(r-1)(rk-r+1)}{k(k-1)}}\gamma + \frac{\frac{(k-1)^2}{r}}{\frac{r^2}{k}}(k-\gamma)\right]\frac{1}{k-1} \\ &= \frac{\gamma k(k-1)^2}{r(r-1)(rk-r+1)} + \frac{k(k-1)(k-\gamma)}{r^2} \end{aligned} \tag{14}$$

where γ is the number of grey neighbors of node v_j.

b) The nodes in $v_j \in V_2$ have degrees as 'k', and so their SMC values are:

$$\begin{aligned} \Psi_2(v_j) &= \left[\frac{\frac{k^2}{r}}{\frac{(r-1)(rk-r+1)}{k(k-1)}}\gamma + \frac{\frac{k^2}{r}}{\frac{r^2}{k}}(k-\gamma)\right]\frac{1}{k} \\ &= \frac{\gamma k^2(k-1)}{r(r-1)(rk-r+1)} + \frac{(k-\gamma)k^2}{r^3} \end{aligned} \tag{15}$$

where γ is the number of grey neighbors of node v_j.

c) The nodes in $u_i \in V_1$ have degrees as $r - 1$ (have γ neighbors), and so their SMC values are:

$$\begin{aligned} \Psi_2(u_i) &= \frac{r-1-\gamma}{(r-1)^2}\left[\frac{r(r-1)(rk-r+1)}{k^3(k-1)}\gamma + \right. \\ &\qquad \left. \frac{r(r-1(rk-r+1)}{k(k-1)^3}(r-1-\gamma)\right] \\ &= \left[\frac{r(rk-r+1)}{k^3(k-1)}\gamma + \frac{r(rk-r+1)}{k(k-1)^3}(r-1-\gamma)\right]\frac{r-1-\gamma}{(r-1)} \end{aligned} \tag{16}$$

d) The nodes in $u_i \in V_1$ have degrees as r (have γ neighbors), and so their SMC values are:

$$\begin{aligned} \Psi_2(u_i) &= \left[\frac{r^3}{k^3}\gamma + \frac{r^3}{k(k-1)^2}(r-\gamma)\right]\frac{r-\gamma}{r^2} \\ &= \left[\frac{r}{k^3}\gamma + \frac{r}{k(k-1)^2}(r-\gamma)\right](r-\gamma) \end{aligned} \tag{17}$$

There are four cases to compute the SMC values of all nodes in the changed graph. Assume that $k = r$, then the centrality values are simplified as follows (at the second step, all nodes in V_2 do not have grey neighbors).

a) $\Psi_2(v_j) = \frac{(k-1)}{k^2-k+1} + \frac{(k-1)}{k}$
b) $\Psi_2(v_j) = \frac{k}{k^2-k+1} + \frac{1}{k}$
c) $\Psi_2(u_i) = \left[\frac{\gamma(k^2-k+1)}{k^2(k-1)^2} + \frac{k^2-k+1}{(k-1)^4}(k-\gamma-1)\right](k-\gamma-1)$
d) $\Psi_2(u_i) = \left[\frac{1}{k^2}\gamma + \frac{1}{(k-1)^2}(k-\gamma)\right](k-\gamma)$

Without losing generality, lets $k = 4$, then

a) $\Psi_2(v_j) = \frac{(k-1)}{k^2-k+1} + \frac{(k-1)}{k} \cong 0.98$

b) $\Psi_2(v_j) = \frac{k}{k^2-k+1} + \frac{1}{k} \cong 0.33$

c) $\Psi_2(u_i) = \left[\frac{k^2-k+1}{k(k-1)^2}\gamma + \frac{k^2-k+1}{(k-1)^4}(k-1-\gamma)\right](k-1-\gamma)$
$= \frac{156+65\gamma}{324}(3-\gamma)$

d) $\Psi_2(u_i) = \left[\frac{1}{k^2}\gamma + \frac{1}{(k-1)^2}(k-\gamma)\right](k-\gamma) = \frac{64-7\gamma}{144}(4-\gamma)$

In case of $\gamma = 1$,

a) $\Psi_2(v_j) \cong 0.98$

b) $\Psi_2(v_j) \cong 0.33$

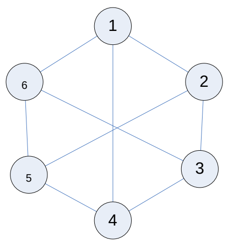
c) $\Psi_2(u_i) \cong 1.36$

d) $\Psi_2(u_i) \cong 1.19$

In this case, all nodes in V_1 are white nodes, and the selected node in the second must be element of V_1. This process is carrying on until all nodes are black or grey ■

***Theorem 5**: Malatya Dominating Set algorithm is optimum for a regular graph of degree (girth graph) 3.*

Proof: The Malatya Dominating Set algorithm obtains the minimum dominating set for the given graph. The proof will be handled by using a similar way to mathematical induction. A sample of 3-degree graph is shown in Fig. 8.



Nodes	1	2	3	4	5	6
Ψ_1	3	3	3	3	3	3
Ψ_2	1	1	1	1	1	1

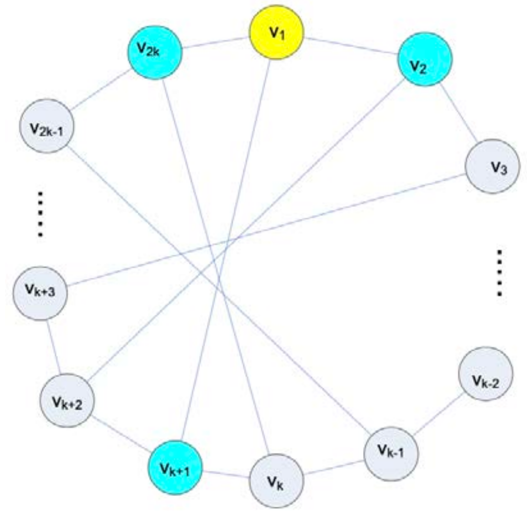
Figure 8. A graph of degree 3

In Fig.8, all the second centrality values are equals and any node is selected to minimum dominating set. Assume that this node is 2, and the nodes 1, 3 and 5 are marked as grey. The nodes 4 and 6 are white nodes of active degrees are equal to zero. In this case, both nodes are added to the minimum dominating set. The obtained result is optimum. The same process can be handled on the graph as seen in Fig.9.

Initially, all nodes of graph seen in Fig.9(a) have 1 as the SMC values. The yellow node is selected to the Minimum Dominating Set V_D, and cyan nodes (grey nodes) are marked as grey nodes. Fig.9(b) illustrates node with the SMC values, and at this case, node v_4 can be selected for the Minimum Dominating Set. The nodes v_3, v_5 and v_{k+4} are marked as grey nodes.

The process can be specified as a principle such as if the node v_r selected to the minimum dominating set, the nodes v_{r-1}, v_{r+1} and v_{k+r} are marked as grey nodes (assume that $|V| = n = 2k$). Each step covers four nodes, and if the number n is multiple of the number 4, algorithm determines V_D of size as $n/4$ ■

a)



b)

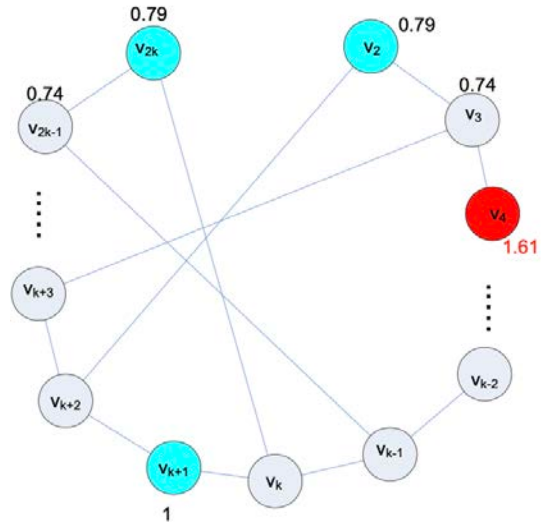


Figure 9. A regular graph $G = (V, E)$ where $|V| = n = 2k$ of degree 3.

***Theorem 6**: Malatya Dominating Set algorithm is optimum for a regular graph like hypercubes.*

Proof: Assume that $H_d = (V, E)$ is a hypercube of size n and dimension $d = lgn$ where $n = 2^d$. All node degrees are equal to $d = lgn$ where n is the number of nodes in H_d. Initially, all nodes have active degrees as $d = lgn$. This situation can be seen in Eq.18 and Eq. 19.

$$\forall v_i \in V, \Psi_1(v_i) = \overbrace{\frac{lgn}{lgn} + \frac{lgn}{lgn} + \ldots + \frac{lgn}{lgn}}^{lgn} \frac{lgn}{lgn} = lgn \quad (18)$$

$$\forall v_i \in V, \Psi_2(v_i) = \overbrace{\frac{lgn}{lgn} + \frac{lgn}{lgn} + \cdots + \frac{lgn}{lgn}}^{lgn} \frac{lgn}{lgn} \frac{1}{lgn} = 1 \quad (19)$$

In this case, any node can be selected to minimum dominating set, and assume that the selected node is u. The $N(u)$ are marked as grey (gray) nodes. The neighbors of grey nodes are depicted as red nodes. The green node and its edges are removed from original graph. Fig.10 illustrates the H5 after one node selection process. Fig.10 depicts that there are four node types with respect to their neighbors' status. The red nodes are 2-hop nodes to the selected node, and white nodes are 3-hop nodes with respect to the selected node. In order to select next node to the minimum dominating set, the centrality values should be re-computed for the re-formed graph after deletion of the selected node (green node). The FMC values are as follows for all types of nodes. Eqs.20-22 are symbolized this situation.

$$\forall v_{grey} \in V, \Psi_1(v_{grey}) = (lgn-1)\frac{lgn-1}{lgn} = \frac{(lgn-1)^2}{lgn} \quad (20)$$

$$\forall v_{red} \in V, \Psi_1(v_{red}) = \frac{2lgn}{lgn-1} + (lgn-2) = \frac{2lgn + (lgn-1)(lgn-2)}{lgn} \quad (21)$$

$$\forall v_{white} \in V, \Psi_1(v_{white}) = \frac{lgn}{lgn} lgn = lgn \quad (22)$$

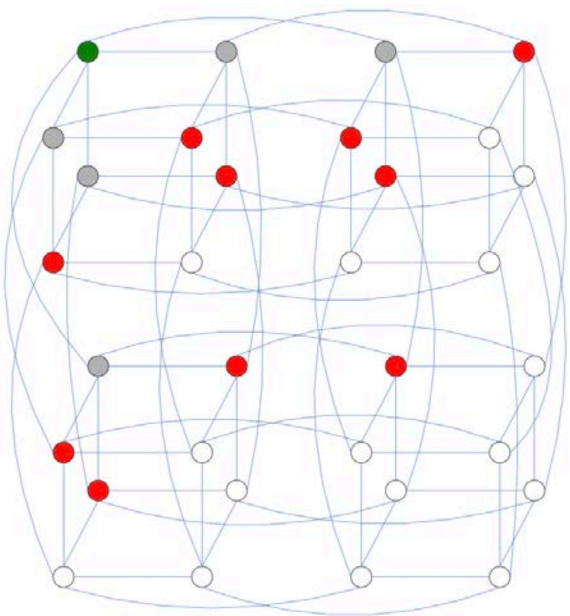


Figure 10. Hypercube of size 5 and the selected node and its neighbors.

The SMC values are as follows for all types of nodes:

$$\forall v_{grey} \in V, \Psi_2(v_{grey}) = \frac{\frac{(lgn-1)^2}{lgn}}{\frac{2lgn + (lgn-1)(lgn-2)}{lgn}} \frac{lgn-1}{lgn-1} \frac{1}{lgn-1} = \frac{lgn-1}{lg^2n - lgn + 2} \quad (23)$$

$$\forall v_{red} \in V, \Psi_2(v_{red}) = \left((lgn-2) \frac{\frac{2lgn + (lgn-1)(lgn-2)}{lgn}}{\frac{lgn}{lgn-2}} + 2\frac{lg^2n - lgn + 2}{(lgn-1)^2} \right) \frac{lgn-2}{lgn} \frac{1}{lgn} \quad (24)$$

$$= \left(2\frac{lg^2n - lgn + 2}{(lgn-1)^2} + (lgn-2)\frac{lg^2n - lgn + 2}{lg^2n} \right) \frac{lgn-2}{lgn} \frac{1}{lgn}$$

$$\forall v_{white} \in V, \Psi_2(v_{white}) = \left(2\frac{lgn}{\frac{2lgn + (lgn-1)(lgn-2)}{lgn}} + (lgn-2)\frac{lgn}{lgn} \right) \frac{lgn}{lgn} \frac{1}{lgn} \quad (25)$$

$$= \left(2\frac{lg^2n}{2lgn + (lgn-1)(lgn-2)} + (lgn-2) \right) \frac{1}{lgn}$$

The all nodes 4-hop or more faraway to the selected node have the SMC values as follows ($v_i \in V$, v_i is 4-hop or more faraway):

$$\Psi_2(v_i) = \overbrace{\frac{lgn}{lgn} + \frac{lgn}{lgn} + \cdots + \frac{lgn}{lgn}}^{lgn} \frac{lgn}{lgn} \frac{1}{lgn} = 1 \quad (26)$$

In order to select next node to minimum dominating set, the SMC values should be compared and the node whose centrality value is maximum, should be selected. The first step is to compare grey and red nodes' SMC values:

$$\Psi_2(v_{grey})k_1 = \Psi_2(v_{red})$$
$$\left(\frac{lgn-1}{lg^2n - lgn + 2} \right) k_1 = \left(2\frac{lg^2n - lgn + 2}{(lgn-1)^2} + (lgn-2)\frac{lg^2n - lgn + 2}{lg^2n} \right) \frac{lgn-2}{lgn} \frac{1}{lgn} \quad (27)$$

where $k_1 = \frac{lg^4n(lgn-1)^2}{2lg^2n(lg^2n-lgn+2)(lgn-2)+(lgn-2)^2(lgn-1)^2(lg^2n-lgn+2)}$ and $\frac{1}{k_2} = \frac{2(lg^2n-lgn+2)(lgn-2)}{lg^2n(lgn-1)^2} + \frac{(lgn-2)^2(lg^2n-lgn+2)}{lg^4n}$ where $\frac{(lgn-2)^2(lg^2n-lgn+2)}{lg^4n} < 1$, $\frac{4(lgn-2)}{lgn(lgn-1)^2} < 1$ and $\frac{2(lgn-2)^2}{lg^2n(lgn-1)} < 1$.

$lim\, n \to \infty \frac{(lgn-2)^2(lg^2n-lgn+2)}{lg^4n} = 1$,

$lim\, n \to \infty \frac{4(lgn-2)}{lgn(lgn-1)^2} = 0$ and

$\lim n \to \infty \frac{2(lgn-2)^2}{lg^2 n(lgn-1)} = 0$ imply that $k_2 > 1$. That's why, $\Psi_2(v_{white}) > \Psi_2(v_{red})$ ■

NON-REDUNCDANCY OF MALATYA DOMINATING SET ALGORITHM

Malatya Dominating Set Algorithm (MDSA)is an efficient algorithm to find the minimum dominating sets for given graphs, since the space and time complexities of MDSA are polynomials. MDSA finds near-optimal solutions for some graphs, and due to this case, MDSA is an optimal or near-optimal algorithm. The requirement of revision in algorithm is needed. Fig.11 illustrates this case, i.e. red nodes are elements of minimum dominating set, and green nodes are elements of neighbors of dominating nodes. The red nodes 10, 20 can be removed from minimum dominating set and in this case, green node 15 should be added to minimum dominating set. Due to this case, "redundant node elimination" should be done in this perspective.

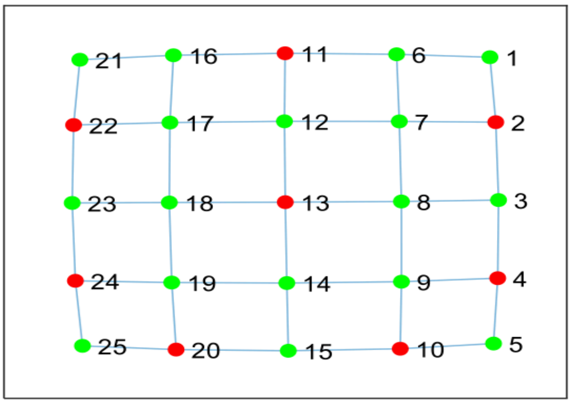


Figure 11. Near-optimal solution for grid 5x5 without Node Justification process.

Fig.11 illustrates the result of MDSA for grid graph of size 5x5. The nodes 10 and 20 are redundant nodes with respect to Definition 3. "Redundant node elimination" algorithm detects node 15, and in this case, black nodes 10 and 20 are converted to grey nodes and node 15 is converted to black node. This is the last step of MDSA and it eliminates redundant nodes from the minimum dominating set and adds new node(s) to minimum dominating set.

TIME AND SPACE COMPLEXITIES OF MDSA

The justification of the optimality of these algorithms were handled in this paper. The justification of these algorithms was handled in two ways: according to obtained solutions, time-space complexities. The time and space complexities were given in subsequent pages.

Theorem 7: *The algorithm 1 consumes the time and space in polynomial forms for given graph $G = (V, E)$ where $|V| = n$ and $|E| = m$.*

Proof: The graph data structures can be performed in two ways. The proof of theorem can be given in two ways: Matrix can be used to represent the given graph. In this case, usage space is only an $n \times n$ matrix, where $|V| = n$. Due to this case, the space complexity of Algorithm 1 is $P_{MC1}(n) = \Theta(n^2)$. The FMC computation requires traversing adjacency matrix once. That's why, the time complexity of Algorithm 2 is $T_{MC1}(n) = \Theta(n^2)$.If the graph is represented by linked list, then the space usage is for Algorithm 1 is $P_{MC1}(n) = \Theta(n.\Delta(G))$. So, the time consumption of Algorithm 1 is $O(\Delta(G). n)$ ■

Theorem 8: *The algorithm 2 consumes the time and space in polynomial forms for given graph G=(V,E) where |V|=n and |E|=m.*

Proof: The graph data structures can be performed in two ways. The proof of theorem can be given in two ways: Matrix can be used to represent the given graph. In this case, usage space is only an $n \times n$ matrix where |V|=n. Due to this case, the space complexity of Algorithm 2 is $P_{MC2}(n) = \Theta(n^2)$. The FMC computation requires traversing adjacency matrix once. That's why, the time complexity of Algorithm 2 is $T_{MC2}(n)=\Theta(n^2)$.If the graph is represented by linked list, then the space usage is for Algorithm 2 is $P_{MC2}(n) = \Theta(n.\Delta(G))$. So, the time consumption of Algorithm 2 is $T_{MC2}(n)=O(\Delta(G).n)$ ■

Theorem 9: *Assume that $G = (V, E)$ is a graph where $|V| = n$ and $|E| = m$. The Algorithm 3 has polynomial time and space complexities.*

Proof: $G = (V, E) = G_0 = (V_0, E_0)$, |V|=n, where |V|=n. In order to select a node to the minimum dominating set requires usage of Algorithm 1 and Algorithm 2 once. $\Psi_1(\dots)$ and $\Psi_2(\dots)$ are computed. After selection of one node to the minimum dominating set, given graph is re-formed. The same process is repeated. The complexity of DominatingSet algorithm can be phrased in two ways (adjacency matrix, linked list). $\gamma(G)$ is the cardinality of minimum dominating set, $T_S(\dots)$ is the time searching node for minimum dominating set.

Case 1: Adjacency matrix:

$$T_{DS}(n) = \sum_{i=0}^{\gamma(G)} \left(T_{MC1}(|V_i|) + T_{MC2}(|V_i|) + T_S(|V_i|)\right) = O(\gamma(G)n^2) \tag{28}$$

Case 2: Linked list:

$$T_{DS}(n) = \sum_{i=0}^{\gamma(G)} \begin{pmatrix} T_{MC1}(\Delta(G_i)|V_i|) \\ +T_{MC2}(\Delta(G_i)|V_i|) + T_S(|V_i|) \end{pmatrix} = O(\gamma(G).n.\Delta(G)) \tag{29}$$

So, the proof is completed ■

CONCLUSIONS

This study presents analytical verifications to strengthen the theoretical foundations of the Malatya Dominating Set Algorithm (MDSA). Previous studies have shown that MDSA has given successful results experimentally on various datasets and graph structures. However, these experimental successes needed to be mathematically proven and it needed to be proven that the algorithm produces optimal or near-optimal results on certain graph types. Accordingly, in our study, it has been mathematically shown that MDSA gives the best results with the least number of nodes on certain graph types and its theoretical framework has been strengthened.

One of the most important contributions of the study is the analytical modeling of the minimum dominating set determination process using the Malatya Centrality Value and proving that it can produce optimal solutions. Mathematical analyses performed especially on Path, Cycle, Star, and Bipartite Graph types have revealed that MDSA can create optimal minimum dominating sets for these graph types. In addition, it has been mathematically proven that the minimum dominating set can be created on regular graph types such as grid-based structures and hypercubes.

Another important feature of MDSA is that it can produce redundant minimum dominating sets. Unlike traditional greedy or heuristic methods, MDSA includes the Redundant Node Elimination mechanism. In our study, it is shown that this mechanism is mathematically verified and that the algorithm can optimize the minimum dominating set size by eliminating unnecessary nodes. This feature shows that MDSA is built on a theoretically stronger foundation compared to other minimum dominating set algorithms.

Analysis in terms of time and space complexity proves that MDSA has polynomial time and memory requirements and offers an efficient approach to solve the dominating set problem. The Minimum Dominating Set (MDS) problem, which is NP-hard, is generally addressed in the literature with heuristic or ILP-based approaches. However, MDSA offers significant advantages over existing methods in the literature since it is both more efficient in terms of time and provides optimal solutions for certain graph types.

In conclusion, this study mathematically proves that MDSA produces analytically optimal results for certain types of graphs and presents a new theoretical framework for minimum dominating set problems. Specifically, the non-redundant minimum dominating set creation mechanism's analytical demonstration adds significantly to the body of research. MDSA should be supported by more comprehensive analyses in the future and made usable in different applications due to its ability to determine minimum dominating sets and produce optimal solutions in polynomial time complexity. Therefore, our study provides a theoretical basis for new research on minimum dominating set problems and takes an important step towards increasing the efficiency of centrality-based algorithms.

REFERENCES

[1] A. Mitra and S. Paul, "Analyzing social networks with dynamic graphs: Unravelling the ever-evolving connections," *Appl. Graph Data Sci.*, pp. 195–214, Jan. 2025, doi: 10.1016/B978-0-443-29654-3.00020-X.

[2] L. Yu, Z. Wang, Y. Liu, and Z. Yang, "Sampled-Data-Based Privacy-Preserving Scaled Consensus for Nonlinear Multiagent Systems: A Paillier Encryption Approach," *IEEE Trans. Syst. Man, Cybern. Syst.*, 2024, doi: 10.1109/TSMC.2024.3517473.

[3] R. Das and M. Soylu, "A key review on graph data science: The power of graphs in scientific studies," *Chemom. Intell. Lab. Syst.*, vol. 240, p. 104896, Sep. 2023, doi: 10.1016/J.CHEMOLAB.2023.104896.

[4] L. D. F. Costa *et al.*, "Analyzing and modeling real-world phenomena with complex networks: a survey of applications," *Adv. Phys.*, vol. 60, no. 3, pp. 329–412, May 2011, doi: 10.1080/00018732.2011.572452.

[5] E. Gayathiri *et al.*, "Computational approaches for modeling and structural design of biological systems: A comprehensive review," *Prog. Biophys. Mol. Biol.*, vol. 185, pp. 17–32, Dec. 2023, doi: 10.1016/J.PBIOMOLBIO.2023.08.002.

[6] S. S. Singh, S. Muhuri, S. Mishra, D. Srivastava, H. K. Shakya, and N. Kumar, "Social Network Analysis: A Survey on Process, Tools, and Application," *ACM Comput. Surv.*, vol. 56, no. 8, Apr. 2024, doi: 10.1145/3648470.

[7] D. Karthika, R. Muthucumaraswamy, M. Bentert, S. Bhyravarapu, S. Saurabh, and S. Seetharaman, "On the Complexity of Minimum Membership Dominating Set," pp. 94–107, 2025, doi: 10.1007/978-3-031-82670-2_8.

[8] M. Shahraeini, G. Kohsari, and M. H. Javidi, "Comparison of Meta-Heuristic Algorithms for Solving Dominating Set Problems in WAMS Design," *Proc. - 2022 8th Int. Iran. Conf. Signal Process. Intell. Syst. ICSPIS 2022*, 2022, doi: 10.1109/ICSPIS56952.2022.10044037.

[9] D. L. L. Moura, A. L. L. Aquino, and A. A. F. Loureiro, "Towards data VSN offloading in VANETs integrated into the cellular network," *MSWiM 2019 - Proc. 22nd Int. ACM Conf. Model. Anal. Simul. Wirel. Mob. Syst.*, pp. 235–239, Nov. 2019, doi: 10.1145/3345768.3355937.

[10] M. M. Daliri Khomami, A. Rezvanian, N. Bagherpour, and M. R. Meybodi, "Minimum positive influence dominating set and its application in influence maximization: a learning automata approach," *Appl. Intell.*, vol. 48, no. 3, pp. 570–593, Mar. 2018, doi: 10.1007/S10489-017-0987-Z/FIGURES/7.

[11] R. Sun, J. Wu, C. Jin, Y. Wang, W. Zhou, and M. Yin, "An efficient local search algorithm for minimum positive influence dominating set problem," *Comput. Oper. Res.*, vol. 154, p. 106197, Jun. 2023, doi: 10.1016/J.COR.2023.106197.

[12] M. A. Akbay, A. López Serrano, and C. Blum, "A Self-Adaptive Variant of CMSA: Application to the Minimum Positive Influence Dominating Set

Problem," *Int. J. Comput. Intell. Syst.*, vol. 15, no. 1, pp. 1–13, Dec. 2022, doi: 10.1007/S44196-022-00098-1/TABLES/4.

[13] J. Raczek and M. Miotk, "Two Approaches to Constructing Certified Dominating Sets in Social Networks," *IEEE Access*, 2025, doi: 10.1109/ACCESS.2025.3532392.

[14] S. Wuchty, "Controllability in protein interaction networks," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 111, no. 19, pp. 7156–7160, May 2014, doi: 10.1073/PNAS.1311231111/SUPPL_FILE/PNAS.201311231SI.PDF.

[15] S. K. Grady, K. A. Peterson, S. A. Murray, E. J. Baker, M. A. Langston, and E. J. Chesler, "A graph theoretical approach to experimental prioritization in genome-scale investigations," *Mamm. Genome*, vol. 35, no. 4, pp. 724–733, Dec. 2024, doi: 10.1007/S00335-024-10066-Z/FIGURES/7.

[16] Y. Tokuhara, T. Akutsu, J. M. Schwartz, and J. C. Nacher, "A practically efficient algorithm for identifying critical control proteins in directed probabilistic biological networks," *npj Syst. Biol. Appl. 2024 101*, vol. 10, no. 1, pp. 1–12, Aug. 2024, doi: 10.1038/s41540-024-00411-y.

[17] J. M. Etancelin, A. Fabbri, F. Guinand, and M. Rosalie, "DACYCLEM: A decentralized algorithm for maximizing coverage and lifetime in a mobile wireless sensor network," *Ad Hoc Networks*, vol. 87, pp. 174–187, May 2019, doi: 10.1016/J.ADHOC.2018.12.008.

[18] N. R. Sivakumar, S. M. Nagarajan, G. G. Devarajan, L. Pullagura, and R. P. Mahapatra, "Enhancing network lifespan in wireless sensor networks using deep learning based Graph Neural Network," *Phys. Commun.*, vol. 59, p. 102076, Aug. 2023, doi: 10.1016/J.PHYCOM.2023.102076.

[19] Y. Sabri and A. Hilmani, "An IoT-Based 5G Wireless Sensor Network Employs a Secure Routing Methodology Leveraging DCNN Processing," *Trans. Emerg. Telecommun. Technol.*, vol. 35, no. 12, p. e70025, Dec. 2024, doi: 10.1002/ett.70025.

[20] S. Saurabh, S. Madria, A. Mondal, A. S. Sairam, and S. Mishra, "An analytical model for information gathering and propagation in social networks using random graphs," *Data Knowl. Eng.*, vol. 129, p. 101852, Sep. 2020, doi: 10.1016/J.DATAK.2020.101852.

[21] F. Okumuş and Ş. Karcı, "MDSA: A Dynamic and Greedy Approach to Solve the Minimum Dominating Set Problem," *Appl. Sci. 2024, Vol. 14, Page 9251*, vol. 14, no. 20, p. 9251, Oct. 2024, doi: 10.3390/APP14209251.

[22] A. Science, S. Yakut, F. Öztemiz, and A. Karcı, "A New Approach Based on Centrality Value in Solving the Maximum Independent Set Problem: Malatya Centrality Algorithm," *Comput. Sci.*, vol. Vol:8, no. Issue:1, pp. 16–23, Jun. 2023, doi: 10.53070/BBD.1224520.

[23] A. Science and F. Öztemiz, "İki Parçalı Eşleştirme ile Maksimum Akış," *Bilgi. Bilim.*, vol. Vol:8, no. Issue:2, pp. 102–109, Dec. 2023, doi: 10.53070/BBD.1386446.

[24] H. Jiang and Z. Zheng, "An Exact Algorithm for the Minimum Dominating Set Problem," 2023.