Journal of Computer Science
https://dergipark.org.tr/en/pub/bbd
Anatolian Science

ISSN,e-ISSN: 2548-1304
Volume:8 , Issue:1 , pp:16-23, 2023
https://doi.org/10.53070/bbd.1224520
Research Paper

A New Approach Based on Centrality Value in Solving the Maximum Independent Set Problem: Malatya Centrality Algorithm

Selman Yakut [1], Furkan Öztemiz [1], Ali Karcı [1]

[1]Department of Software Engineering, İnönü University, Malatya, Turkey

(selman.yakut@inonu.edu.tr, furkan.oztemiz@inonu.edu.tr, ali.karci@inonu.edu.tr)

Received:Dec.26,2022 Accepted:Jan.25,2023 Published:Jun.08,2023

Abstract— Graph structure is widely used to describe problems in different fields. Problems in many areas, such as security and transportation, are among them. The problems can be solved using approaches similar to the graph structure. The independent set problem, which is NP-complete problem, is one of the main problems of graph theory and is used in modeling many problems. The implementation of the independent set problem with the most possible number of nodes in the graph is called Maximum Independent Set Problem. A lot of algorithm approach are proposed to solve the problem. This study proposes an effective approach for the maximum independent problem. This approach occurs two steps: computing Malatya centrality value and determining the maximum independent set. In the first step, centrality values are computed for the nodes forming the graph structure using Malatya centrality algorithm. Malatya centrality value of the nodes in any graph is the sum of the ratios of the node's degree to the neighboring nodes' degrees. The second step is to determine the nodes to be selected for the maximum independent set problem. Here, the node with the minimum Malatya centrality value is selected and added to the independent set. Then, the edges of this node, its adjacent nodes, and the edges of adjacent nodes are subtracted from the graph. By repeating the new graph structure calculations, all vertexes are deleted so that the maximum independent set is determined. It is observed on the sample graph that the proposed approach provides an efficient solution for the maximum independent set. Successful test results and analyzes demonstrate the effectiveness of the proposed approach.

Key Words : *Maximum independent set, Graph theory, Malatya centrality algorithm, Malatya centrality value.*

1. Introduction

The graph is a basic mathematical model and a data structure widely used in computer science (Cormen, Leiserson, Rivest, & Clifford, 2001). Graph structure models problems in many engineering fields, such as transportation, security, computer networks, bioinformatics, and chemistry. Thus, algorithms and solutions developed for the graph are expected to provide solutions to these problems (Uçkan & Karcı, 2020). However, many NP problems cannot be solved in polynomial time defined in the graph structure. The independent set problem is one of these problems.

The Independent set problem is one of the NP-complete problems whose solution is studied in graphs (Thulasiraman & Swamy NS, 2011). The independent set problem is determining the nodes in a graph that do not have a neighborhood. For this process, different subsets of nodes in the graph can be determined. However, determining the maximum number of nodes found for the whole graph structure is called the Maximum Independent Set Problem (MISP). MISP is an NP-hard optimization problem that is difficult to solve in polynomial time. However, many real-life problems can be modeled and solved based on MISP (Li, He, Xu, & Wang, 2020). Approaches such as text summarizations (Uçkan & Karcı, 2020) and detecting fraudulent nodes in duplicate voting pools (Araujo, Farinha, Domingues, Silaghi, & Kondo, 2011) are examples of these problems. Therefore, the algorithms developed for MVCP and the solutions found can also be used for real-life problems. Many algorithms are proposed, and approaches are developed to solve this problem.

Since MISP is an NP-complete optimization problem, no method provides a complete solution with polynomial time complexity. Algorithms that provide complete solutions either have exponential time complexity or provide solutions for some special graphs. Finding solutions close to the optimum solution set using heuristic and metaheuristic approaches is also a complex problem. However, these solutions are obtained under certain constraints. There are different approaches and algorithms for MISP solutions in the literature. However, these approaches and algorithms are divided into two classes. These are the exact approaches that provide a complete solution; the other is heuristic approaches that enable finding solutions close to the optimum solution set.

Exact approaches are approaches that offer solutions for a specific graph (for example, regular graphs) structure and with certain constraints. The exact approach proposed by Xiao and Nagommochi produces successful results for graphs with a certain degree (Xiao & Nagamochi, 2017). The approach proposed by Karcı is based on extracting the basic cut-sets values using special expansion trees (Karcı, 2020). Brandstadt and Mosca used the dynamic programming approach and solved it in polynomial time of the maximum weight-independent solution set for clawless graphs (Brandstädt & Mosca, 2018). By proposing two dynamic programming approaches, Wan et al. tried to solve the problem of independent sets and matches in given dimensions in graphs of order n and tree width at most p (Wan, Tu, Zhang, & Li, 2018). Lamm et al. have developed an exact solution for MISP with a branch-and-reduce-based approach that can also yield results for large graphs (Lamm, Schulz, Strash, Williger, & Zhang, 2019).

Heuristic approaches have been widely used to determine the maximum possible solution set for MISP. In the proposed deterministic greedy approach, an effective approach has been proposed for some graph types (Ballard-myer, 2019). The efficiency of the Karcı algorithm is given in some special graphs, and the efficiency of the algorithm has been verified (Karcı, 2022). Großmann et al. used a memetic algorithm to obtain near-optimal results for MISP in complex and large graphs where data reduction is not possible (Großmann, Lamm, Schulz, & Strash, 2022). The approach proposed by Alkhouri et al. is presented in large graphs using artificial neural networks without data (Alkhouri, Atia, & Velasquez, 2022). The distributed greedy approach was used to determine the maximum weighted independent set for fading channels in wireless networks (Joo, Lin, Ryu, & Shroff, 2016). Das et al. proposed a 2-approach that provides a solution in polynomial time for the maximum independent set problem for a unit disk graph (Das, De, Kolay, Nandy, & Sur-Kolay, 2015).

In this study, an approach that offers an effective solution for MISP is proposed. This approach calculates Malatya centrality value using Malatya centrality algorithm. While computing Malatya centrality value of the nodes, the node's degree and neighboring node degrees are used together. For each vertex in the graph, Malatya centrality value is produced by summing the ratios of its degree with the neighboring vertex degrees. Determining the nodes to be selected for MISP consists of two steps. First, the node with the lowest calculated Malatya centrality values is selected, and the related vertex, edges, and adjacent nodes are removed from the graph. Then, Malatya centrality value is recalculated for the remaining graph structure, and the selection processes are repeated. When the process is completed, the solution set obtained becomes the maximum independent set for MISP.

The rest of the article is organized as follows. Chapter 2 contains preliminary information about the proposed approach with preliminaries. In Chapter 3, the proposed method is discussed. Here, the calculation of the Malatya centrality value and the determination of the solution set for MISP was examined. Chapter 4 gives evaluations of the proposed algorithm and applications on sample graphs. Finally, in the conclusion part, the results related to the proposed algorithm were mentioned.

2. Preliminaries

With the proposed approach, a solution has been developed for the MISP solution by using the Malatya centrality value and vertexes. In order to understand this approach, it is necessary to examine and understand these concepts. These concepts are discussed later in the article.

2.1. Centrality

Centrality is expressed as the assignment of values depending on the positions of vertexes in the graph (Borgatti, 2005). Centrality is widely used in many fields, especially in graph theory. These areas and the application used are intended to determine the effective node in the graph or application. Numerous approaches and algorithms have been proposed to identify central nodes. Of these approaches, vertex connections are one of the decisive parameters for measuring the centrality of the graph. This approach is expressed as degree centrality and is included in the structure of many algorithms, such as PageRank, which is widely used (Kumar, Duhan, & Sharma, 2011).

2.2. Maximum Independent Set Problem

The independent set problem consists of nodes in the graph structure that do not have an adjacent edge. There may be more than one set of solutions that meet these criteria. However, determining the independent solution set using the maximum number of vertexes is expressed as MISP. Determining this set is a strongly NP-hard optimization problem, and it is difficult to determine the optimum solution sets or solutions close to the optimum solution. Many approaches are proposed to solve the problem, including exact approaches and optimization methods.

3. Proposed Malatya Independent Set Algorithm

The proposed approach to effectively solve MISP consists of two steps. These are calculating Malatya centrality values and determining the nodes to be selected to solve the independent set problem. First, Malatya centrality values are calculated for all nodes in the graph. For this calculation, the node's degree and the degrees of its neighboring nodes are used. Malatya centrality value is the sum of the values obtained by dividing the node's degree by the degree of each neighboring node (Karcı, Yakut, & Öztemiz, 2022).

The general outlines of the proposed approach are given in Figure 1. In the figure, firstly, the edge and node data of the sample graph are taken, and Malatya centrality values of this graph are calculated. Then, to determine the independent set, the node with the minimum Malatya centrality value is choose and included in the independent set. With this selected node, the edges and vertexes coincident with this node are removed from the graph. For the new graph structure formed, Malatya centrality algorithm is applied again, and the vertex selection process is continued. This calculation and subtraction are continued until the nodes in the graph are exhausted.

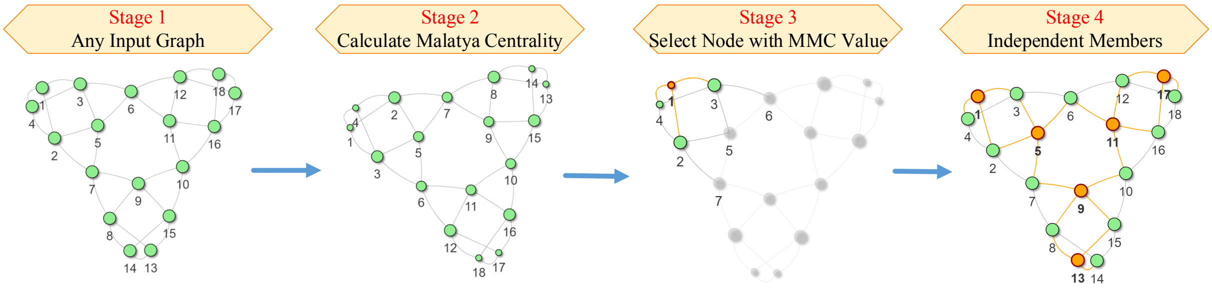

Figure 1. General structure of the proposed algorithm

The approach to MISP consists of the steps given below.

1 Malatya centrality value of the nodes in the graph is calculated using Malatya centrality algorithm. This value is the distinguishable node number and is denoted by ψ(v) for a node v.

2 The node with the minimum ψ(v) value is selected.

3 Along with the selected node, vertexes coincide with this node, and their connections are extracted from the graph.

4 The algorithm is complete when all nodes are deleted; otherwise, it is returned to the first step with the new graph structure.

The algorithm given in Equation 1 is used to calculate Malatya centrality algorithm node ψ(v) values. In this equation, n is the number of nodes in the graph; $d(v_i)$ denotes the degree of nodes, and $N(v_i)$ the set of adjacent nodes.

$$\psi(v_i) = \sum_{\forall v_j \in N(v_i)} \frac{d(v_i)}{d(v_j)} \tag{1}$$

The pseudocode Algorithm 1 of the proposed approach for MISP is given. The given codes contain the operations of the proposed algorithm. Lines 1-10 contain the codes of Malatya centrality algorithm. In rows 11-21, the vertexes to be selected are determined and deleted from the graph together with the edges. Then, the nodes adjacent to this vertex and their incident edges are extracted from the graph. When all vertexes in the graph are deleted, the processes are terminated. Otherwise, the centrality value is recalculated for the new graph, and the operations are continued. Next to this piece of code, the descriptions give explanations about the codes.

Algorithm 1. Proposed Algorithm Pseudocode

Proposed Algorithm

```
G:(V,E)                                                        // G graph
MalatyaCentralityMethod <- function(g){                        // Malatya Algorithm is defined
   VertexList <- c(V(g))                                       // Throw vertex from graph to array
   for (i in VertexList)                                       // Work as many vertexes in the array
       Vdegree <-degree(g,v = V(g)[i])                         // Calculate the node degree of the corresponding vertex
      AdjacentDegree <- degree(g,v = neighbors(g,v = V(g)[i])) // Calculate the node degree of the neighbors of the relevant node
       Value <- Vdegree/AdjacentDegree                         // Degree of related node / degree of adjacent node
       MalatyaCentralityValue <- print(paste(V(g)[i],sum(Value)),digits = 3)  // New centrality value results
   return(MalatyaCentralityValue)                              // Returns Malatya Centrality Value
}
FindMinMalatyaCentralityValue <-function(g){                   // Method that returns vertex name with minimum centrality
       minVertex <- FindMinVertex(MalatyaCentralityMethod(g)); // Calculates the minimum vertex degree
       V <- minVertex;
        neighborsVertex <- neighbors(g,v = minVertex)
        DeleteEdges(minVertex);                                // The edges of the selected minimum vertex are deleted
        DeleteEdges(neighborsVertex);                          // The connections of the neighbors of the minimum vertex are deleted
    return (V);         }
FindMaxIndependentSet <- function(g){                          // Detects maximum independent members
       while(Edge.Count != 0)                                  // It works as long as there is an unreached edge in the graph.
               FindMinMalatyaCentralityValue(g);
                       print(minVertex);                       // Print the maximum independent members
}
```

The operations of the proposed approach in Algorithm 2 are given in mathematical expressions. Here, Malatya centrality values are calculated using Malatya centrality algorithm, and the maximum independent set is determined. In the given approach, the solution set is initially empty. Then Malatya centrality values are calculated and added to the independent set. When all the nodes in the graph are deleted, the solution set is obtained. $\psi(v_i)$ used in this approach is Malatya centrality value, V_c solution set, $d(v_i)$, v_i. the degree of the node, $|V|$ shows the number of nodes in the graph.

Algorithm 2. Mathematical representation of Malatya Algorithm

Mathematical Representation of Proposed Algorithm

Input: Adjacency matrix of G is A and G=(V,E) // G graph

Output: $V_c \subseteq V$, V_c is a set of nodes and it is a solution for independent set problem

$V_c \leftarrow \varnothing$

While $E \neq \varnothing$ do

  $i \leftarrow 1, \ldots, |V|$

    $\psi(v_i) = \sum_{\forall v_j \in N(v_i)} \frac{d(v_i)}{d(v_j)}$

  $V_c = V_c \cup \{\min(\psi(v_i))\}$

  $V = V - \{v_i\}$, and $E = E - \forall (v_i, v_j) \in E$

Output=V_c

4. Experimental Results

An essential advantage of the proposed approach is that it provides a robust solution in polynomial time for the NP-complete problem MISP. In general, the heuristic solutions proposed for MISP yield near-optimal results under certain constraints. In the exact approaches recommended for MISP, it offers solutions in some special graphs and under certain constraints. Furthermore, the proposed approach includes a new structure that can be applied to graphs without restrictions. In order to determine the effectiveness of the proposed approach, the results obtained on the graphs should be examined. Therefore, the proposed approach in this study was run step by step on the sample graphs, and the results were shown in detail. This graph used Malatya centrality algorithm for MISP, and the independent set containing the minimum number of nodes was determined.

In Figure 2, the details of the sample graph used to show the effectiveness of Malatya centrality algorithm are given. The nodes and edges of the graph are detailed in the figure. In the graph, the connections of each node and Malatya centrality values of all nodes are given in Table 1. This graph calculates Malatya centrality value using Malatya centrality algorithm, and a suitable graph model is presented for determining inclusive nodes.

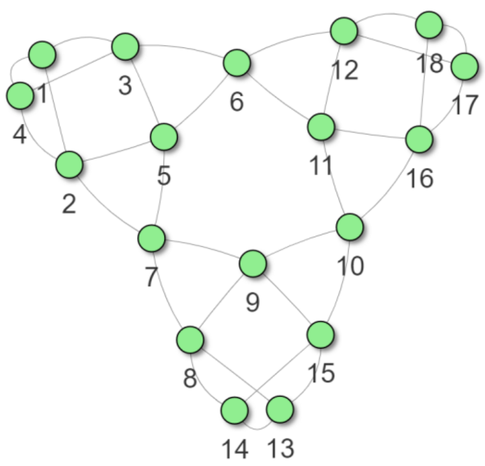

Figure 2. Graph model used in calculation

Table 1. The graph's Malatya Centrality Values

Vertex Name	Centrality Value	Vertex Name	Centrality Value
1	2.500000	10	4.000000
2	4.666667	11	4.000000
3	4.666667	12	4.666667
4	2.500000	13	2.500000
5	4.000000	14	2.500000
6	4.000000	15	4.666667
7	4.000000	16	4.666667
8	4.666667	17	2.500000
9	4.000000	18	2.500000

The sample graph structure gives Malatya centrality values calculated using the proposed method in Figure 3. In this graph, nodes with high Malatya centrality values are shown with a larger circle, while those with low centrality are shown with a small circle. For example, when Table 1. is examined, with Malatya centrality value of 2.5, node 1, 4, 13, 14, 17, and 18 has the smallest centrality values. In the case of equality, choosing any of these nodes is sufficient. In this example, node 1 is selected by selecting the index order.

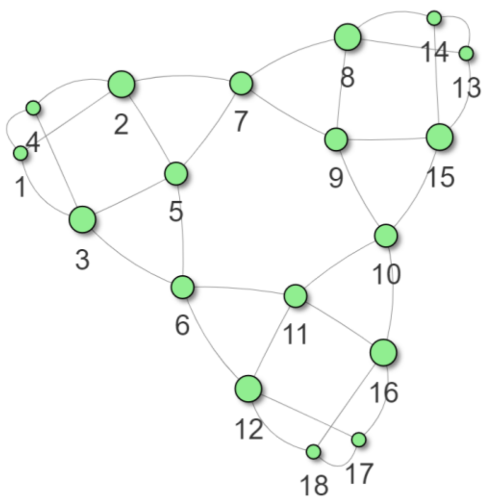

Figure 3. Graph structure according to Malatya centrality value

Using the algorithm proposed for the independent set problem in Figure 4, node 1 was selected as the first member of the independent set, and the related graph structure was given. After the selection process performed using the algorithm, we suggested node 1 and node 1 are deleted from the graph with their adjacent node 2, 3, and 4 edge connections. Then, the graph is updated and prepared for selecting the second independent set member.

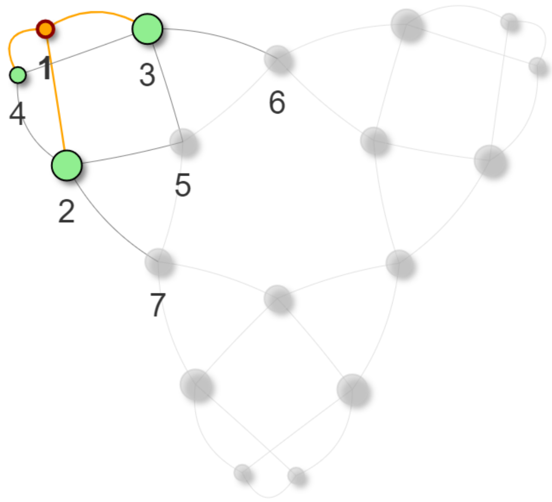

Figure 4. Example structure where the most ineffective node is determined

The proposed algorithm continues to run until no nodes are left in the graph. The graph structure formed after selecting the last independent set member is given in Figure 5. Nodes marked in orange in the figure represent independent set members. The solution set produced by the algorithm we presented for the dependent set problem consists of nodes 1, 5, 9, 11, 13, and 17. For the example graph consisting of 18 nodes and 33 edge relations, the number of elements of the independent set is determined as 6. This value also represents the maximum independent set value for this graph.

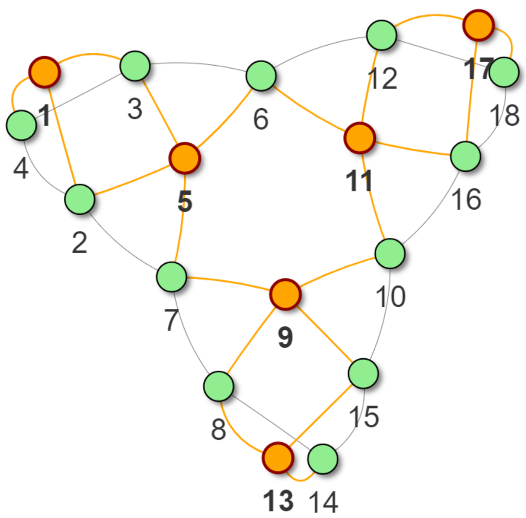

Figure 5. Vertex set used for maximum independent set

5. Conclusion

This study presents a robust algorithm for MISP, one of the crucial problems of graph theory. This algorithm is a polynomial algorithm, unlike solutions from the literature, and offers an effective solution. In the proposed algorithm, Malatya centrality value of the nodes is calculated using Malatya centrality algorithm. While calculating Malatya centrality values, the node's degree is used from the degree of its neighboring nodes. Then, independent set members are selected by prioritizing the minor node from Malatya centrality values obtained. The presented method is an effective method for the selection of independent set members. Although it gives near-optimal results, it detects the maximum independent set members in many graph types. The efficiency of the proposed approach and Malatya centrality algorithm for MISP has been tested on sample graphs. The successful test results and analysis demonstrated the effectiveness of the proposed Malatya centrality algorithm and the independent set solutions.

References

Alkhouri, I. R., Atia, G. K., & Velasquez, A. (2022). A differentiable approach to the maximum independent set problem using dataless neural networks. *Neural Networks*, 155, 168–176. Retrieved from https://doi.org/10.1016/j.neunet.2022.08.008

Araujo, F., Farinha, J., Domingues, P., Silaghi, G. C., & Kondo, D. (2011). A maximum independent set approach for collusion detection in voting pools. *Journal of Parallel and Distributed Computing*, 71(10), 1356–1366. Retrieved from https://doi.org/10.1016/j.jpdc.2011.06.004

Ballard-myer, J. C. (2019). Deterministic Greedy Algorithm for Maximum Independent Set Problem in Graph Theory, 1–14.

Borgatti, S. P. (2005). Centrality and network flow. *Social Networks*, 27(1), 55–71. Retrieved from https://doi.org/10.1016/j.socnet.2004.11.008

Brandstädt, A., & Mosca, R. (2018). Maximum weight independent set for ℓclaw-free graphs in polynomial time *Discrete Applied Mathematics*, 237, 57–64. Retrieved from https://doi.org/10.1016/j.dam.2017.11.029

Cormen, T. H., Leiserson, C. E., Rivest, R., & Clifford, S. (2001). *Introduction to algorithms* (Introducti). London.

Das, G. K., De, M., Kolay, S., Nandy, S. C., & Sur-Kolay, S. (2015). Approximation algorithms for maximum independent set of a unit disk graph. *Information Processing Letters*, 115(3), 439–446. Retrieved from https://doi.org/10.1016/j.ipl.2014.11.002

Großmann, E., Lamm, S., Schulz, C., & Strash, D. (2022). Finding Near-Optimal Weight Independent Sets at Scale. Retrieved from http://arxiv.org/abs/2208.13645

Joo, C., Lin, X., Ryu, J., & Shroff, N. B. (2016). Distributed Greedy Approximation to Maximum Weighted Independent Set for Scheduling With Fading Channels. *IEEE/ACM Transactions on Networking*, 24(3), 1476–1488. Retrieved from https://doi.org/10.1109/TNET.2015.2417861

Karci, A. (2020). Efficient Algorithms for Determining the Maximum Independent Sets in Graphs. *Computer Science*, 5(2), 144–149.

Karci, A. (2022). Verification of Karci Algorithm's Efficiency for Maximum Independent Set Problem in Graph Theory. *Computer Science*, 20–28. Retrieved from https://doi.org/10.53070/bbd.1090368

Karcı, A., Yakut, S., & Öztemiz, F. (2022). A New Approach Based on Centrality Value in Solving the Minimum Vertex Cover Problem: Malatya Centrality Algorithm. *Computer Science*, 55(35), 1–100. Retrieved from https://doi.org/10.53070/bbd.1195501

Kumar, G., Duhan, N., & Sharma, A. K. (2011). Page ranking based on number of visits of links of Web page. In *2011 2nd International Conference on Computer and Communication Technology (ICCCT-2011)* (pp. 11–14). IEEE. Retrieved from https://doi.org/10.1109/ICCCT.2011.6075206

Lamm, S., Schulz, C., Strash, D., Williger, R., & Zhang, H. (2019). Exactly Solving the Maximum Weight Independent Set Problem on Large Real-World Graphs. In *2019 Proceedings of the Twenty-First Workshop on Algorithm Engineering and Experiments (ALENEX)* (Vol. January, pp. 144–158). Philadelphia, PA: Society for Industrial and Applied Mathematics. Retrieved from https://doi.org/10.1137/1.9781611975499.12

Li, Z., He, G., Xu, D., & Wang, S. (2020). Evaluation of Centralized Reader Anti-Collision Protocols for Mobile RFID System Based on Maximum Independent Set: A Simulation Study. *IEEE Access*, 8, 123381–123397.

Retrieved from https://doi.org/10.1109/ACCESS.2020.3006162

Thulasiraman, K., & Swamy NS, M. (2011). *Graphs: theory and algorithms*. Montreal: John Wiley & Sons.

Uçkan, T., & Karcı, A. (2020). Extractive multi-document text summarization based on graph independent sets. *Egyptian Informatics Journal*, 21(3), 145–157. Retrieved from https://doi.org/10.1016/j.eij.2019.12.002

Wan, P., Tu, J., Zhang, S., & Li, B. (2018). Computing the numbers of independent sets and matchings of all sizes for graphs with bounded treewidth. *Applied Mathematics and Computation*, 332, 42–47. Retrieved from https://doi.org/10.1016/j.amc.2018.03.017

Xiao, M., & Nagamochi, H. (2017). Exact algorithms for maximum independent set. *Information and Computation*, 255, 126–146. Retrieved from https://doi.org/10.1016/j.ic.2017.06.001