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

ISSN,e-ISSN: 2548-1304
Volume:10, Issue:2, pp:179-185, 2025
https://doi.org/**10.53070/bbd.1776643**
Research Paper

The Malatya Independent Set Algorithm is Optimum

Ali KARCI
Software Engineering Department, İnönü University, Malatya / Türkiye
{ali.karci@inonu.edu.tr}

Received:Sep.01,2025 Accepted:Sep.11,2025 Published:Dec.01,2025

Abstract— Malatya Independent Set Algorithm is a greedy based algorithm, since it uses the first Malatya Centrality algorithm to find the maximum independent set for given graph. The scope of this paper is to illustrate that Malatya Independent Set Algorithm is optimum, and its process takes place for removing minimum nodes from graph as possible as for each node selection step.

Keywords : *Malatya Centrality Algorithm, Malatya Independent Set Algorithm, Maximum Independent Set.*

1.Introduction

Assume that G=(V,E) where |V|=n, |E|=m. The independent set is a subset of V whose neighbours are not included in the independent set. In other words, assume $D \subseteq V$, and $\forall u,v \in D$, $(u,v) \notin E$, then D is an independent set. The set D with maximum cardinality is called the maximum independent set (MIS).

The power/strength of a node can be considered inversely proportional to the power/strength of its neighbours. If a node's neighbour is stronger than itself, that node is naturally considered the weaker node in that region. Therefore, the number of neighbours of a node has been considered an indicator of that node's power/strength, resulting in the first Malatya centrality definition. This definition and its next definition (the second Malatya centrality) were used to solve many NP-complete problems such as Maximum Independent Set, Minimum Dominating Set, Minimum Vertex Cover Set, Graph Colouring.

Yakut and his friends (Karcı et al, 2022; Yakut et al, 2023a) used the first Malatya centrality definition for solving the minimum vertex-cover set problem. Their algorithm has polynomial time and space complexities. The implementation of the designed algorithm is easy.

Yakut and his friends (Yakut et al, 2023b) used the first Malatya centrality definition for solving the maximum independent set problem. Their algorithm has polynomial time and space complexities. The implementation of the designed algorithm is easy. Yakut (Yakut, 2025) proposed a new method for solving graph colouring based on the first Malatya centrality algorithm.

Öztemiz and Yakut (Öztemiz and Yakut, 2024a) designed an algorithm to determine the dominance of each node in the given graph based on the Malatya centrality approach. The experimental results of the designed algorithm for familiar networks in were given in this study. Öztemiz and Yakut (Öztemiz and Yakut, 2024b) proposed a new method to solve the clique problem and they have also illustrated the efficiency of the proposed method by given experimental results.

Öztemiz (Öztemiz, 2025a) proposed a new algorithm to solve the edge colouring problem in the given graph. Öztemiz (Öztemiz, 2025b) redefined the first Malatya centrality, and designed a new algorithm to solve the Maximum Independent Set problem. He called this algorithm as Differential Malatya Independent Set Algorithm. The analytical verification of algorithm was given in this paper, and the superiority of the designed / proposed algorithm with respect to algorithms in the literature was illustrated by given the experimental results for many data set.

Bakan and Yakut (Bakan and Yakut, 2023) proposed a method to summarize the text based on the Malatya centrality approach. They illustrated the efficiency of the proposed method by given the experimental results.

Okumuş and Karci (Okumuş and Karci, 2024) proposed a new algorithm to solve the minimum dominating set problem, and they supported their proposed efficiency by experimental results based on the Malatya centrality approach. Karci and her friends (Karci et al, 2023) proposed a new method to determine the dominance of each node in the given graph based on the second Malatya centrality method. Karci and Okumuş (Karci and Okumuş, 2023) verified the analytical efficiency of the Malatya Dominating Set Algorithm (MDSA).

The aim of this paper is to verify the optimality of MISA analytically. This is the motivation of this study, and the organization of this paper is as follow. The second section includes a brief description of MISA, The analytical verification of MISA is given in the third section.

2. Malatya Independent Set Algorithm (MISA)

The Malatya centralities were defined to solve NP-Complete problems such as Maximum Clique, Maximum Independent Set, Minimum Dominating Set, Minimum Connected Dominating Set, Minimum Vertex-Cover Set, etc. problems. There is more than algorithm which were designed, to solve these problems. MISA is one of these algorithms to solve the maximum independent set problem.

The first Malatya centrality value is calculated by dividing a node's degree by its neighbour's degree, resulting in a ratio. This process is performed for all neighbour nodes, and these ratios are then summed to calculate the centrality of the corresponding node. The resulting sum is divided by the node's degree to obtain the first Malatya centrality value. On other words, the sum of the fractions in the form of one over the degree of each neighbour (fraction) will give the first Malatya centrality value of the corresponding node.

MISA was designed to the maximum independent set problem, and the MISA step can be given as follow. Initially, the adjacency matrix of graph is constructed from a text file where the list of edges of graph are in this file. Then the first Malatya centrality value is computed for each node, and a node with minimum centrality value is selected to MIS. This process carries on until each node is MIS or MIS neighbours list (Algorithm 1, Algorithm 2, Algorithm 3 and Algorithm 4).

Algorithm 1. IndependentSet

```
A←GetAdjacencyMatrix('Bipartite30x20.txt')
Terminated←0
//D=Degree matrix, MC1=first Malatya centrality,
// IS=independent set
while Terminated=0
   FirstMC(A,D,MC1)
   NodeSelection(A,D,NodeColour,MC1,IS)
   AddIsolatedNode(D,Nodecolour,IS)
   Termination(A,NodeColour,Terminated)
```

Algorithm 2. FirstMC(A,D,MC1)

```
i←1,2,…,n
   j←1,2,…,n
      if (A(i,j)=1) && (D(j)≠0)
            MC1(i)←MC1(i)+D(i)/D(j)
      MC1(i)←MC1(i)/D(i)
```

Algorithm 3.NodeSelection(A,D,NodeColour,MC1,IS)

```
SelectedNode←arg max(MC1)
i←1,2,…,n
   if (MC1(i)<MC1(SelectedNode)) and (MC1(i)≠0)
         SelectedNode←i
  IS←IS∪{SelectedNode},   NodeColour(SelectedNode)←2;
  i←1,2,…,n
     if A(SelectedNode,i)=1
        j←1,2,…,n
           if (A(i,j)=1)
              A(i,j)←0    A(j,i)←0
        A(SelectedNode,i)←0, A(i,SelectedNode)←0
        NodeColour(i)←1
```

```
Algorithm 4. GetAdjacencyMatrix(FileName)
File←fopen(FileName, 'r')
nodes←{}  edges←{}
while ~feof(File)
   Row←fgetl(File)
   Parts←strsplit(Row, ',')
   Node1←Parts{1}
   Node2←Parts{2}
   if ~ismember(Node1, Nodes)
      Nodes{end+1}←Node1
   if ~ismember(Node2, Nodes)
      Nodes{end+1}←Node2
Edges{end+1}←[Node1,Node2]
nNodes←length(Nodes)
nEdges←length(Edges)
i←1,2,…,nNodes
  Parts2←strsplit(Edges{i}, ',')
  Node1←str2num(Parts2{1})
  Node2←str2num(Parts2{2})
  A(Node1,Node2)←1
  A(Node2,Node1)←1
```

The execution results of MISA for some graphs were given in Fig.1. The red nodes are elements of MIS, and blue nodes are neighbours of these nodes. Fig. 1 illustrates two solutions for two graphs. The first graph is the Banana Tree and obtained solution is depicted by "red" nodes, and it is maximum independent set. The next graph contains three cliques, so, MISA should obtain a solution of size 3. This case illustrated as on the right in Fig.1.

Fig.2 also includes two maximum independent set solutions for the given graphs. These solutions are also obtained by applying MISA to these graphs. The left graph contains four leafs and two cliques. MISA obtained all leafs as elements of MIS and it also obtained one node for each clique.

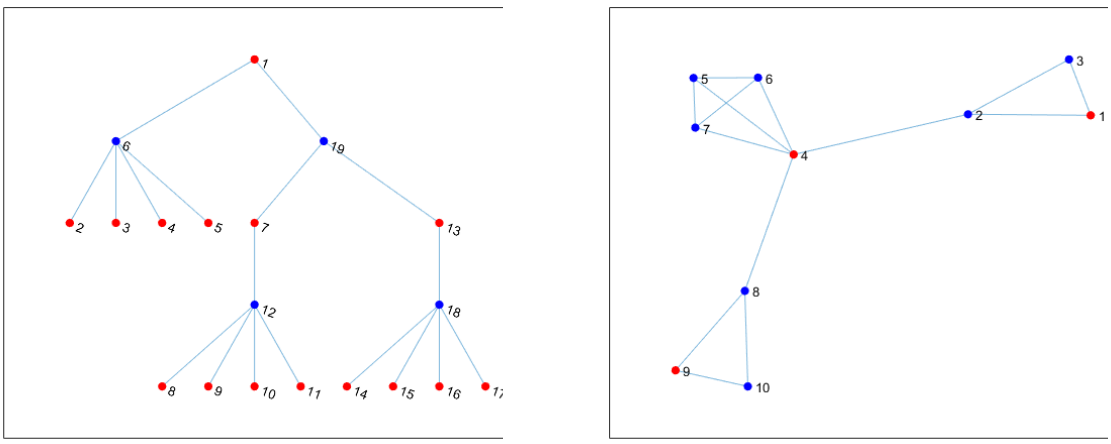

Figure 1. The maximum independent sets for two graphs obtained by MISA.

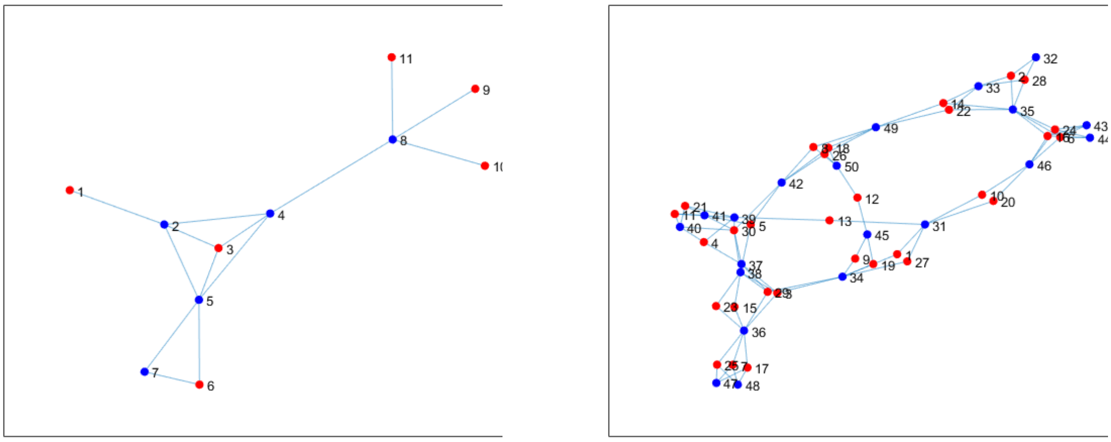

Figure 2. The results of MISA for some example graphs.

3. MISA is Optimum

The MISA is a greedy based algorithm with time and space complexities as polynomials.

Theorem 1. *MISA is an optimum algorithm with respect to the obtained solution.*

Proof**:** Assume that G=(V,E) where |V|=n, |E|=m. The obtained optimum is V_{MIS} and $|V_{MIS}|$=r. The obtained solution is $V_{MIS} = \{v_{k_1},\ v_{k_2}, v_{k_3}, \dots, v_i, \dots, v_{k_r}\}$ and $v_j \in N(v_i)$. We assume that V_{MIS} is not maximum. In this case, $|N(v_i)| \le |N(v_j)|$, $V_{MIS}^2 = V_{MIS}^2 \cup \{v_j\} \setminus \{v_i\}$.

Assume that the first Malatya centrality for node u is

$$\boldsymbol{\Psi_1(u)} = \sum_{u_t \in N(u)} \frac{\mathbf{1}}{d(u_t)} = \overbrace{\frac{\mathbf{1}}{d(u_1)} + \frac{\mathbf{1}}{d(u_2)} + \cdots + \frac{\mathbf{1}}{d(u_{|N(u)|})}}^{u_t \in N(u), 1 \le t \le |N(u)|}$$

has |N(u)| terms and this means that there are |N(u)| neighbours of u. If u is in the maximum independent set, then its |N(u)| neighbours cannot be elements of the maximum independent set (V_{MIS}).

If we return to nodes v_i and v_j. The number of neighbours of v_j is greater than or equal to the number of neighbours of v_i. This case concludes in the shrinking or unchanged the size of V_{MIS}^2. Assume that $N(N(v_i))$ is the set of neighbours of neighbours of v_i. In other words, they are the second level neighbours of v_i. v_i has less or equal number of neighbours with respect to its second level neighbours, since its $\Psi_1(v_i)$ is less than its neighbours $\Psi_1(\dots)$ values.

This case can be shown by using some special types of graphs.

Case 1: Tree: The centrality value of v will be less than the centrality value of its parent u, since d(v)<d(u). In this case the node v will be selected to MIS, and v and its parents will be removed from the graph. In the case of selecting nude u will conclude in removing node u and its leafs neighbours and also parent(u) (Fig.3).

Case 2: Ring: Initially, all nodes in a ring have equal centrality values. In this case, one of the node will be selected to MIS. The selected node with its two neighbours are removed from the graph. The remaining graph can be regarded as a tree graph.

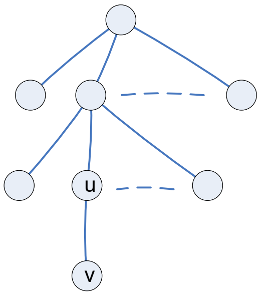

Figure 3. a leaf node and its parent.

Case 3: Grid: A 2D grid graph has four corner nodes whose node degrees are equal to 2. The neighbours of corner nodes have degrees as 3. So, the centrality value for corner node is

$$\Psi_1 = \frac{1}{3} + \frac{1}{3} = \frac{2}{3}$$

The centrality values of corner nodes are minimum, and one of the corner nodes is selected for MIS. Removing selected node and its neighbours will conclude in 5 corner nodes. The corner nodes have the same values for centrality. Selecting corner node causes the number of removed nodes to be minimum. This process goes on in this way.

Case 4: Regular Graph: Assume that G=(V,E) and ∀u,v∈V, d(u)=d(v)=r. Initially, all nodes have centrality value as $\Psi_1 = 1$, One of the node can be selected for MIS, and then selected node and its neighbours are removed from graph. The number of removed node is r+1. After selecting the first node, there are r-1 nodes have degrees as r-1. So, their centrality value will be

$$\Psi_1(u) = \sum_{u_t \in N(u)} \frac{1}{d(u_t)} = \overbrace{\frac{1}{d(u_1)} + \frac{1}{d(u_2)} + \cdots + \frac{1}{d(u_{r-1})}}^{r-1} = \frac{r-1}{r} < 1$$

So, one of these node will be selected for the next step.

Case 5: Random Graph: G=(V,E) is a random graph. Assume that ∀u,v∈V,

$$\Psi_1(u) = \sum_{u_t \in N(u)} \frac{1}{d(u_t)} = \overbrace{\frac{1}{d(u_1)} + \frac{1}{d(u_2)} + \cdots + \frac{1}{d(u_{|N(u)|})}}^{u_t \in N(u), 1 \le t \le |N(u)|} \text{ and}$$

$$\Psi_1(v) = \sum_{v_t \in N(u)} \frac{1}{d(v_t)} = \overbrace{\frac{1}{d(v_1)} + \frac{1}{d(v_2)} + \cdots + \frac{1}{d(v_{|N(v)|})}}^{v_t \in N(v), 1 \le t \le |N(v)|}$$

The same fractions in $\Psi_1(u)$ and $\Psi_1(v)$ are removed. The remaining fractions should be sorted in ascending order for both centrality values. The centrality value with larger denominators will be smaller and it is selected for the next step. This means that after selection process, the number of removed nodes will be less than with respect to other node ■

Theorem 2: *MISA is an optimum algorithm.*

Proof: Assume that G=(V,E) is a graph. u,v∈V, and Ψ_1(u) is the first Malatya centrality value of node u, and d Ψ_1(v) is the first Malatya centrality value of node v. d(u) is the degree of node u and d(v) is the degree of node v. The following cases can be taken in consideration.

a) Assume that d(u)=d(v), then both nodes have the same number of the neighbours. The summation of degrees of each node can be computed as follow.
$D_u = \sum_{v_i \in N(u)} d(v_i)$ and $D_v = \sum_{v_j \in N(u)} d(v_j)$
i) If $D_u > D_v$, then Ψ_1(u)< Ψ_1(v). In this case, u is selected to MIS.
ii) If $D_v > D_u$, then Ψ_1(v)< Ψ_1(u). In this case, v is selected to MIS.
iii) If $D_u = D_v$, then Ψ_1(u)=Ψ_1(v). In this case, u or v can be selected to MIS.

All of these cases conserve the optimality of MIS.

b) Assume that d(u)>d(v), then

$$\Psi_1(u) = \sum_{u_t \in N(u)} \frac{1}{d(u_t)} = \overbrace{\frac{1}{d(u_1)} + \frac{1}{d(u_2)} + \cdots + \frac{1}{d(u_{|N(u)|})}}^{u_t \in N(u), 1 \le t \le |N(u)|} \text{ and}$$

$$\Psi_1(v) = \sum_{v_t \in N(u)} \frac{1}{d(v_t)} = \overbrace{\frac{1}{d(v_1)} + \frac{1}{d(v_2)} + \cdots + \frac{1}{d(v_{|N(v)|})}}^{v_t \in N(v), 1 \le t \le |N(v)|}$$

The number of terms in $\Psi_1(v)$ is less than the number of terms in $\Psi_1(u)$. Selecting node v will reduce the number of nodes excluded from the MIS set, since the neighbours of the selected node are excluded from the set MIS. Therefore, the potential for MIS maximization will continue.

c) Assume that d(u)<d(v), then

$$\boldsymbol{\Psi_1(u) = \sum_{u_t \in N(u)} \frac{1}{d(u_t)} = \overbrace{\frac{1}{d(u_1)} + \frac{1}{d(u_2)} + \cdots + \frac{1}{d(u_{|N(u)|})}}^{u_t \in N(u), 1 \le t \le |N(u)|} \text{ and}}$$

$$\boldsymbol{\Psi_1(v) = \sum_{v_t \in N(u)} \frac{1}{d(v_t)} = \overbrace{\frac{1}{d(v_1)} + \frac{1}{d(v_2)} + \cdots + \frac{1}{d(v_{|N(v)|})}}^{v_t \in N(v), 1 \le t \le |N(v)|}}$$

The number of terms in $\Psi_1(u)$ is less than the number of terms in $\Psi_1(v)$, since |V(v)|>|N(u)|. Selecting node u will reduce the number of nodes excluded from the MIS set, since the neighbours of the selected node are excluded from the set MIS. Therefore, the potential for MIS maximization will continue ■

Theorem 3*: MISA maximizes the cardinality of MIS.*

Proof: Assume that G=(V,E), |V|=n, and |E|=m. Without losing generality, $v_i \in V$, $d(v_1) \le d(v_2) \le \ldots \le d(v_n)$. $\Psi_1(v_i)$ has the $d(v_i)$ terms (fractions), and all of these fractions will be less than or equal to one. A smaller value for these fractions indicates that the corresponding node has neighbours of higher degrees. In this case, selecting this node will exclude less nodes from the set MIS. Without losing generality,

$$\Psi_1(v_1) \le \Psi_1(v_2) \le \Psi_1(v_3) \le \cdots \le \Psi_1(v_n)$$

This means that the node v_1 has at least neighbours and selecting v_1 concludes in less nodes excluded from MIS. The removing the selected nodes and its neighbours from the graph will conclude in at least one node **with** minimum degree. The first Malatya centrality of this node will be less all of other nodes. Then this node will be subject to selection for the next step. This process carries on in this way.
Assume that u and v are two nodes in the given graph.

$$\Psi_1(u) = \sum_{u_t \in N(u)} \frac{1}{d(u_t)} = \overbrace{\frac{1}{d(u_1)} + \frac{1}{d(u_2)} + \cdots + \frac{1}{d(u_q)}}^{u_t \in N(u), 1 \le t \le |N(u)| = q}$$

$$\Psi_1(v) = \sum_{v_t \in N(v)} \frac{1}{d(v_t)} = \overbrace{\frac{1}{d(v_1)} + \frac{1}{d(v_2)} + \cdots + \frac{1}{d(v_r)}}^{u_t \in N(u), 1 \le t \le |N(u)| = r}$$

a) $\forall u_i, v_j$, $d(u_i)>d(v_j)$ then $\frac{1}{d(v_j)} > \frac{1}{d(u_i)}$. If r>q then, $\Psi_1(v) > \Psi_1(u)$ and the selected node will be u. The number of removed nodes from graph will be 1+q, since 1+r>1+q. In the case of selecting node v, the number of removed nodes from graph will be increased.
b) In the next case, $\Psi_1(v)$ and $\Psi_1(u)$ can be expressed as follows.

$$\Psi_1(u) = \frac{\sum_{i=1}^{q-1} P(q,q-1)}{\prod_{i=1}^{q} d(u_i)} \text{ and } \Psi_1(v) = \frac{\sum_{j=1}^{r-1} P(r,r-1)}{\prod_{j=1}^{r} d(v_i)}$$

where the expression P(q,q-1) is the q-1 permutation calculation of the q neighbour degrees of node u and is also used to express the product of the degrees. The expression P(r,r-1) is the r-1 permutation calculation of the r neighbour degrees of node v and is also used to express the product of the degrees. The greater centrality value indicates the more second level neighbours. So, selecting smaller one concludes in larger node number for the next step ■

4. Conclusion

This paper attempts to analytically demonstrate that the MISA algorithm achieves optimum results. It also demonstrates that each node selection step removes the fewest possible nodes from the graph. This allows the MISA algorithm to reach the maximum number of elements possible.

References

Yakut, S., Öztemiz, F., Karcı, A., (2023a)"A new robust approach to solve minimum vertex cover problem: Malatya vertex-cover algorithm" The Journal of Supercomputing, Vol:79, p:19746–19769.

Yakut, S., Öztemiz, F., Karcı, A.,(2023b) " A New Approach Based on Centrality Value in Solving the Maximum Independent Set Problem: Malatya Centrality Algorithm", Journal of Computer Science, Vol:8, p:16-23.

Karcı, A., Yakut,S., Öztemiz, F., (2022) "A New Approach Based on Centrality Value in Solving the Minimum Vertex Cover Problem: Malatya Centrality Algorithm", Journal of Computer Science, Vol:7, p:81-88.

Öztemiz, F., Yakut, S.,(2024a), "An Effective Method for Determining Node Dominance Values: Malatya Centrality Algorithm", 32nd Signal Processing and Communications Applications Conference (SIU-2024).

Öztemiz, F., Yakut, S., (2024b), "Analysis of the Malatya Centrality-Based Clique Method on DIMACS Benchmarks and Random Graphs", Malatya Turgut Ozal University Journal Of Engineering And Natural Sciences , vol:5, p:63-69.

Öztemiz, F., (2025a), "An Effective Algorithm for Edge Coloring: Malatya Edge Coloring Algorithm", Turkish Journal of Science & Technology, Vol: 20, p:309-325.

Öztemiz, F., (2025b) "A greedy approach to solve maximum independent set problem: Differential Malatya independent set algorithm", Engineering Science and Technology, an International Journal, Vol:63.

Bakan, C.T., Yakut, S., (2023) "Development of Text Summarization Method based on Graph Theory and Malatya Centrality Algorithm", Special Issue: Journal of Computer Science, International Artificial Intelligence and Data Processing Symposium, Vol:IDAP-2023, p:90-99.

Yakut, S.(2025), "A robust and efficient algorithm for graph coloring problem based on Malatya centrality and sequent independent sets", Egyptian Informatics Journal, Vol:30, 100676.

Okumuş, F., Karci, Ş.,(2024), "MDSA: A Dynamic and Greedy Approach to Solve the Minimum Dominating Set Problem",Applied Science, Vol:14, https://doi.org/10.3390/app14209251.

Karci, Ş., Okumuş, F., Karci, A., (2023), "Calculating the Centrality Values According to the Strengths of Entities Relative to their Neighbours and Designing a New Algorithm for the Solution of the Minimal Dominating Set Problem", Journal of Computer Science, Vol:8, p:50-56.

Karci, Ş., Okumuş, F., (2025), "Analytical Validation of the Malatya Dominating Set Algorithm: Constructing Optimal Dominating Sets Without Redundant Nodes", Muş Alpaslan University Journal of Sciences, Vol:13, p:46-56, 10.18586/msufbd.1643589.