NATURENGS
E-ISSN 2717-8013
https://dergipark.org.tr/en/pub/naturengs

Analysis of the Malatya Centrality-Based Clique Method on DIMACS Benchmarks and Random Graphs

Furkan Öztemiz [a], Selman Yakut [a,*]

[a] Inonu University, Department of Software Engineering, Malatya Türkiye – 44280
*Corresponding author

ARTICLE INFO

Received 26.11.2024
Accepted 25.12.2024

Doi: 10.46572/naturengs.1591530

ABSTRACT

The Maximum Clique Problem (MCP) is an NP-complete problem that aims to find the largest clique (complete subgraph) within a given graph. This problem has widespread applications in fields such as social network analysis, bioinformatics, chemical information systems, and communication networks. Although various solution methods have been proposed in the literature, an effective method that provides a general solution across different graph types is not yet available. In this study, we demonstrate that our previously proposed maximum clique method provides an effective solution for MCP on DIMACS benchmark graphs and random graphs generated using different models such as Erdos-Renyi, Forest Fire, Watts-Strogatz, and Regular Random models. The proposed method begins by computing the complement of the original graph. In the complement graph, the maximum independent set is determined using the Malatya Independent Set Algorithm, an efficient algorithm well-recognized in the literature. The maximum independent set identified in the complement graph is then used to determine the subgraph corresponding to the maximum clique in the original graph. The effectiveness of the proposed method was validated through tests on widely recognized DIMACS benchmark graphs. Additionally, to evaluate its performance and robustness on unpredictable graphs, tests were conducted on random graphs of varying complexities generated using Erdos-Renyi, Forest Fire, Watts-Strogatz, and Regular Random models. The results of these tests demonstrated that the method produced optimal or near-optimal MCP solutions for both DIMACS and random graphs. The successful outcomes further highlight the efficiency of the proposed method and its potential applicability in MCP-related domains.

Keywords: Maximum clique; Maximum clique problem; Independent sets; Malatya centrality; Malatya independent set algorithm

1. Introduction

The Maximum Clique Problem (MCP) involves identifying the largest complete subgraph in a given graph, representing the largest set of nodes that are directly connected to each other. These fully connected subsets are widely utilized in various domains, including social network analysis, bioinformatics, chemical information systems, and communication networks [1]. For instance, in social networks, the largest group of individuals who all know each other forms a clique, representing mutual friend groups [2]. Due to the lack of general solutions across different graph types in the literature, MCP is classified as a challenging NP-complete problem [3].

As an NP-complete problem, MCP solutions often rely on heuristic algorithms or approximation methods. Additionally, certain problem formulations within the domain of clique problems focus on identifying specific subsets, such as finding the largest clique, determining the largest weighted clique in weighted graphs, identifying all maximal cliques, or verifying whether a graph contains cliques above a certain size. These problems necessitate determining the maximum clique in many applications and graph types.

Numerous methods have been proposed in the literature to solve the MCP [1]. However, these approaches often fail to produce effective solutions for large-scale graphs or to provide general solutions applicable to diverse graph types. Due to its complexity and intractability on large and complex graphs, MCP remains NP-complete. Consequently, heuristic or approximate solutions are commonly employed, though these methods may produce suboptimal or non-optimal results. Furthermore, these solutions are often influenced by the algorithm's performance, as well as the structure and type of the graph. Some methods may require extensive memory usage or lack scalability, particularly when applied to large datasets. Exact methods, while offering precise solutions for specific graph types, are typically ineffective across diverse graph structures [4]. As such, MCP solutions are often tailored to specific use cases and graph structures, highlighting the need for scalable and effective approaches applicable to diverse graph types.

* Corresponding author. e-mail address: selman.yakut@inonu.edu.tr
ORCID : 0000-0002-0649-1993

In this study, we demonstrate the effectiveness of our previously proposed maximum clique method [5] on benchmark graphs and random graphs generated using various models. The proposed method begins by computing the complement of the original graph. Subsequently, the maximum independent set is determined in the complement graph using the Malatya Independent Set Algorithm (MISA), which is based on the Malatya Centrality Algorithm (MCA). The maximum independent set identified in the complement graph is then used to derive the maximum clique in the original graph, as the nodes in the maximum clique correspond to those of the maximum independent set in the complement graph.

To validate the proposed method, tests were first conducted on DIMACS benchmark graphs widely cited in the literature. Additionally, to assess the method's performance on unpredictable graphs, tests were performed on random graphs of varying complexity and density generated using different models. The successful results obtained from tests on DIMACS benchmark graphs and random graphs of varying densities confirm that the proposed method provides effective solutions for these graph types.

This paper is organized as follows: Section 2 reviews the literature on MCP. Section 3 details the proposed method. Section 4 presents evaluations and experimental results. Finally, Section 5 discusses the conclusions and implications of this study.

2. Literature

Numerous methods have been proposed in the literature to solve the MCP. These methods can generally be categorized into three main classes: exact algorithms, heuristic methods, and approximation algorithms [6]. Exact algorithms, while effective for specific types of graphs, often fail to provide efficient solutions across diverse graph types. Techniques such as backtracking and branch-and-bound, for example, produce precise solutions for small to medium-sized graphs but may struggle with large and complex graphs. Heuristic algorithms, on the other hand, leverage techniques such as local search, tabu search, and genetic algorithms, enabling them to handle large-scale graphs effectively. However, the performance of heuristic algorithms is often dependent on parameters such as the number of iterations, and the resulting solution set can vary across runs of the algorithm. Approximation algorithms are typically employed to reduce computational costs for large graphs, trading off some accuracy for scalability. The choice of solution approach for MCP depends on the specific graph structures and application requirements. While exact algorithms guarantee accurate results, heuristic and approximation methods offer practical solutions for large-scale problems. Consequently, different solution strategies are suitable for varying application domains and resource constraints.

For instance, Wang et al. proposed two efficient local search algorithms for the maximum weighted clique problem, utilizing strong configuration checking and perturbation strategies. Their methods achieved notable success on benchmark graphs such as DIMACS and BOLISH [1]. Similarly, Peng et al. developed a hybrid artificial bee colony algorithm for solving the maximum semi-clique problem. Their experimental results showed that the algorithm improved the best-known solutions for 46 out of 112 problem instances and matched the best solutions for 63 instances, demonstrating its effectiveness [7]. Moreover, Yu et al. introduced a local density-based heuristic algorithm designed for large-scale applications, which outperformed traditional methods in terms of accuracy and computational speed [4].

Pattabiraman and Patwary introduced an algorithm leveraging innovative pruning techniques to efficiently identify maximum cliques in large sparse graphs [8]. Douik et al. proposed a framework incorporating various optimal and heuristic solutions based on clique problems to address challenges in communication and signal processing [9]. Belachew and Gillis developed and validated a novel solution algorithm for the MCP using a continuous characterization grounded in the symmetric rank-one non-negative approximation of a given matrix [10]. Zhou et al. presented SMC-BRB, a branch-and-bound algorithm designed for solving the MCP in large and sparse graphs. Experimental results demonstrate that SMC-BRB outperforms state-of-the-art algorithms in both efficiency and accuracy [11]. Pelofske achieved precise MCP solutions for graphs with 120 nodes and 6395 edges using a hybrid method combining parallel quantum annealing with graph decomposition [12]. Hasan et al. employed a metaheuristic algorithm, Chemical Reaction Optimization, to achieve superior results compared to state-of-the-art methods on three benchmark datasets. Their approach involved tuning initial parameters, redesigning reaction operators, and incorporating an additional repair operator to enhance performance [13].

Seda applied integer programming models integrated with exact methods and advanced heuristics to solve the MCP efficiently for large-scale instances, delivering both optimal solutions and high-quality approximations within reasonable timeframes [14]. Segundo et al. developed a new exact algorithm for the MCP, which improves branching schemes and bounding techniques, demonstrating superior performance in identifying maximum cliques within complex graph structures [15]. Reba et al. enhanced the MaxCliqueDyn algorithm, originally designed for finding maximum cliques in protein graphs, by integrating machine learning-based improvements. This augmented approach doubles the search speed in molecular docking graphs used in drug design and outperforms existing methods in specific graph types [16].

Jiang et al. proposed a variant of the Branch-and-Bound algorithm that incorporates upper-bound and shared lower-bound mechanisms, providing highly scalable and efficient solutions to the MCP. Their approach achieves near-linear and, in some cases, super-linear speedup,

particularly on DIMACS and BHOSLIB datasets [6]. Marinelli introduced a novel mathematical model for solving the maximum semi-clique problem, demonstrating its effectiveness on large and dense graphs [17]. Abello et al. proposed approximate solutions for computing cliques and quasi-cliques in large multi-digraphs by employing graph decomposition and a semi-external GRASP method. Their approach was tested on telecommunication datasets, demonstrating its practical applicability [18]. Ertem et al. analyzed the complexity of the Maximum Independent Union of Cliques Problem across various graph types and developed an integer programming formulation, complemented by an exact combinatorial branch-and-bound algorithm to solve this problem [19]. Chang introduced a novel algorithm for finding maximum cliques in large sparse graphs, showcasing its effectiveness through extensive empirical evaluations on real-world datasets [20]. Smith et al. presented a tabu search algorithm for solving the MCP, which incorporates an exact algorithm for subproblems. Comparative analysis revealed that their method not only achieves faster results but also identifies larger cliques in certain cases compared to existing algorithms [21]. Jin et al. developed an algorithm based on the Bron-Kerbosch framework to enumerate all maximal cliques in large social networks. Additional experiments demonstrated its competitiveness with existing methods, particularly for solving maximal clique variants in very large graphs [22].

Solutions proposed in the literature for solving the MCP offer various advantages depending on specific graph types or application contexts. Exact algorithms are effective for certain graph types but often fail to deliver efficient solutions across diverse graph structures. Heuristic and approximation methods, while capable of addressing large-scale problems and graphs, can be sensitive to parameters such as initial conditions and the number of iterations. Depending on the graph type or problem characteristics, the appropriate approach can be selected based on application requirements and resource constraints. There is a critical need in the literature for a general approach that can be applied to various graph types and deliver effective solutions within a reasonable timeframe. Moreover, the development of methods that are independent of graph structure or other variables, and that produce optimal or near-optimal and robust solutions, could significantly impact both graph theory and its practical applications.

3. Material and Method

To demonstrate the effectiveness of the MCA-based maximum clique method [5], the approach was tested on numerous graphs for which analytical proofs are unknown. This method identifies the set of nodes that constitute the solution to the MCP on unweighted and undirected graphs. The proposed method produces solution sets independent of graph type, and the results are optimal or near-optimal. The testing process in this study is summarized in five stages, as illustrated in Figure 1.

In Stage 1, unweighted and undirected random and benchmark graphs, composed of nodes and edges, are used as inputs to the method. The random graphs are generated using the Erdos-Renyi, Forest Fire, Watts-Strogatz, and Regular Random models [23]. Benchmark graphs are selected from the popular DIMACS dataset. For the algorithm to address problems in specific domains, the problem must first be transformed into the required graph format. In Stage 2, the input graphs are formatted appropriately for testing. The complement of the original graph is constructed by adding edges between nodes that are not connected in the original graph and removing edges where connections exist [24]. The Maximum Clique Problem is solved using the graph-theoretical approach MaxClique = Maksimum Independent Set ($\overline{G}$)[19]. In Stage 3, the complement graph of the original graph is generated. In Stage 4, the MCA-based MISA [25] is applied to the complement graph to determine the maximum independent set of nodes. The nodes in the maximum independent set of the complement graph correspond to the nodes in the maximum clique of the original graph. According to graph theory, the maximum independent set nodes in the complement graph are equivalent to the maximum clique nodes in the original graph [26]. Finally, in Stage 5, the nodes identified as the maximum clique in the graph are presented.

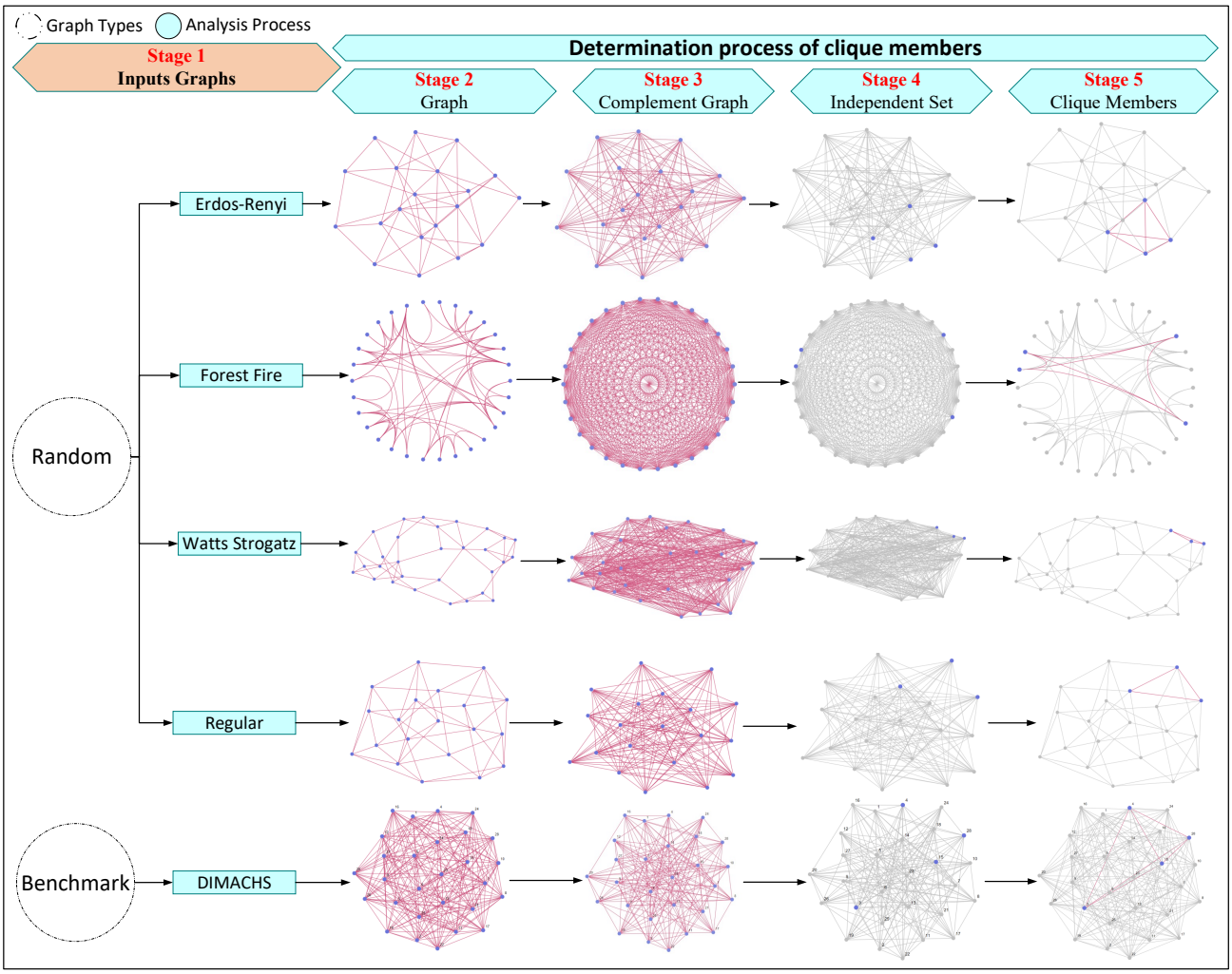

Figure 1. Graphical abstract of the new maximum clique method

3.1. Independent set and clique

In any graph G = (V, E), an independent set is a subset of vertices such that no two vertices in the set share an edge. Formally, if $S \subseteq V$ is an independent set, then $\forall u, v \in$ S,(u,v) $\notin$ E.The maximum independent set is defined as the largest independent set of vertices in a given graph, containing the maximum number of vertices [27]. The size of the independent set is denoted as α(G) and is mathematically formulated as follows:

$$\alpha(G) = \max\{|S|: S \subseteq V \; ve \; S \; is \; independent \; set\} \quad (1)$$

In a graph G = (V, E), a clique is a subset of vertices where every pair of vertices is connected by an edge. Formally, if $C \subseteq V$ is a clique, then $\forall u, v \in C, u \neq v \Rightarrow (u, v) \in E$. In graph theory, the maximum clique is the largest subset of vertices that forms a clique [28]. The size of the maximum clique is denoted as $\omega(G)$, which is mathematically expressed as follows:

$$\omega(G) = \max\{|C|: C \subseteq V \; ve \; C \; is \; clique\} \quad (2)$$

Both the maximum independent set and the MCP is NP-complete, making their solutions computationally challenging. Consequently, these problems are typically addressed using heuristic or approximation methods.

3.2. Solution to the Maximum Clique Problem Using the MISA

The MISA provides effective solutions for the Maximum Independent Set Problem independent of the graph type [25]. MISA produces optimal or near-optimal solutions in polynomial time for various types of graphs. The selection of nodes in MISA solutions is based on the MCA. MCA is a recently developed, efficient centrality algorithm that determines the dominance values of nodes [29]. The centrality value of a node in MCA is calculated as the sum of the ratios of the node's degree to the degrees of its neighboring nodes. The mathematical representation of MCA is given in Equation 3 [30]:

$$\psi(v_i) = \sum_{\forall v_j \in N(v_i)} \frac{d(v_i)}{d(v_j)} \quad (3)$$

Where $\psi(v_i)$ is the Malatya centrality value of *i.* node, *N* is the set of nodes, $d(v_i)$ is the degree of node v_i, $d(v_j)$ is the degrees of the neighbors of v_i.

In the algorithm, MCA is initially applied to the graph to compute the Malatya centrality values of all nodes. Nodes with the lowest Malatya centrality value are selected first to form the MISA solution. The selected node and its neighbors are then removed from the graph.

This process is repeated iteratively, with MCA being recalculated for the updated graph, until all nodes are removed, and the MISA solution set for the graph is determined.

By taking the complement of a graph G denoted as $\bar{G}$, the maximum independent set solution identified in $\bar{G}$ corresponds to the maximum clique solution in the original graph G. This solution approach is widely recognized in the literature, and MISA has demonstrated its effectiveness in producing successful clique solutions across various types of graphs. The pseudocode for the implemented algorithm is provided in Algorithm 1.

Algorithm 1. Pseudocode of the maximum clique method

Maksimum independent set and maksimum clique[5]

$\bar{G}$ <- Complement (G)

Input : Adjacency matrix of $\bar{G}$ is A and $\bar{G}$ =(V,E)

Output: $V_{ind} \subseteq V$, V_{ind} :It is the independent set solution in the graph $\bar{G}$

$V_{ind} \leftarrow \emptyset$

While $E \neq \emptyset$ do

i←1, …, |V|

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

$$V_{ind} = V_{ind} \cup \{\min(\psi(v_i))\}$$

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

Output=V_{ind} : $V_{ind}(\bar{G}) = V_{clique}(G)$

4. Experimental results

In this study, tests were conducted on 61 different graphs, including 36 DIMACS benchmark graphs and 25 random graphs. DIMACS benchmark graphs are widely used for testing significant NP graph problems, such as vertex coloring, maximum clique, and maximum independent set. Table 1 presents the clique results obtained by the MISA, which is based on the MCA, for DIMACS graphs. Additionally, the table includes the known optimal clique results for these graphs. Optimal results produced by MISA are marked with an asterisk (*). For example, the MANN-a9 graph consists of 45 vertices and 72 edges. MISA determines the maximum clique value for this graph as 16, which is the optimal result. Another example is the brock200-4 graph, which contains 200 vertices and 6811 edges. After applying MISA, the maximum clique value is identified as 14, whereas the known optimal maximum clique value for this graph is 17. An overall examination of the table shows that, while MISA achieves optimal results for some DIMACS graphs, it generally provides approximate solutions.

Table 1. MISA results on DIMACS benchmark graphs

$\bar{G}$	V	E	Optimum	MISA
MANN-a9	45	72	16	16*
MANN-a27	378	702	126	125
brock200-1	200	5,066	21	20
brock200-2	200	10,024	12	9
brock200-3	200	7,852	15	13
brock200-4	200	6,811	17	14
brock400-1	400	20,077	27	22
brock400-2	400	20,014	29	22
brock400-3	400	20,119	31	23
brock400-4	400	20,035	33	22
c-fat200-1	200	18,366	12	12*
c-fat200-2	200	16,665	24	24*
c-fat200-5	200	11,427	58	58*
c-fat500-1	500	120,291	14	14*
c-fat500-2	500	115,611	26	26*
c-fat500-5	500	101,559	64	64*
c-fat500-10	500	78,123	126	126*
hamming6-2	64	192	32	32*
hamming6-4	64	1,312	4	4*
hamming8-2	256	1,024	128	128*
hamming8-4	256	11,776	16	16*
johnson8-2-4	28	168	4	4*
johnson8-4-4	70	560	14	14*
johnson16-2-4	120	1,680	8	8*
johnson32-2-4	496	14,880	16	16*
san200-0.9-1	200	1990	70	48
san200-0.9-2	200	1,990	60	41
san200-0.9-3	200	1,990	44	34
san400-0.5-1	400	39,900	13	9
san400-0.7-1	400	23,940	40	22
san400-0.7-2	400	23,940	30	18
san400-0.7-3	400	23,940	22	18
sanr200-0.7	200	6,032	18	15
sanr200-0.9	200	2,037	42	41
sanr400-0.5	400	39, 816	13	11
sanr400-0.7	400	23, 931	21	18

Table 2 presents analyses conducted on various random graphs. For instance, a graph generated using the Erdos-Renyi model with parameters 3000(p=~0.30) p≈0.30 contains 3000 vertices and 1,343,049 edges. The complement of this graph has 3,155,451 edges. After applying MISA, the maximum clique value was determined to be 8.

Another example involves a graph generated using the Forest Fire model with parameters (200, 0.2, 1, 2), consisting of 200 vertices and 643 edges. The complement of this graph contains 19,257 edges. MISA identified the maximum clique value as 5. The table also includes analyses of large graphs. For instance, a graph generated using the Erdos-Renyi model with parameters 30000 (p≈0.25) consists of 30,000 vertices and 112,498,073 edges, while its complement graph has 337,486,927 edges. After applying MISA, the maximum clique value was determined to be 9. Although the maximum clique results obtained do not guarantee

optimality, the ability of MISA to produce solutions in polynomial time for large graphs is a significant advantage.

Table 2. Maximum clique results in random and large graphs

Random Graphs		(G)		$(\bar{G})$		MISA- $\alpha(\bar{G})$ Maximum Clique
		V	E	V	E	
Erdos-Renyi	1000(p=~0.20)	1000	100,323	1000	399,177	7
	2000(p=~0.25)	2000	498,996	2000	1,500,004	7
	3000(p=~0.30)	3000	1,343,049	3000	3,155,451	8
	4000(p=~0.35)	4000	2,796,887	4000	5,201,113	11
	5000(p=~0.40)	5000	5,000,287	5000	7,497,213	12
Forest Fire	(100, 0.2, 1, 2)	100	323	100	4,627	6
	(200, 0.2, 1, 2)	200	643	200	19,257	5
	(300, 0.2, 1, 2)	300	962	300	43,888	5
	(400, 0.2, 1, 2)	400	1,298	400	78,502	6
	(500, 0.2, 1, 2)	500	1,614	500	123,136	5
Watts Strogatz	(100, 0.1)	100	200	100	4,750	3
	(200, 0.1)	200	400	200	19,500	3
	(300, 0.1)	300	600	300	44,250	3
	(400, 0.1)	400	800	400	79,000	3
	(500, 0.1)	500	1000	500	123,750	3
Regular	(100, 3)	100	150	100	4800	2
	(100, 5)	100	250	100	4700	3
	(100, 8)	100	400	100	4550	3
	(100, 10)	100	500	100	4450	3
	(100, 15)	100	750	100	4200	3
Big Graphs	10000(p=~0.20)	10000	9,998,560	10000	39,996,440	8
	10000(p=~0.25)	10000	12,499,449	10000	37,495,551	9
	20000(p=~0.20)	20000	39,993,590	20000	159,996,410	8
	30000(p=~0.20)	30000	90,012,939	30000	359,972,061	8
	30000(p=~0.25)	30000	112,498,073	30000	337,486,927	9

5. Conclusions

In this study, the effectiveness of our previously proposed maximum clique method for solving the MCP was demonstrated on DIMACS benchmark graphs and random graphs generated using various models. The proposed method takes unweighted and undirected graphs as input and first computes the complement of the original graph. Subsequently, the nodes forming the maximum independent set in the complement graph are identified. To determine the maximum independent set, the MISA, which is based on the MCA, was employed as an efficient algorithm recognized in the literature. The proposed method's effectiveness in producing solutions for MCP was validated through tests conducted on graphs of varying types and complexities. These tests utilized DIMACS benchmark graphs as well as random graphs generated using the Erdos-Renyi, Forest Fire, Watts-Strogatz, and Regular Random models. To further evaluate the method's robustness on unpredictable graphs, random graphs with varying complexities and densities were generated using different generator models. The test results indicated that the proposed method effectively and robustly identified the maximum cliques in benchmark graphs and various random graphs. In conclusion, this study demonstrated that the proposed method provides effective and successful solutions for MCP across different graph types, including benchmark graphs and various types of random graphs.

Future work could focus on identifying cliques of different sizes and extending the approach to solve MCP in other graph types, such as weighted graphs. Additionally, the exploration of problems and application areas where MCP is utilized could serve as a significant avenue for further research.

References

[1] **Y. Wang, S. Cai, J. Chen, and M. Yin,** "SCCWalk: An efficient local search algorithm and its improvements for maximum weight clique problem," Artif. Intell., vol. 280, p. 103230, 2020, doi: 10.1016/j.artint.2019.103230.

[2] **M. J. Brusco, D. Steinley, and A. L. Watts**, "A maximal-clique-based set-covering approach to overlapping community detection," Optim. Lett., no. 0123456789, 2023, doi: 10.1007/s11590-023-02054-0.

[3] **D. Kumlander,** "Some Practical Algorithms to Solve The Maximum Clique Problem," 2005.

[4] **Z. Yu, X. Wang, and J. Zhao,** "A local dense decision-making method for solving the maximum clique problem in large-scale undirected graphs," Comput. Electr. Eng.,

vol. 102, no. April, p. 108220, 2022, doi: 10.1016/j.compeleceng.2022.108220.

[5] **S. Yakut and F. Öztemiz,** "A New Maximum Clique Method," in International Informatics Congress, 2024, pp. 200–208.

[6] **H. Jiang, K. Bai, H. J. Liu, C. M. Li, F. Manyà, and Z. H. Fu,** "Parallel Bounded Search for the Maximum Clique Problem," J. Comput. Sci. Technol., vol. 38, no. 5, pp. 1187–1202, 2023, doi: 10.1007/s11390-022-1803-8.

[7] **B. Peng, L. Wu, Y. Wang, and Q. Wu,** "Solving maximum quasi-clique problem by a hybrid artificial bee colony approach," Inf. Sci. (Ny)., vol. 578, pp. 214–235, 2021, doi: 10.1016/j.ins.2021.06.094.

[8] **B. Pattabiraman, M. M. A. Patwary, A. H. Gebremedhin, W. K. Liao, and A. Choudhary,** "Fast algorithms for the maximum clique problem on massive sparse graphs," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 8305 LNCS, no. 00, pp. 156–169, 2013, doi: 10.1007/978-3-319-03536-9_13.

[9] **A. Douik, H. Dahrouj, T. Y. Al-Naffouri, and M. S. Alouini,** "A Tutorial on Clique Problems in Communications and Signal Processing," Proc. IEEE, vol. 108, no. 4, pp. 583–608, 2020, doi: 10.1109/JPROC.2020.2977595.

[10] **M. T. Belachew and N. Gillis,** "Solving the Maximum Clique Problem with Symmetric Rank-One Non-negative Matrix Approximation," J. Optim. Theory Appl., vol. 173, no. 1, pp. 279–296, 2017, doi: 10.1007/s10957-016-1043-6.

[11] **M. Zhou, Q. Zeng, and P. Guo,** "SMC-BRB: an algorithm for the maximum clique problem over large and sparse graphs with the upper bound via s+ -index," J. Supercomput., vol. 79, no. 7, pp. 8026–8047, 2023, doi: 10.1007/s11227-022-04982-7.

[12] **E. Pelofske, G. Hahn, and H. N. Djidjev,** "Solving larger maximum clique problems using parallel quantum annealing," Quantum Inf. Process., vol. 22, no. 5, pp. 1–22, 2023, doi: 10.1007/s11128-023-03962-x.

[13] **M. Hasan, M. R. Islam, and A. G. Mugdha,** "Solving maximum clique problem using chemical reaction optimization," Opsearch, vol. 60, no. 3, pp. 1230–1266, 2023, doi: 10.1007/s12597-023-00654-z.

[14] **M. Seda,** "The Maximum Clique Problem and Integer Programming Models, Their Modifications, Complexity and Implementation," Symmetry (Basel)., pp. 1–19, 2023.

[15] **P. San Segundo, F. Furini, D. Álvarez, and P. M. Pardalos,** "CliSAT: A new exact algorithm for hard maximum clique problems," Eur. J. Oper. Res., vol. 307, no. 3, pp. 1008–1025, 2023, doi: 10.1016/j.ejor.2022.10.028.

[16] **K. Reba, M. Guid, K. Rozman, D. Janežič, and J. Konc,** "Exact maximum clique algorithm for different graph types using machine learning," Mathematics, vol. 10, no. 1, 2022, doi: 10.3390/math10010097.

[17] **F. Marinelli, A. Pizzuti, and F. Rossi,** "LP-based dual bounds for the maximum quasi-clique problem," Discret. Appl. Math., vol. 296, pp. 118–140, 2021, doi: 10.1016/j.dam.2020.02.003.

[18] **J. Abello, P. Pardalos, and M. Resende,** "On maximum clique problems in very large graphs," AT&T Labs Res. Tech. Rep., no. November 1998, pp. 119–130, 1999, doi: 10.1090/dimacs/050/06.

[19] **Z. Ertem, E. Lykhovyd, Y. Wang, and S. Butenko,** "The maximum independent union of cliques problem: complexity and exact approaches," J. Glob. Optim., vol. 76, no. 3, pp. 545–562, 2020, doi: 10.1007/s10898-018-0694-2.

[20] **L. Chang,** "Efficient maximum clique computation and enumeration over large sparse graphs," VLDB J., vol. 29, no. 5, pp. 999–1022, 2020, doi: 10.1007/s00778-020-00602-z.

[21] **D. H. Smith, R. Montemanni, and S. Perkins**, "The use of an exact algorithm within a tabu search maximum clique algorithm," Algorithms, vol. 13, no. 10, pp. 1–12, 2020, doi: 10.3390/A13100253.

[22] **Y. Jin, B. Xiong, K. He, Y. Zhou, and Y. Zhou, "**On fast enumeration of maximal cliques in large graphs," Expert Syst. Appl., vol. 187, no. September 2021, p. 115915, 2022, doi: 10.1016/j.eswa.2021.115915.

[23] **S. Horvát et al.,** "IGraph/M: graph theory and network analysis for Mathematica," J. Open Source Softw., vol. 8, no. 81, p. 4899, 2023, doi: 10.21105/joss.04899.

[24] **S. A. Mojallal and P. Hansen,** "On the difference of energies of a graph and its complement graph," Linear Algebra Appl., vol. 595, pp. 1–12, 2020, doi: 10.1016/j.laa.2020.02.026.

[25] **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. 55, no. 35, pp. 1–100, Jan. 2023, doi: 10.53070/bbd.1224520.

[26] **N. Bousquet, A. Lagoutte, and S. Thomassé,** "Clique versus independent set," Eur. J. Comb., vol. 40, no. February, pp. 73–92, 2014, doi: 10.1016/j.ejc.2014.02.003.

[27] **M. Xiao, S. Huang, Y. Zhou, and B. Ding,** "Efficient Reductions and a Fast Algorithm of Maximum Weighted Independent Set," in In Proceedings of the Web Conference 2021 (WWW '21). Association for Computing Machinery, 2021, pp. 3930–3940. [Online]. Available: https://doi.org/10.1145/3442381.3450130

[28] **P. M. Pardalos and J. Xue,** "The maximum clique problem," J. Glob. Optim., vol. 4, no. 3, pp. 301–328, 1994, doi: 10.1007/BF01098364.

[29] **F. Öztemiz and S. Yakut,** "An Effective Method for Determining Node Dominance Values: Malatya Centrality Algorithm," 32nd IEEE Conf. Signal Process. Commun. Appl. SIU 2024 - Proc., pp. 1–4, 2024, doi: 10.1109/SIU61531.2024.10601155.

[30] **S. Yakut, F. Öztemiz, and A. Karci,** "A new robust approach to solve minimum vertex cover problem: Malatya vertex-cover algorithm," J. Supercomput., vol. 79, no. 17, pp. 19746–19769, 2023, doi: 10.1007/s11227-023-05397-8.