# Final Report Database Systems - **Improved Algorithms for Maximal Clique Search in Uncertain Networks. 2019 IEEE 35th International Conference on Data Engineering (ICDE)**

Shaurya Tiwari[1], Aditya Sugandhi[2], and Jessica Neelam[3]

[1]Graduate, Department of Computer Science, FSU

Keywords:
Cutting-edge Algorithm
Maximal Clique Enumeration
Uncertain Graphs
Databases
Graph Size
Clique Integrity

Abstract Our report delves into the analysis of cutting-edge algorithms implemented for the enumeration of maximal cliques in uncertain graphs, as proposed in "Improved Algorithms for Maximal Clique Search in Uncertain Networks." We tackle the computational inefficiencies inherent in current methods by examining two core-based pruning algorithms and a cut-based optimization technique from the paper. These approaches offer considerable reductions in graph size while maintaining the integrity of cliques. We implemented these algorithms and assessed their performance against established benchmarks using real-world datasets, showcasing their potential to significantly advance the domain of uncertain graph analysis in network studies and Bio-informatics.

 Submitted: November 3rd, 2023 as the Final Report DBS.

## 1. Introduction to the paper

Uncertainty is an inherent characteristic of real-world networks, where connections between entities are often probabilistic rather than deterministic. Uncertain graphs, which represent these networks, have gained prominence in various domains, including social networks, biological networks, and sensor networks. Maximal clique enumeration, the task of identifying cohesive subgroups within these networks, plays a crucial role in understanding their underlying structure and dynamics. Existing algorithms for maximal clique enumeration in uncertain graphs face significant challenges in handling large-scale networks due to their high time complexity. The computational burden grows exponentially with the number of nodes in the graph, limiting the applicability of these methods in real-world scenarios.

## 2. Problem Statement

The state-of-the-art algorithm for enumerating all maximal (k,$\tau$)- cliques is very costly when handling large uncertain graphs, as its time complexity is proportional to $2^n$ where n is the number of nodes in the uncertain graph. This challenge hinders the effective analysis and processing of large uncertain datasets, limiting the potential benefits of maximal clique enumeration in real-world applications.

## 3. Related Terms

### 3.1 Prerequisites

1. **Uncertain Graph:** Uncertain graphs are a type of graph where the edges are associated with probabilities, representing the likelihood of their existence. Each edge in this graph has an associated probability, which represents the likelihood that this connection (edge) exists.
2. **Clique:** In graph theory, a clique is a subset of vertices of an undirected graph such that every two distinct vertices in the clique are adjacent; that is, every two vertices in the clique are connected by an edge.
3. **Maximal Clique:** A maximal clique is a clique that cannot be extended by including one more adjacent vertex, meaning it's not a subset of a larger clique. Note that there can be more than one maximal clique in a graph.

### 3.2 Mathematical Reasoning:

**Uncertain graphs** are graphs where the existence of edges is uncertain. This uncertainty can be modeled in a variety of ways, such as using probabilities, fuzzy sets, or rough sets. **Cliques** are subgraphs in which all pairs of vertices are adjacent. **Maximal cliques** are cliques that are not contained in any other clique. The (k, $\tau$)-clique problem is the problem of finding all cliques of size at least k in a graph with probability at least . **Finding a maximal** (k, $\tau$)-clique is a more challenging problem, as it requires finding a clique that is both large and has a high probability of existing. Given an uncertain graph $G$ and two parameters $k$ and $\tau$, the problem is to find a maximal clique $C$ in $G$ such that:

1. $|C| \geq k$
2. $P(C \text{ is a clique in } G) \geq \tau$
3. $C$ is maximal, meaning that there is no clique $C'$ in $G$ such that $C' \supset C$ and $P(C' \text{ is a clique in } G) \geq \tau$

## 4. Contributions

The main contributions of this paper include:

1. **Two novel core-based pruning algorithms:**
   - These algorithms effectively reduce the graph size without missing any maximal cliques by identifying and pruning nodes that are not part of any core, a minimal subset of nodes $n'(where n' < n)$ that maintains the clique property.
   - The algorithms utilize a dynamic approach on the Bron-Kerbosch algorithm to efficiently iden-

tify cores using backtracking, avoiding redundant computations.
    - The algorithms significantly reduce the search space for maximal cliques, leading to substantial improvements in execution time.
2. **A cut-based optimization technique:**
    - This technique further enhances pruning performance by identifying and analyzing cuts of cliques, sets of nodes that, when removed, disrupt the clique property.
    - The technique employs a novel cut-identification method to efficiently identify cuts without introducing significant overhead.
    - The technique prunes nodes based on the identified cuts, eliminating additional nodes that are unlikely to be part of any maximal cliques.
    - The technique further reduces the computational burden and improves the overall efficiency of the maximal clique enumeration process.
3. **Improved scalability:**
    - The proposed algorithms exhibit superior scalability compared to existing methods, enabling the analysis of large uncertain graphs that were previously intractable.
    - The combination of core-based pruning and cut-based optimization significantly reduces the computational complexity of the algorithm.
    - The algorithms demonstrate superior performance on datasets of varying sizes, effectively handling large-scale uncertain graphs.
4. **Enhanced effectiveness:**
    - The algorithms maintain the completeness of the results, ensuring that no maximal clique is missed.
    - The algorithms preserve the accuracy of the results while achieving significant improvements in efficiency.
    - The algorithms guarantee that all maximal cliques satisfying the given size and cohesiveness requirements are correctly identified within the pruned nodes $n'$ instead of n (where $n' < n$).

## 5. Related Work

The state-of-the-art method for tackling the issue of mining maximal cliques in uncertain graphs seems to be an area of ongoing research with various approaches being optimized for different aspects of the problem. No single method stands out as the definitive best, but several notable techniques and improvements are highlighted in the passage:

1. **Core-based and Cut-based Optimization Techniques:** The current work under discussion focuses on advanced pruning techniques to develop faster algorithms for computing maximal cliques in uncertain graphs. These are proposed to overcome the limitations of existing methods for large uncertain graphs by reducing computational complexity.
2. **Bron-Kerbosch Algorithm with Improvements:** The Bron-Kerbosch algorithm, especially with the greedy pivoting technique introduced by Tomita et al. [1] and the degeneracy ordering heuristics by Eppstein et al. [3], has been considered optimal in terms of time complexity for listing all maximal cliques. This algorithm is widely recognized in the literature for its efficiency and has seen various enhancements to adapt to different data structures and memory constraints.
3. **Branch and Bound Techniques:** Several papers have focused on branch and bound techniques with various optimizations, such as the Russian Doll Search framework, MaxSAT bounds, and coloring bounds. These techniques are tailored for deterministic graphs but have served as a foundation for subsequent research into uncertain graphs.
4. **Algorithms for Special Graph Data:** Research by Viard et al. [4] and Li et al. [5] on temporal graphs and signed graphs, respectively, represents specialized approaches to the problem, addressing the unique characteristics of these graph types.
5. **Parallel Algorithms and I/O Efficiency:** The work by Cheng et al. [6], [2] for disk-resident graphs and parallel computation addresses the practical aspects of maximal clique enumeration, making it suitable for real-world applications where memory and computational resources are significant constraints.
6. **Algorithms for Large Uncertain Graphs:** While earlier algorithms, such as the one presented by Miao et al. [7], have endeavored to tackle the maximal clique problem in uncertain graphs, their efficiency is compromised when applied to large graphs due to their exponential time complexity. Many existing algorithms are designed within the framework of deterministic graphs, rendering these techniques unsuitable for direct application to sizable uncertain graphs.

In summary, the state-of-the-art in maximal clique mining in uncertain graphs is multifaceted, with the latest work aiming to improve upon the computational efficiency of existing methods, especially for large-scale and uncertain graph data. It involves a combination of core-based pruning to reduce the size of the problem space, cut-based techniques to enhance the efficiency of the pruning process, and adaptations of proven deterministic graph algorithms to the uncertain graph context.

## 6. Novelty of the Selected Paper

### 6.1 Proposed Idea

The paper *Improved Algorithms for Maximal Clique Search in Uncertain Networks* by Rong-Hua Li et al. proposes two new core-based pruning techniques, an improved enumeration technique, and a cut-based optimization technique to improve the efficiency of maximal clique search in uncertain graphs.

### 6.2 Differences from Existing Work

The novel aspects of the proposed algorithms are as follows:

- **Core-based pruning techniques:** The proposed core-based pruning techniques are more effective than traditional core-based pruning techniques for reducing the size of the graph to be searched in uncertain graphs.
- **Improved enumeration algorithm:** The proposed enumeration algorithm is based on the Bron-Kerbosch algorithm, but it incorporates several new optimizations to improve its efficiency for uncertain graphs.
- **Cut-based optimization technique:** The proposed cut-based optimization technique further improves the pruning performance of the core-based pruning techniques.
- **Color-based upper bounding techniques:** The research focuses on utilizing two advanced color base bounding

rules in addition to a basic coloring scheme with a greedy approach to acquire tighter bounds while finding the maximum clique.

### 6.3 Effectiveness

The proposed algorithms are effective for solving the problem of maximal clique search in uncertain graphs because they:

- Reduce the size of the graph to be searched: The core-based pruning techniques and the cut-based optimization technique significantly reduce the size of the graph to be searched, which leads to significant performance improvements.
- Are specifically designed for uncertain graphs: The proposed algorithms are specifically designed for uncertain graphs, which makes them more efficient for real-world graph problems unlike the existing algorithms that are designed for deterministic graphs.

### 6.4 Mathematical Outlook

The proposed core-based pruning techniques can be analyzed using the following mathematical framework:

Let $G$ be an uncertain graph and $C$ be a maximal $(k, \tau)$-clique in $G$. A core of a clique $C$ is a subset of $C$ such that all pairs of vertices in the subset are adjacent with probability at least $\tau$.

The first core-based pruning technique works by removing from $G$ all vertices that are not contained in any core of a maximal $(k, \tau)$-clique. This pruning technique can be justified using the following theorem:

**Theorem:** If a vertex $v$ is not contained in any core of a maximal $(k, \tau)$-clique in $G$, then $v$ cannot be contained in any maximal $(k, \tau)$-clique in $G$.

The second core-based pruning technique works by removing from $G$ all vertices that are contained in a core of a maximal $(k, \tau)$-clique, but the probability of the core existing is less than $\tau$. This pruning technique can be justified using the following theorem:

**Theorem:** If the probability of a core of a maximal $(k, \tau)$-clique in $G$ is less than $\tau$, then the core cannot be contained in any maximal $(k, \tau)$-clique in $G$.

The proposed cut-based optimization technique works by partitioning the graph $G$ into a set of subgraphs and then pruning each subgraph using the core-based pruning techniques. The partitioning of the graph $G$ is performed using a cut-based algorithm, which is a type of graph partitioning algorithm that minimizes the number of edges between the subgraphs by discarding all the least promising edges.

The proposed enumeration algorithm is based on the Bron-Kerbosch algorithm, but it incorporates several new optimizations to improve its efficiency for uncertain graphs. For example, the proposed algorithm uses a new branching strategy and a new pruning strategy.

## 7. Technical Section

The proposed research introduces novel algorithms and methodologies distributed across three distinct phases within the proposed approach for addressing the problem statement associated with uncertain graphs. These methodological phases encompass the integration of Core-Based Pruning Algorithms, the enumeration of all maximal uncertain cliques, and the pursuit of maximum uncertain clique identification.

### 7.1 Algorithms

1. **Core-Based Pruning Algorithm followed by Cut-Based Optimization:** In the core-based algorithms, since cores in this context are subsets of vertices in a graph $G$, each vertex in the subset has to have a degree greater than the threshold $\tau$. The first core-based pruning algorithm is based on the $(k, \tau)$-core the state-of-the-art algorithm. We show that any maximal $(k, \tau)$-clique must be contained in the $(k, \tau)$-core. To efficiently implement this pruning rule, we devise a new dynamic programming (DP) based algorithm to compute the $(k, \tau)$-core. Compared to the existing algorithm, the new DP-based algorithm reduces the time complexity for computing the $(k, \tau)$-core from $O(md_{\max})$ to $O(m\delta)$, where $d_{\max}$ is the maximum degree of the nodes in the deterministic graph $\tilde{G}$ of $G$ and $\delta$ ( $\delta \leq d_{\max}$ ) is the degeneracy of $\tilde{G}$. The algorithms are referred to as DPCore and DPCore+ respectively. The DPCore makes use of a threshold $\tau$-degree (the largest integer $r$ that meets $P(r(du(G) \geq r)) \geq \tau$) to compute the $(k, \tau)$-core, while the DPCore+ makes use of an additional parameter $\tau$-core number (the largest $k$ such that there is a $(k, \tau)$-core containing $u$), a truncated version of threshold called truncated $\tau$-degree, and a top product probability metric all of which get rid of less promising nodes while generating the subgraphs ensuring a smaller subgraph which could be enumerated more efficiently. For further optimization, a cut-based approach is incorporated into the new DPCore+ algorithm suggested, the cut-based approach ensures that when a cut $\chi = (S, T)$ is made to generate a cut set $E_\chi$, all edges in $E_\chi$ can be deleted if there is no maximal $(k, \tau)$-clique subgraph containing the edges in $E_\chi$. Since the deletions of the edges in $E_\chi$ will partition $G_C$ into several small connected uncertain subgraphs, the computational costs for finding all maximal $(k, \tau)$-cliques can be significantly reduced.
2. **Enumerating all Maximal Uncertain Cliques:** A new maximal $(k, \tau)$-clique enumeration algorithm, called MUCE, is used by integrating the core-based pruning techniques that were stated earlier into the backtracking enumeration algorithm that already exists. Unlike in the backtracking enumeration algorithm, we also integrate the $(\text{Top}k, \tau)$-core pruning technique into the backtracking enumeration algorithm to prune unpromising search branches. Each connected part in the algorithm calls the backtracking enumeration algorithm MUCE to find all maximal $(k, \tau)$-cliques. MUCE admits five parameters $(R, C, X, k, \tau)$. $R$ denotes a $\tau$-clique which may be expanded to a maximal $(k, \tau)$-clique. $C$ is the set of candidate nodes that is used to expand the current $\tau$-clique $R$. $X$ denotes a set of nodes that can expand the current $\tau$-clique $R$, but have already been explored in a different search path by the algorithm. The algorithm first checks whether the current $\tau$-clique $R$ is a maximal $(k, \tau)$-clique or not. If so, the algorithm outputs $R$ and terminates the current search path. To avoid repeatedly enumerating the same maximal $(k, \tau)$-clique, the nodes in $C$ are selected following a lexicographical ordering. Also, the algorithm incorporates techniques from the existing systems to determine the set $X$ for the expanded $\tau$-clique $R'$ (line 20). Subsequently, the algorithm recursively calls the same procedure to expand the $\tau$-clique

$R'$ (line 21). After processing a node $u$, the algorithm adds it into $X$, because $u$ has already been processed in the current search path which cannot be explored in the following recursions.

3. **Maximum Uncertain Clique Search:** Specifically, the new approach keeps track of the size of the largest maximal $(k,\tau)$-clique $C^*$ found so far when obtaining a maximal $(k,\tau)$-clique, denoted by $\sigma$ (where $|C^*| = \sigma$). If the size of the candidate set $C$ is smaller than $\sigma - |R|$ (or $|R \cup C| < \sigma$), the algorithm can early terminate, because all maximal $(k,\tau)$-cliques in the current search subspace are no larger than $\sigma$. Since the upper bound using the candidate set $C$ is not as tight as expected, we could choose a color-based upper bound. Here, we assign a color to each node in $G$ using a degree-ordering based greedy coloring algorithm so that no two adjacent nodes have the same color. To optimize further, we could use an advanced approach. The basic color-based upper bound only considers the clique size constraint, which ignores the clique probability constraint. Here, we develop a tighter color-based upper bound based on both the clique size and clique probability constraints. This ensures a higher upper bound and a smaller clique set to enumerate at each step.

### 7.2 Analysis and Implementation

- **DPCore, DPCore+:** The *Improved Algorithms for Maximal Clique Search in Uncertain Networks* paper proposes two new core-based pruning techniques **(DP-Core & DP-Core++)** and a cut-based optimization technique to reduce the size of the graph to be searched, which is a key performance improvement for large uncertain graphs. State-of-the-art methods typically focus on optimizing existing algorithms, such as branch and bound and the Bron-Kerbosch algorithm. While DPCore prunes the nodes based on the graph size and edge probability, DPCore+ uses the degeneracy of the graph G and TopK product probability while pruning out the nodes. TopK product probability ensures that the higher product probability of the edges would ensure the existence of a maximal clique.
- **MUCE, MUCE+, MUCE++:** Two new algorithms MUCE+ and MUCE++ are implemented to the state of the art algorithm MUCE for improved Maximal Uncertain Clique Enumeration. MUCE+ uses the the $(k,\tau)$-core pruning rule in addition to the existing MUCE algorithm while MUCE++ integrates the the $(Topk,\tau)$-core pruning rule. Both of which optimize the required nodes to be enumerated through, resulting in efficient memory usage while working with huger uncertain graphs. Both MUCE+ and MUCE++ are integrated with the cut-based optimization technique to further optimize the final results. This approach ensures that no maximal clique is left behind while no additional resources are utilized to enumerate through unpromising nodes.
- **MUCE, MUCE+, MUCE++:** To address the retrieval of the maximum clique of all the cliques in an uncertain graph, the author introduces MaxUC and MaxUC+ in contrast to a general enumerating technique such as MaxRDS. The candidate-bounding and color-bounding techniques used in MaxUC and MaxUC+ respectively provide tighter upper bounds and smaller search spaces at each interation resulting in a faster retrieval of the maximum clique. While both the algorithms outperform the state-of-the-art algorithm MaxRDS, the color-based greedy bounding in MaxUC+ makes it quicker and more efficient than its counterparts.

The DP-based pruning algorithms proposed in the paper are very effective for pruning uncertain networks without missing any maximal cliques. The results of extensive experiments on six real-world datasets demonstrate that the proposed algorithms outperform the state-of-the-art algorithms by a significant margin.

The *Improved Algorithms for Maximal Clique Search in Uncertain Networks* paper develops new pruning techniques that are specifically designed for uncertain graphs. State-of-the-art methods typically apply optimizations that are more general-purpose, such as the Russian Doll Search framework and MaxSAT bounds.

## 8. Experiments

### 8.1 Tools and Resources:

**Software:**

1. **C++**: To perform Maximal Clique Search in networks using different algorithms and prune methods that are defined in the paper
2. **Python**: To visualize comparison of different algorithm and prune methods over graphs.
3. **System Requirements**: We performed our experiments on **Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz** with **64GB** RAM.

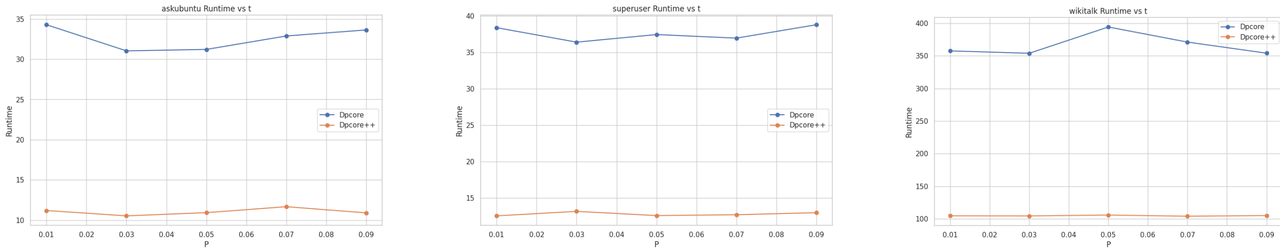
**Data Sets:**

1. **Ask ubuntu**: https://snap.stanford.edu/data/sx-askubuntu.html
2. **WikiTalk**: https://snap.stanford.edu/data/wiki-Talk.html
3. **Superuser**: https://snap.stanford.edu/data/sx-superuser.html

### 8.2 Experimental Setup:

A total of five experiments are performed, on three datasets, WikiTalk, AskUbuntu, SuperUser.

- **Exp1: Runtime of DPCore and DPCore+:**
  **Parameters:** varying values of k and $\tau$ against runtime. **Conclusions:** DPCore+ is much faster than DPCore on both WikiTalk, Askubuntu, and Superuser. Moreover, DPCore+ is more than one magnitude faster on all three datasets.
- **Exp2: Runtime of MUCE, MUCE+, and MUCE++:**
  **Parameters:** varying values of k and $\tau$ against runtime. **Conclusions:** MUCE++, along with MUCE+, exhibits significant runtime efficiency improvements compared to MUCE, on larger datasets, both MUCE++ and MUCE+ demonstrate runtime performance at least one order of magnitude faster than MUCE.
- **Exp3: Runtime of MaxUC, MaxRDS, and MaxUC+:**
  **Parameters:** varying values of k and $\tau$ against runtime. **Conclusions:** MaxUC+ outperforms competitors significantly, showcasing its efficiency in comparison to MaxRDS and MaxUC.
- **Exp4: Memory overhead:**
  **Parameters:** AskUbuntu, SuperUser, and WikiTalk




(a) Runtime of DPCore, DPCore+ on AskUbuntu.
(b) Runtime of DPCore, DPCore+ on SuperUser.
(c) Runtime of DPCore, DPCore+ on WikiTalk.

FIGURE 1. Exp1: Runtime of DPCore and DPCore+.

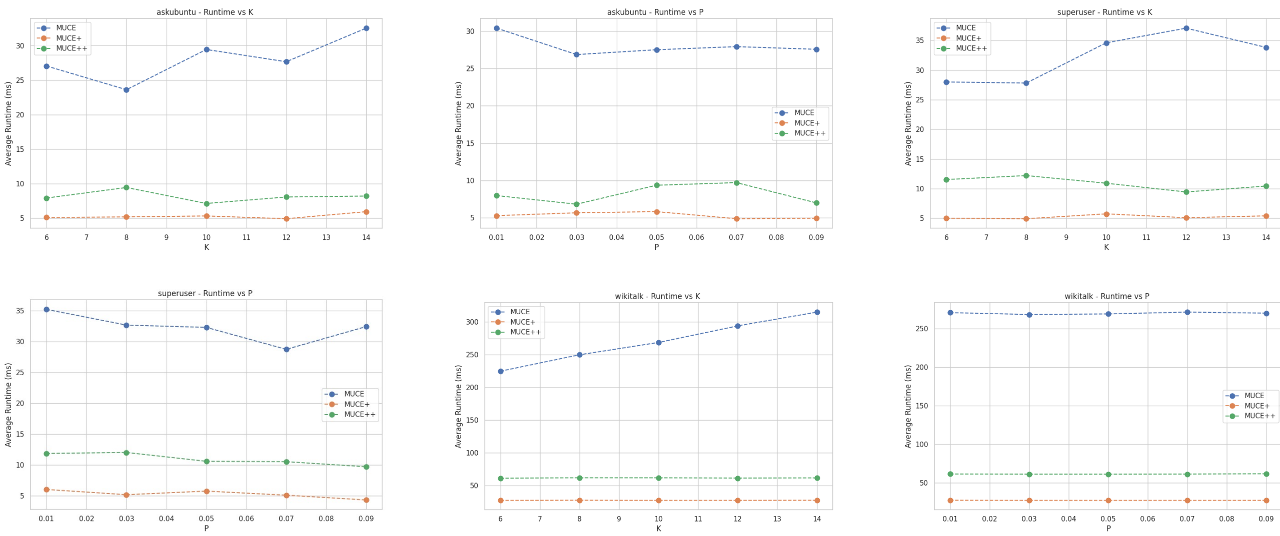



FIGURE 2. Exp2: Runtime of MUCE, MUCE+, and MUCE++ on AskUbuntu, SuperUser, WikiTalk with varying K and $\tau(P)$ respectively.

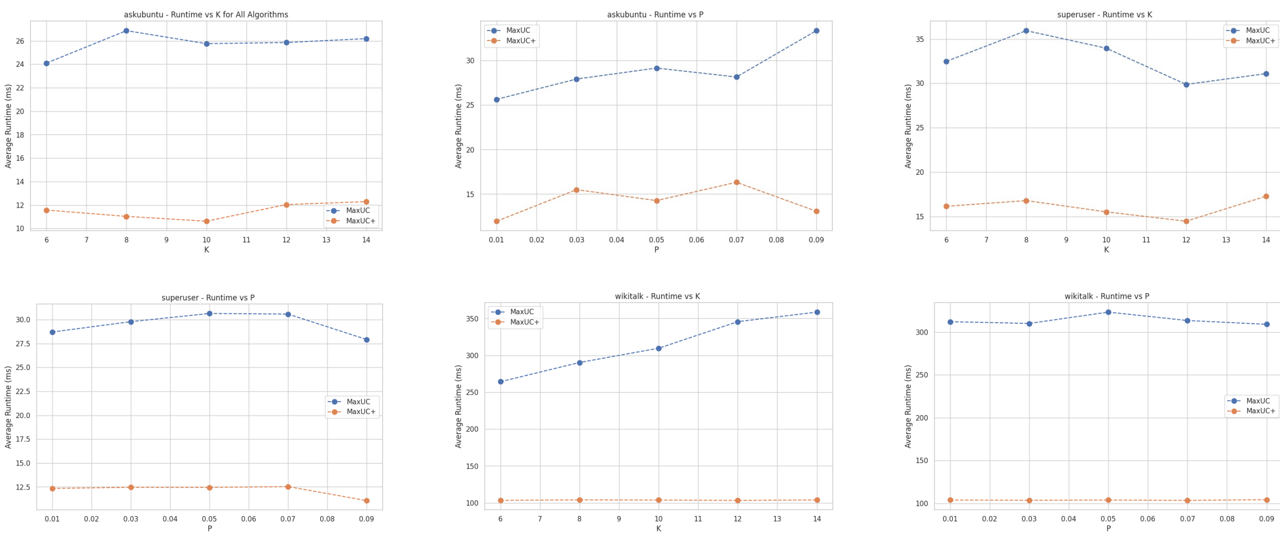
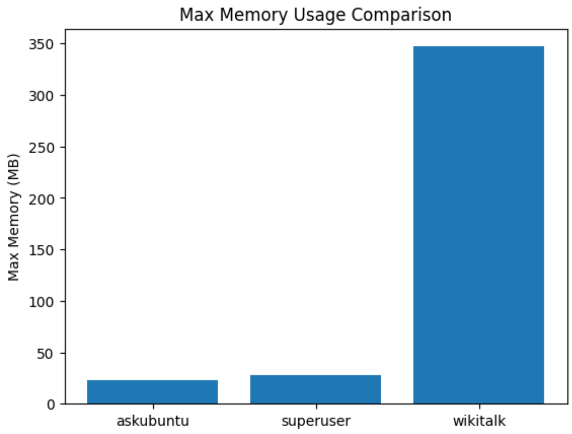
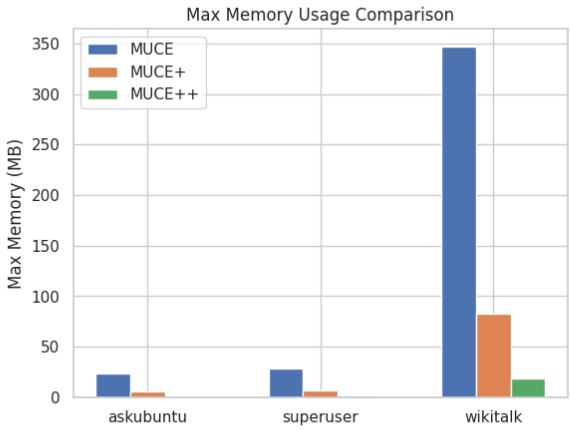



FIGURE 3. Exp3: Runtime of MaxUC, MaxRDS, and MaxUC+ on AskUbuntu, SuperUser, WikiTalk with varying K and $\tau(P)$ respectively.

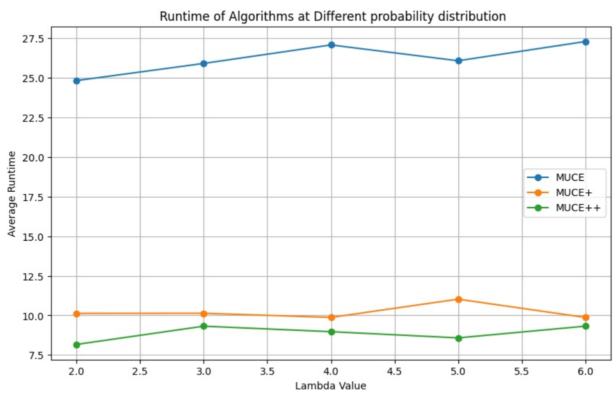



(a) Memory usage among the three datasets.

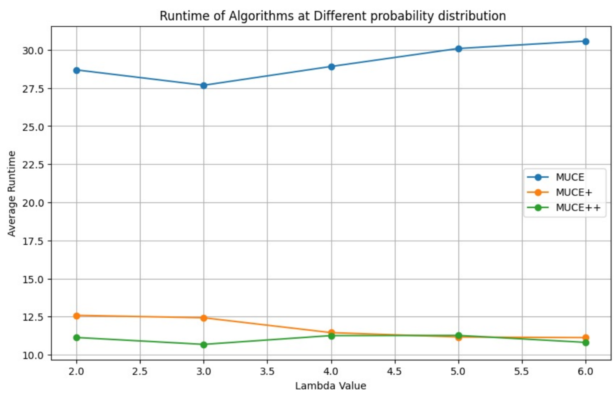



(b) Memory usage during clique enumeration using MUCE, MUCE+, MUCE++.

FIGURE 4. Exp4: Memory overhead




(a) Runtime of enumeration algorithms with varying values of $\lambda$ on AskUbuntu.




(b) CRuntime of enumeration algorithms with varying values of $\lambda$ on SuperUser.

FIGURE 5. Exp5: Effect of different probability distributions on Runtime.

against memory. **Conclusions:** Low memory usage reported in DPCore+ when compared to DPCore; MUCE++ when compared to MUCE+ and MUCE; MaxUC+ when compared to MaxUC and MaxRDS for all three datasets.

- **Exp5: Effect of different probability distributions on Runtime:**
  **Parameters:** Exponential distribution: varying from 2 to 6, Uniform distribution: in a range [0, 1] against time.
  **Conclusions:** Different trends in the runtime can be observed with the increase of $\lambda$ on AskUbuntu dataset, it can also be observed that the maximum clique search algorithms are robust with respect to different probability distributions in most cases.

### 8.3 Evaluation Methods:

This paper evaluates the proposed algorithms using three real-world datasets, which encompass social networks and communication networks. Subsequently, we compare the performance of the proposed algorithms with state-of-the-art counterparts on these datasets, considering runtime and memory efficiency as key metrics.

### 8.4 Performance Metrics

The paper evaluates the performance of the proposed algorithms in terms of time complexity, memory usage, and accuracy, while evaluating the precision of various algorithms with varying parameters. The paper also tests the effect of different probability distributions (i.e., varying $\lambda$).

### 8.5 Reported Results

The *Improved Algorithms for Maximal Clique Search in Uncertain Networks* paper shows that the proposed algorithms achieve significant performance improvements over state-of-the-art methods, especially for large uncertain graphs. State-of-the-art methods achieve good performance on a variety of datasets, but their scalability is limited for large uncertain graphs.

### 8.6 Experimental conclusions and revelations:

In our implementation of the project, we encountered a challenge regarding edge weights, as the datasets we utilized were unweighted. To address this, we adopted a strategy of assuming an initial edge weight of 1 and incremented it by 1 for each interaction to approximate the edge probabilities as suggested by the authors. This approach led to relatively uniform data, which in turn influenced our algorithmic results to show minimal variation. Nonetheless, this uniformity in edge weights does not contradict the efficacy of the pruning techniques we employed. Notably, the pruned algorithms exhibited significantly reduced runtimes, emphasizing the effectiveness of these techniques in enhancing computational efficiency. Additionally, it's important to mention that our results were obtained using advanced hardware, specifically an Intel Xeon Silver 4114 CPU operating at 2.20 GHz with 64GB RAM. This superior computing power likely contributed to the improved performance and efficiency observed in our experiments.

## 9. Conclusion

The proposed algorithms in the paper "Improved Algorithms for Maximal Clique Search in Uncertain Networks" are effective for solving the problem of maximal clique search in uncertain graphs because they reduce the size of the graph to be searched and are specifically designed for uncertain graphs. The proposed algorithms have been experimentally shown to achieve significant memory and runtime improvements over state-of-the-art methods, especially for large uncertain graphs.

## 10. References


1. E. Tomita, A. Tanaka, and H. Takahashi, “The worst-case time complexity for generating all maximal cliques and computational experiments,” *Theor. Comput. Sci.*, vol. 363, no. 1, pp. 2842, 2006.
2. J. Cheng, L. Zhu, Y. Ke, and S. Chu, “Fast algorithms for maximal clique enumeration with limited memory,” in *KDD*, 2012.
3. C. Bron and J. Kerbosch, Finding all cliques of an undirected graph (algorithm 457), Commun. ACM, vol. 16, no. 9, pp. 575576, 1973.
4. D. Eppstein, M. Loffler, and D. Strash, “Listing all maximal cliques in large sparse real-world graphs,” *ACM Journal of Experimental Algorithmics*, vol. 18, 2013.
5. T. Viard, C. Largeron, and C. Tresp, “Temporal clique mining in uncertain graphs,” in *KDD*, 2011.
6. R.-H. Li, D. Zhang, X. Huang, J. X. Yu, L. Qin, and W. Tian, “Mining maximal cliques in signed graphs,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 10, pp. 2466-2478, 2014.
7. J. Cheng, L. Zhu, and Y. Ke, “Parallel maximal clique enumeration for disk-resident graphs,” in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013, pp. 245-254.
8. Y. Miao, J. Pei, and G. Wang, “Mining maximal cliques in uncertain graphs,” in *Proceedings of the 2010 SIAM International Conference on Data Mining*, 2010, pp. 635-646.


## 11. Team work

- **Shaurya Tiwari:** Data extraction, cleaning, manipulation and input generation.
- **Aditya Sugandhi:** Pruning techniques implementation, Experimentation and analysis.
- **Jessica Neelam;** Enumeration and search algorithm implementation.

## 12. Source code

Here is a link to our GitHub repository: https://github.com/adityasugandhi/Improved-Algorithms-in-finding-maximal-and-maximum-clique-in-uncertain-networks/tree/main