# Weighted MUSE for Frequent Sub-graph Pattern Finding in Uncertain DBLP Data

Shawana Jamil, Azam Khan[1], Zahid Halim and A. Rauf Baig
Department of Computer
National University of Computer and Emerging Science, Islamabad Pakistan
{ i070849, zahid.halim, rauf.baig}@nu.edu.pk

[1] Preston University Islamabad Campus, Islamabad, Pakistan
azamafridi@yahoo.com

*Abstract*— **Studies shows that finding frequent sub-graphs in uncertain graphs database is an NP complete problem. Finding the frequency at which these sub-graphs occur in uncertain graph database is also computationally expensive. This paper focus on investigation of mining frequent sub-graph patterns in DBLP uncertain graph data using an approximation based method. The frequent sub-graph pattern mining problem is formalized by using the expected support measure. Here *n* approximate mining algorithm based Weighted MUSE, is proposed to discover possible frequent sub-graph patterns from uncertain graph data.**

## I. Introduction

With the explosive growth of digital data in every field of life, amount of data is increment at a very high rate. To extract or mine knowledge from these large amounts of data, data mining come forward. The main reason that data mining attracted a great attention of researchers in the information industry in recent years is the availability of huge amounts of data and the need of turning this data into useful information and to extract hidden knowledge. Data mining can be performed on all kinds of information repository. This includes relational databases, data warehouses, transactional databases, advanced database systems, protein and gene sequences data base, social networks, flat files and World Wide Web.

As data mining techniques and methods are increasingly applied to non traditional information repository, existing approaches for finding frequent item set cannot be used as they cannot model requirement of these domains. An alternative way to model these objects is to use graph for instance protein-protein interaction (PPI) can be well modeled via graphs. With graphs the problem of finding frequent item set / pattern is transformed into problem of discovering frequent sub-graph pattern. This study is conducted to find frequent subgraph pattern in uncertain graph data.

Graph mining has always been an area of great interest, but in last couple of years studies on graph mining are limited to exact graphs only, that are precise and complete. However, in real world graph data are generally uncertain due to noise, incompleteness and inaccuracies. These kinds of graphs are called uncertain graphs [1]. An uncertain graph essentially represents a probability distribution over all of the certain graphs in the forms of which the uncertain graph may actually exist [8]. Each of these certain graphs is called an implicated graph. Research study shows that finding frequent sub-graphs in an uncertain graphs database is an NP complete task. Finding frequency at which these sub-graphs occurs in uncertain graph database is also computationally expensive.

Mining in uncertain graph data is important in many applications. For example to predicts the membership of a new protein in a partially known protein complex by mining a Protien-Protien Interaction(PPI) network as an uncertain graph, [8] to model a wireless networks as an uncertain graph and extracts the most probable delivery sub-graph to coordinate with the design of routing protocols [8]. It can also be applied in the area of cheminformatics to find similarities between chemical compounds from their structural formula in order to identify new medicine. Another area of interest to mine uncertain graphs is in social networks. The success and explosive popularity of social networks makes it important to differentiate, compare them with other traditional and to develop appropriate tools to manage and maintain these Social networks effectively, as well as lead to the development of new communication models and behavioral theories in social sciences. [10]

In this paper our focus is on the frequent subgraph pattern finding in uncertain graph data sets of social networks. The remaining of the paper is organized as follows: Section 2 gives a survey of the previous work done on mining uncertain graphs databases and networks and summarizes recent papers that address frequent subgraph pattern identification in social networks graphs. Traditional definitions which can help in understanding of graph theory are in presented in Section 3. We follow MUSE algorithmic solution for finding frequent subgraphs in uncertain graph database [8]. The details of MUSE approach with some modifications are presented in section 4. Section 5 provides an experimental evaluation of MUSE algorithm on social network (DBLP dataset) data. Finally, Section 6 summarizes the paper and lists directions for future research.

## II. RELATED WORK

Graphs are a natural and universal representation across various disciplines and domains in which topological structures are involved. Large amount of data represented by graphs is known as graph data, and needs intelligent tools to analyze and understand them. Frequent subgraph mining [4] is one of the powerful tools to study the structures of this graph data. All traditional topological data structures, arrays, trees, vectors, can be represented as graphs. And uncertainties because of noise in accuracy or dynamic nature of topological structure make it an uncertain graph [1].

Recent research [6, 7, 8] has shown that promising number of algorithm have been developed for exact graph mining but little work has been done to find frequent sub-graph pattern in uncertain graph data. But in real life uncertainties are inherent in graph data, in particular, the structures of graphs are uncertain. A number of algorithms have been proposed for mining frequent subgraphs on certain graph data, which have been surveyed in [2] but all these algorithms can not handle uncertainties. In real practice uncertainties are inherent in graph data. However, some state-of-the art techniques are adopted by these algorithms such as minimum DFS codes [4] for representing subgraphs and the right-most extension technique [4] for extending subgraphs.

To the best of our knowledge [8] is the first who studied frequent subgraph mining on uncertain graph data independently. They proposed expected support to evaluate the significance of sub-graphs. In particular, the expected support of a sub-graph S is the expected value of the supports of S in all implicated graph databases.

Work in [6] proposes an approximate mining algorithm, called MUSE, to discover an approximate set of frequent sub graph patterns from an uncertain graph database. But this is an interval based algorithm in which Monte Carlo algorithm [3] is used to find interval for estimating expected support value to find frequent sub-graphs. Monte Carlo algorithm do has its own computational complexity. Their purposed approach focuses to find frequent sub-graph patterns in PPI dataset. But there approach works with the classic labeling of graphical data. The data set which they take is not much dynamic thus may provide different results in changing graph of social network like, Facebook, twitter and Myspace, which do change in time and people contact each other at specific time points and form various complex relationships.

In large social networks uncertainty arises for various reasons [11]. The probability of an edge may represent the uncertainty of a link prediction [12] or the influence of a person to another [14, 13]. Thus, in the perspective to social-network applications, we are interested to find people connected or influence the most.

Frequent sub-graph finding also helps in the dynamic social network. An approach proposed by [18], which find frequent subgraph in dynamic networks based on time intervals, where interactions between objects usually occur for a certain period of time only. Here uncertainties depend on link existence or removal is with respect to time.

In field of computer science frequent sub-graph discovery can help to find author and coauthor relationship in different research areas by mining the DBLP data. DBLP (Digital Bibliography & Library Project). DBLP server provides bibliographic information on major computer science journals and proceedings to analyze and predict the author and coauthor relationship, usability and worth of his/her research using citation analysis and the diversity of research of a specific author. [16]

The focus of our study is on finding frequent patterns to predict co-authorships in the DBLP based on historical data. We took DBLP post year 2007 data, from the DBLP database[1]. According to this database two authors are linked together if they have coauthored a journal or a conference paper. Probabilities on each link are also used to get feeling that the greater the weight of the edge more likely is to be present in the future.

It is observed that the evolution of graphs theory over time has been addressed predominantly dealt with topics in graph data mining to find sub-graph pattern in exact graphs, only a few papers [18, 19] define terminology for mining dynamic networks, but to the best of our knowledge so far no paper presents an efficient algorithm for detecting frequent sub graphs within uncertain graphs.

## III. BACKGROUND

In this section, we formally explain the frequent subgraph pattern finding problem and outline briefly some basic definition to understand this concept in case of uncertain data.

### *A. Problem Statement*

An uncertain graph in an uncertain graph database is a five characters touple G = ((V,E), Σ,L, P), where (V,E) is an undirected graph, Σ is a set of labels, $L : V \cup E \rightarrow \Sigma$ is a function assigning labels to vertices and edges, and $P : E \rightarrow (0, 1]$ is a function assigning existence possibility values to edges. The existence possibility P((u, v)), of an edge (u,v) means the possibility of the edge existing between the endpoints u and v in an exact graph instance. Specifically, P((u, v)) = 1 indicates that edge (u, v) definitely exists. Thus according to [1], an exact graph is a special uncertain graph with existence possibilities of 1 on all edges. Unlike an exact graph, an uncertain graph implicates a set of exact graphs.

Given an uncertain graph of social network like DBLP, the problem is to find all frequent subgraphs. As DBLP data is growing exponentially day by day researchers are often interested in identifying research areas of different authors, to predict co-authorship, and to identify usability and worth of his/her research using citation analysis.

It has been shown in [16] that DBLP data networks are generally subject to uncertainties and frequent subgraph pattern mining has been shown to be an effective approach in predicting co-authorships. It is important to find subgraph patterns that not only occur frequently in uncertain graphs but also have high degree of confidence in terms of uncertainty to exist in reality.

[1] http://dblp.uni-trier.de/

We map DBLP network graph as uncertain labeled graph then, let G = (V,E, Σ, P) be a probabilistic graph, where V denotes the author and E denote the link between two authors. Two authors are linked if they have coauthored a journal or a conference paper. In order to obtain a probabilistic graph, Σ is a set of labels of vertices and edges. The variable P denotes the probabilities associated with the edges; p(e) denotes the probability of edge e Є E. To generate all possible graph Let G' be a subgraph graph that is sampled from G according to the probabilities P, that is, each edge e Є E is selected to be an edge of G' with probability p(e). If EG' denotes the set of edges of G', then the probability associated with G is:

$$\Pr[G] = \prod_{e \varepsilon E_G} p(e) \prod_{e \varepsilon E / E_G} (1 - p(e))$$

Edge "e" can be described as a tuple of the form ($v_i$, $v_j$, p(e)) where $v_i$; $v_j$ Є V and p(e) Є[0,1].

So, the frequent subgraph pattern mining problem regarding DBLP can be stated as follows. Given an uncertain DBLP graph database D and an expected support threshold, and we have to find all frequent subgraph patterns by using isomorphism and embedding, in database D.

### B. *Formal Definitions*

A labeled graph G is a set of vertices V, in which a pair of vertices, u, v can be linked by edges E, and in which both vertices and edges may bear labels L. A graph $G_s$ = ($V_s$, $E_s$) is a sub-graph of G if $V_s$ Є V and $E_s$ Є E, denoted by $G_s$ Є G.

The graph isomorphism problem is the question whether there exists a bijection f between the nodes of two graphs $G_1(V_1,E_1)$ and $G_2(V,E_2)$ such that ($v_{1a}$ , $v_{1b}$ ) Є $E_1$ if and only if ($v_{2a}$ , $v_{2b}$ )Є$E_2$ where, $v_{2a}$ = f($v_{1a}$ ) and $v_{2b}$ = f($v_{1b}$ ). If $G_1$ is isomorphic to $G_2$, we refer to ($v_{1a}$ , $v_{1b}$ ) and ($v_{2a}$ , $v_{2b}$ ) as corresponding edges in the rest of this paper. It is unclear whether this problem is in NP or in P, and all attempts to classify it have failed so far.

Given two graphs $G_1$ = (V, $E_1$) and $G_2$ = ($V_2$, $E_2$), the sub-graph isomorphism problem consists in finding a sub-graph of $G_1$ that is isomorphic to $G_2$. This problem is known to be NP-Complete [9], if a graph $G_1$ is a sub-graph of graph G and isomorphic to another graph $G_3$, then $G_1$ is often referred to as an embedding of $G_3$ in G. $G_1$ is also a frequent sub-graph if it contains at least t embeddings of $G_3$, where MINSUP is a user-defined frequency threshold parameter. In application domains like bioinformatics, motif is often used as synonym for frequent sub-graph.

## IV. Methodology

### A. *MUSE*

We follow the MUSE (Mining Uncertain Subgraph Patterns) an approximation algorithm proposed by [6] with some of our modifications. As discovering all frequent subgraph patterns in an uncertain graph database is a very challenging problem. Finding expected support of an uncertain graph and then to find frequent subgraph is a NP complete problem [6, 15, 5]. It is NP complete task to examine all subgraphs to find the frequent ones. The hardness of this problem raised an opportunity to find an approximation algorithm MUSE. It is proposed to find an approximate set of frequent subgraph patterns in an uncertain graph database [6]. It finds set of all approximated frequent subgraph patterns in following manner.

Let $min_{sup}$ be the expected support threshold and ε Є [0, 1] be a relative error tolerance. MINSUP is a threshold value by which, all subgraph patterns with expected support at least $min_{sup}$ are output, but all subgraph patterns with expected support less than (1 − ε)$min_{sup}$ are not output. Moreover, decisions are for overlapping relationship with expected
support in between the [(1 − ε) · $min_{sup}$, $min_{sup}$) is done by approximated interval using FPRAS a Monte Carlo based method proposed by Karp n Luby [3]. Complete MUSE algorithm can be found in [6].

In the above mentioned approach exact algorithm is used to find expected support of a sub graph pattern. But this approach become exponential when embeddings are greater than 20, and also it evaluate expected support for every embedding having repeated edges with repeated weights more than once. So to avoid this repeated evaluation of expected support we propose weighted MUSE.

### B. *Weighted MUSE*

We have modified the MUSE by assigning weights factor w (0, 1) to the edges of embeddings includes in the identified frequent sub graph pattern.

WEIGHTED MUSE

```
input: An uncertain graph database D = {G1,G2, . . . , Gn}, a
threshold minsup ϵ [0, 1], a relative error tolerance ε ϵ [0, 1] and
a real number δ ϵ [0, 1]. Initially  Weight factor  of all edges is
wk=(1) where k is no of edges in subgraph
output: an approximate set of frequent subgraph patterns in D.
F ← ∅;
T ←{all subgraph patterns in D  with one edge};
while T = ∅ do
S  ←  pop(T)
   for i ← 1 to n do
      Find the subgraph isomorphisms from S to Gi;
         if  1>Σwk/Σe(S)>0.5

         Xi ←{all embeddings of S in Gi};
     [l, u]  ←  Approx-Exp-Sup(S, D, minsup, ε, δ, X1,X2,
   . . . , Xn);
   if l ≥ (1 − ε)minsup and u ≥ minsup then
      F ← F  {S};
      For all ek  in S decrease wk  by 0.1;
      Y ←{all direct superpatterns of S};
      foreach S'  Y do
         if Parent(S'_) = S then
          Push(S', T);
return F;
```

**Figure 1: Working of Weighted MUSE**

At every iteration if any embedding traversed earlier we decrease the weight factor to 0.1. By doing so we assign low priorities to the embeddings so that if next time any embedding with same edges an links weights occurs, exact algorithm do not waste time to evaluate the expected support as it already calculated previously. It will help to reduce computational cost and also in scanning of edges incident on the vertices while finding embeddings of S in the uncertain graphs, which also contributes in the finding of subgraph pattern to obtain all direct supergraphs of S.

If any graph embedding have an edge which is also a single edge graph in stack that is also present in the subgraph pattern founded earlier then we assign a low priority to it as this edge will be covered in the identified frequent subgraph as at line 11 of figure 1. And if the subgraph in the stack has low priority than a threshold value w= 0.5, as at line 6 of figure 1, on all of its edges than we prune them and do not traverse the path of it and do not put on stack for further traversing, and place this pattern in the list of frequent sub-graphs. This saves time and also by doing so exact algorithm can work with combinations of $P_r$ $(C_i)$ for more than 100 combinations.

## V. Experiments and Results

The weighted MUSE algorithm is implemented in C# and experiments were performed to evaluate the efficiency, approximation quality and scalability of Weighted MUSE in Comparisons to MUSE.

### A. Dataset

We have taken experiments using freely available real DBLP uncertain graph database. The DBLP was obtained from the DBLP.Uni-tier[2]. Post year 2007 data can be acquired from the DBLP website.[3]

This data set includes 135096 documents data, but we have take only 2000 documents from which we have make four partitions of documents having following specification. We

| Data set | \|V\| | \|E\| | unique directed links |
|---|---|---|---|
| dblp1 | 100 | 170 | 112 |
| dbpl2 | 150 | 213 | 159 |
| dblp3 | 200 | 294 | 182 |
| dblp4 | 330 | 384 | 346 |

**Figure 2: Summary of the uncertain graph database of DBLP**

have created a DBLP graph by considering an undirected edge between two authors if they have coauthored a journal paper. As DBLP is certain graph data base to use it in our work we make this data uncertain by introducing duplicate edges have identical nodes but with different edge probabilities.

2 http://dblp.uni-trier.de/xml/
3 http://dblp.mpi-inf.mpg.de/dblp-mirror/index.php

Assignment of probabilities on edges is described in the problem statement. DBLP has few probability distributions of edges and mostly edges have probability between 0.191 and 0.233 as shown the Figure 3.

### B. EFFICIENCY ANALYSIS of Weighted MUSE

This section lists the empirical assessment of the proposed method. The algorithm was coded in C#. All the experiments were run on windows XP with 1.3GHz Intel processors and 1GB of memory.

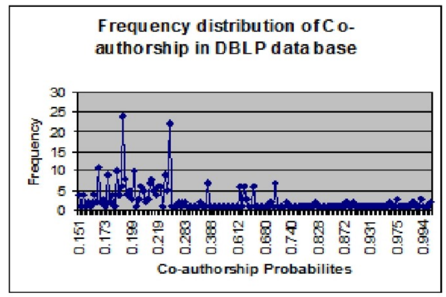


**Figure 3: Existence probability distribution of links in chosen DBLP dataset**

Time efficiency of Weighted MUSE on real DBLP graph database is evaluated with respect to the threshold $min_{sup}$ and the parameters ε and δ. It has been observed that weighted MUSE gives promising results as compare to original MUSE in terms of time.

Table 2 (Appendix-I) shows that the execution time of Weighted MUSE is much better than original MUSE. In weighted MUSE time decreases substantially with the increase in $min_{sup}$ and perform much better than MUSE. The reason of this behavior is increase in $min_{sup}$ threshold value. The number of output frequent sub-graphs patterns decreases quickly with increase in $Min_{sup}$. Same happens for all different sizes of dataset. Time complexity analysis of weighted MUSE and MUSE and results of no. of graph generated are summarizes in Figure 4 (a), (b) in Appendix-I respectively. The time difference between MUSE and weighted MUSE shows that our proposed approach gives good results in term of time. The reason of this is the weight assignment due to which the less computation need to calculate expected support. As far as no of generated frequent sub graphs are concern, it has been observed there is no big difference between the total numbers of generated subgraphs.

Table 3 (Appendix-I) shows the execution time of MUSE and weighted MUSE by Varying ε from 0.01 to 0.4, $min_{sup}$ = 0.3 and δ = 0.1. The change in no of generated subgraphs is not so significant for small data set having less no of frequent patterns in comparison to variation in $min_{sup}$. But as data set gets large, difference in time execution and no of generated frequent graph for both algorithms is significant.

The reason of this behavior is use of weighted MUSE. It works well on large data set and reduces search process by assigning priority through weight factor. As the data gets large the frequency of subgraphs also increases. Time complexity analysis and results of no. of graph generated for both algorithms are summarizes in Figure 5 (a), (b) (Appendix-I) respectively.

Table 4 (Appendix-I) shows the execution time of weighted calculate the time complexity of weighted MUSE. The variation in δ is from 0.01 to 0.3, with constant $min_{sup} = 0.3$ and ε = 0.1. The difference in elapsed time is not as dissimilar from previous case with constant δ, for smaller size of data set but as the dataset size increases, the execution time decreases rapidly with the increase in δ. This is because the time complexity of the weighted MUSE is proportional to ln(2/δ) which is the time complexity of exact algorithm. As our proposed approach utilizes exact algorithm by assigning weights and pruning of repetitive graph structures we are able to decrease time without any significant loss in frequent pattern detection. Time complexity analysis and results of no. of graph generated for both algorithm with varying δ are summarizes in Figure 6 (a) and (b) (Appendix-I) respectively.

### C. *ACURRACY ANALYSIS*

We follow the same approach for finding Accuracy with the variation Є and δ as used by [6]. We do follow precision and recall metrics. Precision is the percentage of true frequent subgraph patterns in the output sub-graph patterns. Recall is the percentage of returned sub-graph patterns in the true frequent subgraph patterns. Since, it is NP-hard to find all true frequent subgraph patterns [6]. We take the sub-graph patterns identified using $min_{sup}$. It is observed that the accuracy of weighted MUSE is 100% when dataset is small. When data set gets large increase in $min_{sup}$ gives false patterns as recalls are larger but the precision is constant.

## VI. CONCLUSION

This paper focus on the investigation of mining frequent subgraph patterns in DBLP uncertain graph data. The frequent subgraph pattern mining problem is formalized by using the expected support measure. An approximate mining algorithm based Weighted MUSE, is proposed to discover possible frequent subgraph patterns from uncertain graph data. The analysis and the experimental results show that Weighted MUSE has better efficiency as compare to MUSE in terms time complexity. Due to assignment of priority by assigning weight factor we get good results in minimum time s compare to original MUSE implementation..

## REFERENCES


[1] C. C. Aggarwal, Y. Li, J. Wang, and J. Wang, "Frequent pattern mining with uncertain data", In KDD, pages 29–38, 2009.

[2] C. C. Aggarwal and H. Wang. Managing and Mining Graph Data. Springer, 2010.

[3] R. M. Karp and M. Luby, "Monte-carlo algorithms for enumeration and reliability problems", In FOCS, pages 56–64, 1983

[4] X. Yan and J. Han. gSpan: Graph-based substructure pattern mining. In ICDM, pages 721–724, 2002.

[5] X. Yan, H. Cheng, J. Han, and P. S. Yu. Mining significant graph patterns by leap search. In SIGMOD, pages 433–444, 2008.

[6] Z. Zou, J. Li, H. Gao, and S. Zhang, "Frequent subgraph pattern mining on uncertain graph data", In Proceeding of the 18th ACM conference on information and knowledge management, CIKM '09, pages 583-592, 2009.

[7] Z. Zou, J. Li, H. Gao, and S. Zhang. Finding top-k maximal cliques in an uncertain graph. In ICDE, pages 649–652, 2010.

[8] Z. Zou, J. Li, H. Gao, and S. Zhang, "Discovering frequent subgraph over uncertain graph database under probabilistic sementics" , The 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'10), Washington, DC, USA, July 25-28, 2010, pages 633–642 .

[9] M. R. Garey and D. S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, 1979.

[10] David Jensen and Jennifer Neville "Data Mining in Social Networks".

[11] E. Adar and C. Re. Managing uncertainty in social networks. IEEE Data Eng. Bull., 30(2):15–22, 2007.

[12] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In CIKM, 2003.

[13] D. Kempe, J. M. Kleinberg, and ´E. Tardos. Maximizing the spread of influence through a social network. In KDD, 2003.

[14] D. Z. Wang, E. Michelakis, M. Garofalakis, and J. M. Hellerstein. Bayesstore: managing large, uncertain data repositories with probabilistic graphical models. Proc. VLDB Endow., 1(1):340–351, 2008.

[15] Valiant , The complexity of computing the permanent. Theory of ComputerScience, 8:189–201, 1979.

[16] M. Potamias, F. Bonchi, A. Gionis, G. Kollios, "KNN in uncertain graphs", In Proceedings of the VLDB Endowment, 36th International Conference on Very Large Data Bases, Vol. 3, No. 1, September 1317, 2010, Singapore.

[17] B. Wackersreuther , P. Wackersreuther , A. Oswald , "Frequent Subgraph Discovery in Dynamic Networks", In MLG '10 Washington, DC USA. 2010.

[18] K. M. Borgwardt, H.-P. Kriegel, and P. Wackersreuther. Pattern Mining in Frequent Dynamic Subgraphs. In ICDM, pages 818{822, 2006.

[19] M. Lahiri and T. Y. Berger-Wolf. Structure Prediction in Temporal Networks using Frequent Subgraphs. In CIDM, pages 35- 42, 2007.

## Appendix-I

| Weighted MUSE | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Data set | MinSUP=0.2 | | MinSUP=0.3 | | MinSUP=0.4 | | MinSUP=0.8 | |
| | Total SubGraph | Ellapsed Ttime | Total SubGraph | Ellapsed Ttime | Total SubGraph | Ellapsed Ttime | Total SubGraph | Ellapsed Ttime |
| dblp1 | 13 | 0:0:0:100 | 13 | 0:0:0:99 | 13 | 0:0:0:74 | 12 | 0:0:0:400 |
| dbpl2 | 23 | 0:0:0:120 | 23 | 0:0:0:88 | 23 | 0:0:0:76 | 22 | 0:0:0:40 |
| dblp3 | 38 | 0:0:0:220 | 38 | 0:0:0:186 | 38 | 0:0:0:100 | 26 | 0:0:0:70 |
| dblp4 | 34 | 0:0:0:515 | 19 | 0:0:0:320 | 15 | 0:0:0:228 | 2 | 0:0:0:100 |
| MUSE | | | | | | | | |
| dblp1 | 13 | 0:0:0:15 | 13 | 0:0:0:16 | 13 | 0:0:0:15 | 12 | 0:0:0:15 |
| dbpl2 | 23 | 0:0:0:80 | 23 | 0:0:0:30 | 23 | 0:0:0:30 | 22 | 0:0:0:30 |
| dblp3 | 38 | 0:0:0:80 | 38 | 0:0:0:70 | 38 | 0:0:0:60 | 26 | 0:0:0:60 |
| dblp4 | 33 | 0:0:0:1000 | 15 | 0:0:0:680 | 15 | 0:0:0:660 | 2 | 0:0:0:150 |

**Table 1: Results with Relative error tolerance Є=0.1,andReal number δ= 0.3**

| Weighted MUSE | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Data set | Є=0.01 | | Є=0.2 | | Є=0.3 | | Є=0.4 | |
| | Total SubGraph | Ellapsed Ttime | Total SubGraph | Ellapsed Ttime | Total SubGraph | Ellapsed Ttime | Total SubGraph | Ellapsed Ttime |
| dblp1 | 13 | 0:0:0:100 | 13 | 0:0:0:100 | 13 | 0:0:0:40 | 13 | 0:0:0:40 |
| dbpl2 | 23 | 0:0:0:150 | 23 | 0:0:0:99 | 23 | 0:0:0:20 | 23 | 0:0:0:20 |
| dblp3 | 39 | 0:0:0:215 | 39 | 0:0:0:200 | 40 | 0:0:0:40 | 40 | 0:0:0:50 |
| dblp4 | 19 | 0:0:0:778 | 20 | 0:0:0:525 | 22 | 0:0:0:80 | 34 | 0:0:0:80 |
| MUSE | | | | | | | | |
| dblp1 | 13 | 0:0:0:150 | 13 | 0:0:0:155 | 13 | 0:0:0:60 | 13 | 0:0:0:60 |
| dbpl2 | 23 | 0:0:0:156 | 23 | 0:0:0:145 | 23 | 0:0:0:60 | 23 | 0:0:0:60 |
| dblp3 | 39 | 0:0:0:248 | 39 | 0:0:0:225 | 38 | 0:0:0:60 | 38 | 0:0:0:70 |
| dblp4 | 19 | 0:0:0:825 | 19 | 0:0:0:676 | 20 | 0:0:0:105 | 34 | 0:0:0:115 |

**Table 2: Results with MINSUP =0.3, Relative error tolerance and Real number δ= 0.1**

| Weighted MUSE | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Data set | Real Number δ=0.01 | | Real Number δ=0.2 | | Real Number δ=0.3 | | Real Number δ=0.4 | |
| | Total | Ellapsed | Total | Ellapsed | Total | Ellapsed | Total | Ellapsed |
| | SubGraph | Time | SubGraph | Ttime | SubGraph | Time | SubGraph | Time |
| dblp1 | 13 | 0:0:0:20 | 13 | 0:0:0:20 | 13 | 0:0:0:20 | 13 | 0:0:0:20 |
| dbpl2 | 23 | 0:0:0:20 | 23 | 0:0:0:20 | 23 | 0:0:0:30 | 23 | 0:0:0:30 |
| dblp3 | 39 | 0:0:0:10 | 39 | 0:0:0:30 | 39 | 0:0:0:40 | 39 | 0:0:0:40 |
| dblp4 | 23 | 0:0:0:150 | 19 | 0:0:0:90 | 19 | 0:0:0:90 | 19 | 0:0:0:90 |
| MUSE | | | | | | | | |
| dblp1 | 13 | 0:0:0:25 | 13 | 0:0:0:27 | 13 | 0:0:0:30 | 13 | 0:0:0:40 |
| dbpl2 | 23 | 0:0:0:55 | 23 | 0:0:0:60 | 23 | 0:0:0:76 | 23 | 0:0:0:80 |
| dblp3 | 39 | 0:0:0:20 | 39 | 0:0:0:35 | 39 | 0:0:0:47 | 39 | 0:0:0:60 |
| dblp4 | 23 | 0:0:0:220 | 19 | 0:0:0:180 | 19 | 0:0:0:170 | 19 | 0:0:0:190 |

**Table 3: Results with minSUP = 0.3 and Real number 0.1**

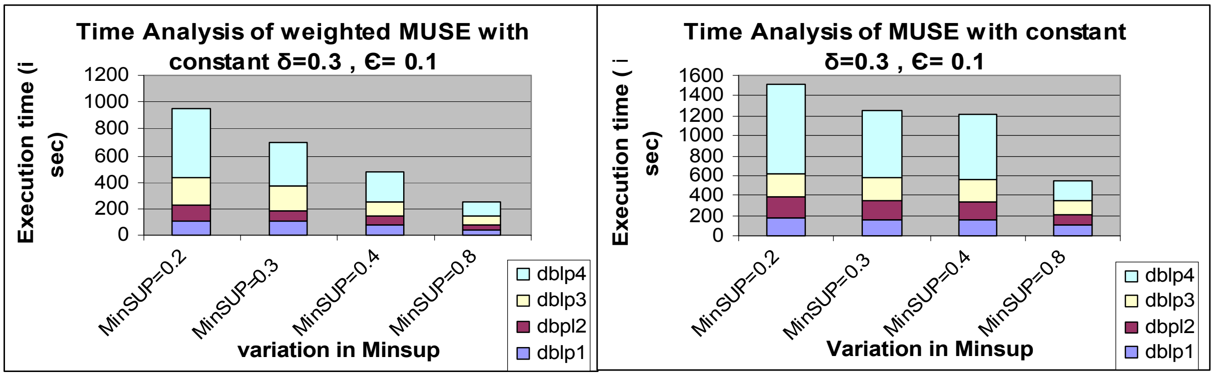


**Figure 4 (a): Time Analysis of Weighted MUSE and orignal MUSE with constant δ=0.3 , Є= 0.1**

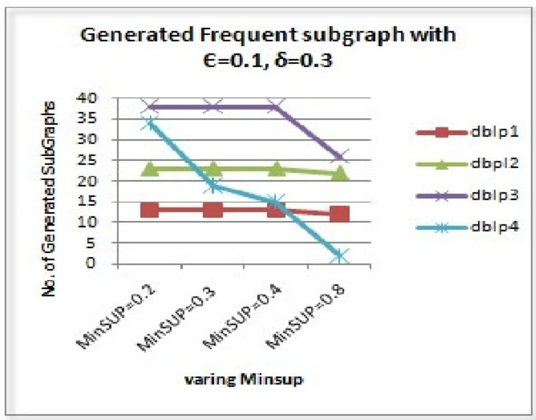


**Figure 4 (b): No of generated frequent sub-graphs with constant δ=0.3 , Є= 0.1**

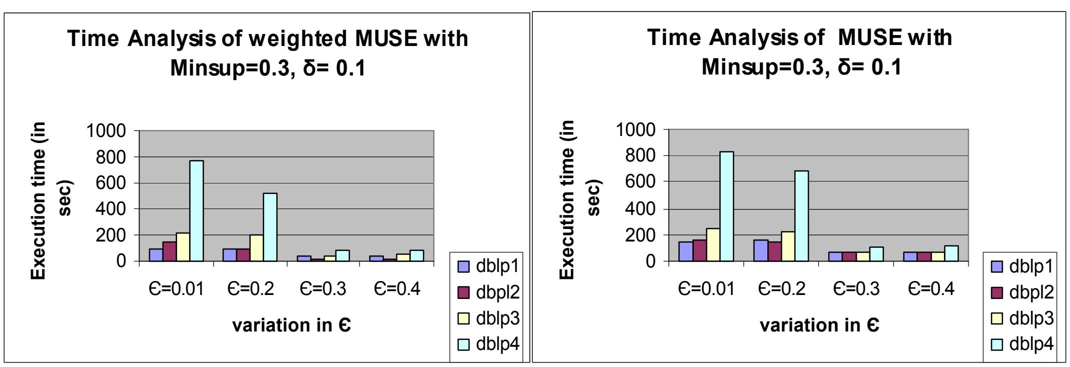


**Figure 5 (a): Time complexity analysis of Weighted MUSE and orignal MUSE with constant minsup=0.3 δ= 0.1 and varying Є from 0.01 to 0.4**

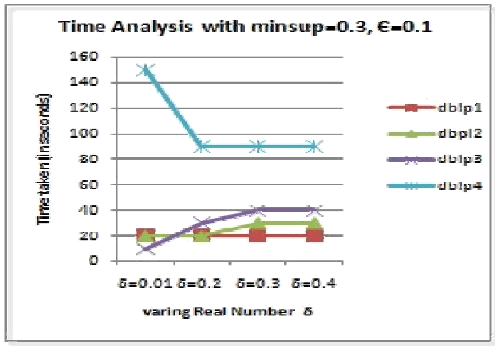


Figure: 5 (b) . No of generated frequent sub-graphs with constant $min_{sup}$=0.2 , Є= 0.1 and varying Є from 0.01 to 0.4

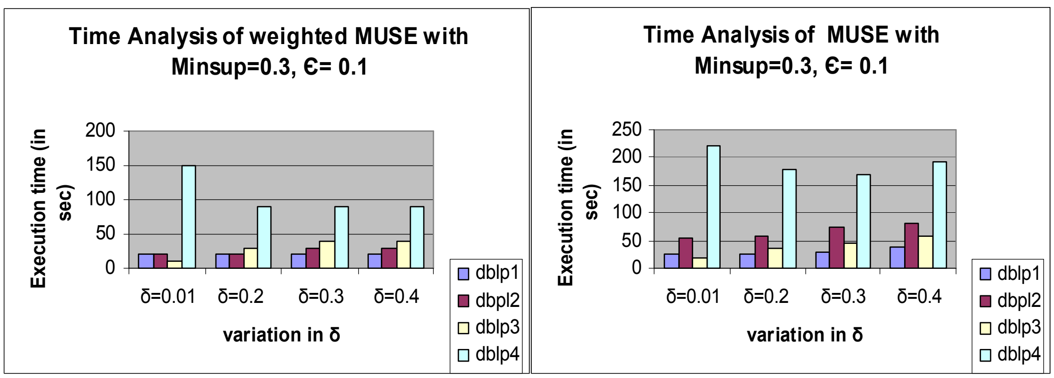


Figure 6 (a): Time complexity analysis of Weighted MUSE and orignal MUSE with constant minsup=0.3 δ= 0.1 and varying Є from 0.01 to 0.4

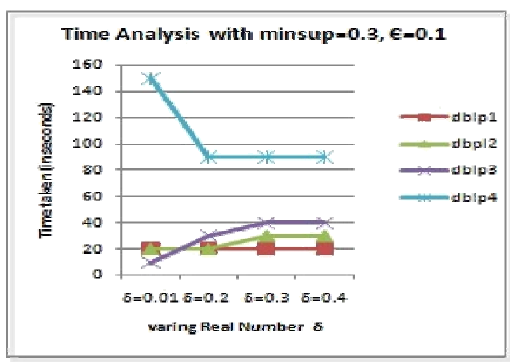


Figure: 6 (b) . No of generated frequent sub-graphs with constant $min_{sup}$=0.2 , Є= 0.1 and varying δ from 0.01 to 0.4