International Journal of Database Theory and Application
Vol.8, No.6 (2015), pp.323-332
http://dx.doi.org/10.14257/ijdta.2015.8.6.29

Implementation of Six Degrees of Seperation Using Two Dimensional Bi-direction Search to Connect People

Shubham Kumar, Shivank Sharma and N. Ch. S. N. Iyengar

School of Computing Science and Engineering, Vellore Institute of Technology University, Vellore-632014, Tamil Nadu, India
Shubham_kmr@yahoo.com, shivank072@gmail.com, nchsniyr@vit.ac.in

Abstract

Six Degrees of Separation is a theory that has recently been popularized due to the emergence of various social networking platforms. This theory states that any two random people in the world can be associated with each other with no more than six intermediate links. Though this theory has not been verified yet it has a great potential for practical implementation. In this paper we use this theory to create an application which is going to help the user in finding other individuals who share some common links and can be of benefit to the user. This is achieved through maintaining a central database which contains the details of all the users using our application. A user can search for other people on our database based on certain parameters which would then display the resultant links between the user and the selected target. This establishing of links is achieved using two dimensional bi-direction search algorithm.

Keywords: *Six degrees of separation, social networking platforms, two dimensional bi-direction search.*

1. Introduction

The idea of Six Degrees of Separation was first proposed by Frigyes Karinthy in 1929 as a work of fiction, in his short story Chains (Láncszemek). This theory proposed the idea that any two random people in the world can be connected to each other in no more than six intermediate steps. A step being a connection between two familiar individuals. As this theory gained popularity a lot of attempts have been made over the years to prove the same. The experiments are usually governed by a lot of dynamic factors like diversity of users, size, complexity, structure, depth, connectivity, and various other elements depending on the nature of the test network chosen. On an average the degree of separation found through various experiments ranges from 5.89 to 7.12.

In this paper we use the theory of six degrees of separation to present the idea for an application using which people can easily find other individuals who might belong to a particular organization, city or country. The main idea behind the development of this desktop application is bi-fold. First it will provide the average degrees of separation between each node in the database that will be calculated and updated regularly whenever a new link is searched for. This will give us a good estimate as to the validity of the theory. The second use of this application would mainly be to help people find others who might prove to be of benefit to them. For example a student from India planning on going abroad to a university for higher education can search for other people from his country who might already be studying in that particular university and get a firsthand account to all his queries. The application will also provide links as to how that user is connected to the search results if such a link within six degrees exists, which if nothing can turn out to be a fun link finding game to say the least, similar to the game six degrees of Kevin Bacon [1].

ISSN: 2005-4270 IJDTA

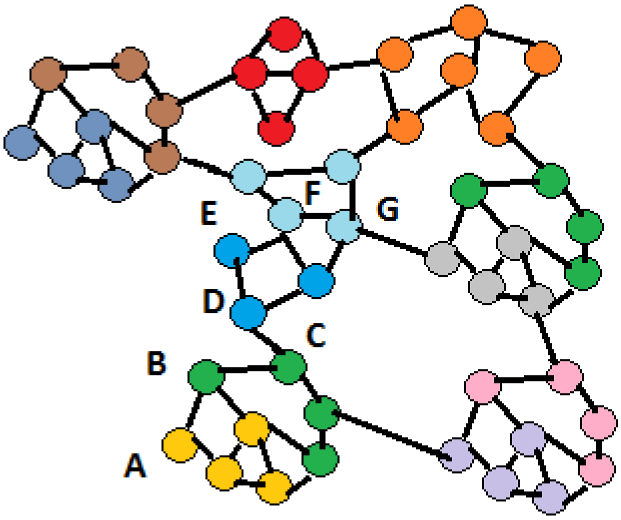


Figure 1. Six Degrees of Separation

The computing logic used in our application makes use of two dimensional bi-direction search algorithm. There are various reasons as to why we chose this algorithm some of them being that the search domain for the establishment of the links is very huge and the number of comparisons that are needed to be made grow exponentially as the depth of the links increase. Also there is no proper decision making process at junctions that eliminate a considerable amount of nodes. These factors lead to the need of large amount of computations and transactions. As we are planning to release our application for the general public we have decided to stick with a single algorithm to avoid unexpected exceptions and maintain consistency. The best algorithm that suits our case is two dimensional bi-direction search which usually performs better than other search algorithms like Depth first search, Best first search and Breadth first search, although this might not always be true.

The rest of this paper is structured as follows. In Section 2, we go on to describe the related works in the field of six degrees of separation. In Section 3, we explain our problem statement and the work that we are presenting in this paper. Moving further in Section 4, we talk about the design and implementation of our application. In Section 5, we talk about the experimental results that we obtained followed by the conclusion in Section 6. Finally in Section 7, we end this paper with the future work that we hope to perform in this field.

2. Related Works

In a study the validity of six degrees of separation was tested by making use of Facebook users and utilizing ant colony optimization. Their experiment showed an average separation of 5 degrees between their test subjects [2]. An instant messaging system was established to provide online-chatting services for people so as to collect their social relationship information. Routing algorithms and damping factors were applied to the system so as to prove the theory's validity [3]. Another experiment performed by [4] made use of new search techniques to obtain optimal solutions to prove the validity of six degrees of separation on twitter.

Mathematical modeling and online measurements were used to validate the application of six degrees of separation in online societies. The minimum diameter problem in graph theory is applied to calculate the maximum and average number of connections between any two online users [5]. In another experiment by [6] the use of complex network analysis was made to obtain the collaboration pattern at research, institute and state levels by forming corresponding networks based on the number of grants collaborated.

Facebook [7], the popular social networking website had an application named “Six Degrees” which was developed by Karl Bunyan. The average separation between all the users was 5.73 and the maximum was found to be 12. The application was however shut down in June 2009 due to Facebook’s caching policy issues which allowed sharing its resources for only 24 hours thus causing appreciable variations in the calculated degrees of separation. Facebook itself released an in depth analysis using probabilistic algorithms which concluded that out of 729 million users with 69 billion friendship links the degree of separation between any two user was a mere 4.74.

SixDegrees.com [8] was a first of its kind social networking site that was functional between 1997 and 2001. It had almost 3,500,000 registers users who were connected through personal acquaintances, friends, family members etc. up to 3 degrees of separation. It eventually became dysfunctional in the late 2000 as the concept of the site did not attract much public attention.

LinkedIn [9], the statistically designed corporate centered social networking site partially makes use of the six degrees of separation and displays the links or connections up to 3 degrees. This site enables the user to search professionals based upon the desired characteristics and communicate with them. The first step of communication is getting connected to a strong link which is the 1st connection. This is the immediate friend of the user. The weak links or the 2nd and 3rd connections are then established when the user enters their network. This enhances the ease of communication and builds a strong professional network of the person.

A lot of application are present online which use the concept of six degrees of separation in different ways, however we have not found any of them to be working in a manner similar to our application which tries to search for people based on input parameters and provide the corresponding six degree link between the user and the search results.

3. Two Dimensional Bi-direction Search

Since in our application we are dealing with a very large search domain, our main objective while selecting the algorithm was to minimize the time taken to establish a valid link from the user to the target. The time taken to establish a link can depend on various factors like the depth of the target, connectivity of the nodes, complexity of the domain, diversity and various other dynamic factors.

Two dimensional bi-direction search is capable of moving in both the direction in the search domain, which means we can search for links from the user to the target and also from the target to the user simultaneously. This is beneficial for us as it helps to reduce the depth that each search side has to visit.

The algorithm can be simplified further and broken down into two sub algorithms executing for a single iteration alternately. This means that the whole program can run on a single thread. The division can be described as a non recursive depth first search algorithm initiating from the user and a non recursive breadth first search initiating from the target. Therefore since the algorithm as a whole is moving along two axes, hence the name two dimensional search.

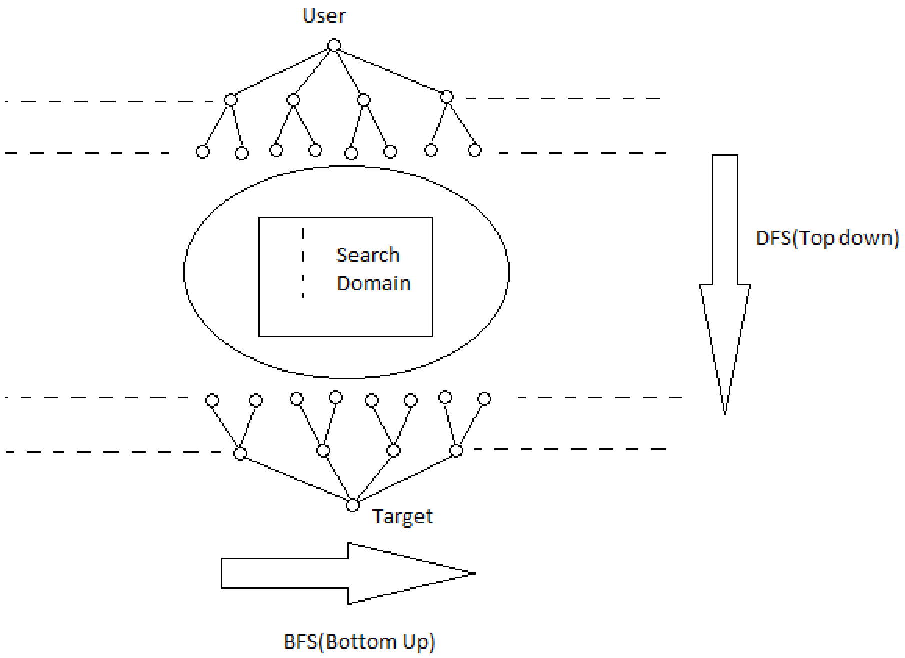


Figure 2. Two dimensional bi-direction search

Now since the application is an implementation of six degrees of separation each search side is limited to a maximum depth of six levels. Since the maximum depth is six levels and the size of the search domain is vast, obtaining valid links between the user and the target can take very long when the target is placed at greater depths. This is the reason we have introduce two hash tables that keeps record of all the nodes examined by both the search directions. Whenever a new node is visited by either of the search direction it is compared to the nodes present in the hash tables. An element known as a pivot keeps track of all the nodes inside the hash tables of both the search sides and returns a concatenated valid link whenever there is a common node found in both the hash tables. A valid link will also be returned if either side reaches its target because both the source and target are included in their corresponding hash tables.

Even after all these attempts to optimize the link finding process, it is frequently observed that the final link obtained may not be optimal. This means that the degrees of separation between the user and the target found may not be the smallest possible link in the search domain. This is one of the most common situations that arise while executing our algorithm since it only searches for a single link. To prevent this we implemented the concepts of heuristic ranking. This is a concept which helps the algorithm to decide which path to take out of the several paths that extend from a junction node. Since our database stores the information about the users as attributes such as the organization they work in, their state, their current city and their country. We decided to prioritize the path taken at a node junction based on similar attributes between the current node and the target node in the following manner:

Priority: country>organization>current city>state

There is no concrete evidence that proves the validity of this ranking scheme, but it was found to yield faster links in most of the cases that we compared. This method of heuristic ranking even though it helps to reduce expansion of nodes, makes use of transactions with the database at each new node the algorithm visits, which sometimes increases the time taken to establish the links. Also since the search domain is very large, similarity in attributes can only be expected at greater depths near the end of the links. Therefore all the heuristic transactions made near the beginning of the links at shallow depths usually just end up increasing the total time needed to establish a link from the user to the target.

Also the extent to which the search domain is connected cannot be determined beforehand. This complex connectivity often leads to the formation of cycles which are invalid links between the user and the target and should be ignored. This is achieved by maintaining a **look-back** table which keeps the record of all the nodes visited for the present link by the search algorithm. These tables are very similar to the hash tables and help us to check any new node the algorithm visits against the nodes in the look-back table. If any new node matches a node in the look-back table it is ignored and therefore the algorithm prevents the formation of a cycle. Preventing the formation of cycles is very important as it leads to a significant increase in the time taken to find the links and also amounts to unnecessary increase in the degrees of separation between the user and the target which should be minimized at all cost.

4. Design and Implementation

The designing of our application is done using java on the front end and MySQL as the database which works at the backend. For the purpose of testing and obtaining results we asked for volunteers who would use our application to register themselves and add their friends on their profile. If their friends were not present on our database we created a small interface which would easily let them enter the details of their friends who would then become available on our database as well. At the beginning of the test of this application we had 305 volunteers who agreed to use the application and populate our database. The total population of our database rose to 515 through indirect additions by the volunteers. The average number of friends for each user during the experiment was 23.

Our application aims to achieve two objectives, first to establish links between the user and the target and second to count the average degrees of separation between each user in our database which would be updated dynamically whenever someone searches for links. To achieve these objectives we could either obtain a single link between the user and the target or we could let the algorithm run for as long as it takes to find all the links. The decision to go with the latter would give us an accurate value for the average degrees of separation between all nodes in our database. This is because by finding all the links we will be able to choose the smallest degree of separation between any two nodes and use it to find the average value for the whole database. But searching in this manner would take very long execution time to complete a single request from a user. This time was sometimes found to exceed 30 minutes which is not acceptable as no user will wait for that long to get an output from the application. Therefore we decided to stop the execution of the algorithm as soon as a link was found. This would mean that the search time would come down as acceptable and the value for the average degrees of separation would have to be compromised.

Now as stated earlier our application is also dedicated to finding people in our database based on entered attributes like organization or current city. A user can easily find people belonging to a particular university, organization or town and see how they our connected to them. The need for doing this could be numerous as the user might not directly be in touch with the target but might need their assistance or the target might be the right person to answer the user's questions. Our application is designed to satisfy these criteria where we allow the user to search our database based on certain specific attributes like organization and then try to connect the user with the selected search results, if such a link between them exists within six degrees of separation.

The algorithm used for our application is given below.

4.1. Algorithm

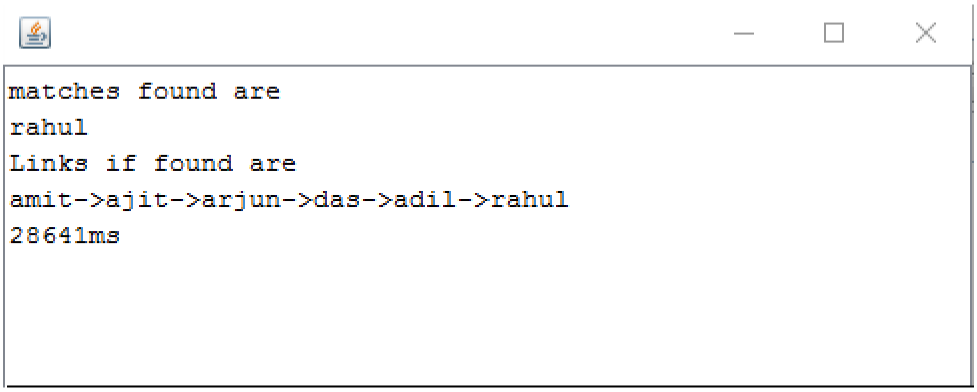
```
    Input user,target;
    key=1;permit_1=1;permit_2=1;
    next_node_1=user;
    next_node_2=target;
    while(link not found and result set not empty)
    {
    while(key=1){
    next_node_1=dfs(permit_1,next_node_1);
    next_node_1=look_back_1(next_node_1);
    hash_table_1(next_node_1);
    if(pivot){
            concat(pivot);
            end;
    }
    key=0;
    }
    while(key=0){
    next_node_2=bfs(permit_2,next_node_2);
    next_node_2=look_back_2(next_node_2);
    hash_table_2(next_node_2);
    if(pivot){
            concat(pivot);
            end;
    }

    key=1;
    }
}
```

5. Experimental Results

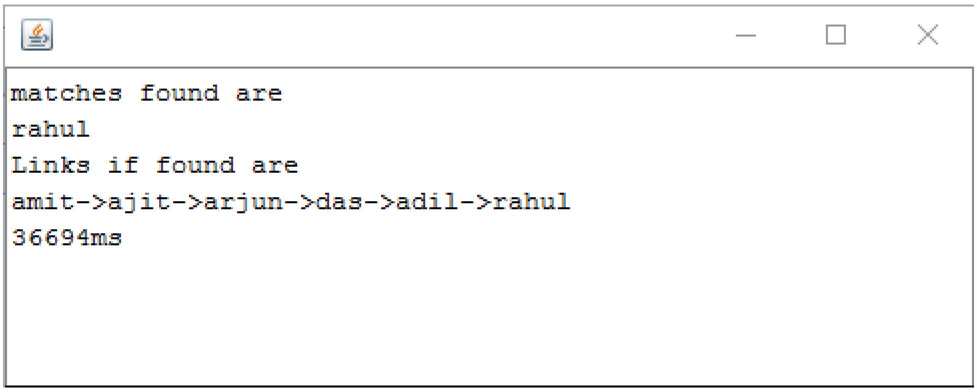
To obtain the experimental results we decided to find the links between every node in our database. This would provide us an average value of the degrees of separation that holds for our database and also give us the performance parameters of our algorithm like average runtime for various degrees, average nodes visited for various degrees and average nodes rejected for various degrees.

The use of heuristic priority and look-back tables in our algorithm has to a considerable extent decreased the search time taken to find the link between the user and the target. To demonstrate this we have taken a case in which the degree of separation extends up till the 6th degree and calculated the execution time. It was found that the algorithm without heuristic priority fetched a runtime 28.11% more than when the algorithm was executed with heuristic priority. The output of the test case is shown below in Figure 3 and 4.



```
matches found are
rahul
Links if found are
amit->ajit->arjun->das->adil->rahul
28641ms
```

Figure 3. Test Execution of the Algorithm with Heuristic priority



```
matches found are
rahul
Links if found are
amit->ajit->arjun->das->adil->rahul
36694ms
```

Figure 4. Test Execution of the Algorithm without Heuristic Priority

Figure 5 gives us the average time taken to find the links between the nodes at various degrees of separation. As expected, for degrees of separations 1, 2, 3 and 4 the time taken is minimal. The time taken becomes noticeable from the 5th degree where it reaches an average value of 100 seconds. It increases further more for the 6th and the last degree of separation by approximately 3 times to a value of 315 seconds. This is in accordance to what was expected as the number of transactions requested by heuristic priority increases as the number of nodes increase.

Figure 6 gives us an estimate about the average number of nodes visited by the algorithm while finding the links between the nodes for various degrees of separation. A significant increase can be observed for the next degree of separation as the number of nodes that could be potential targets increases exponentially at every level.

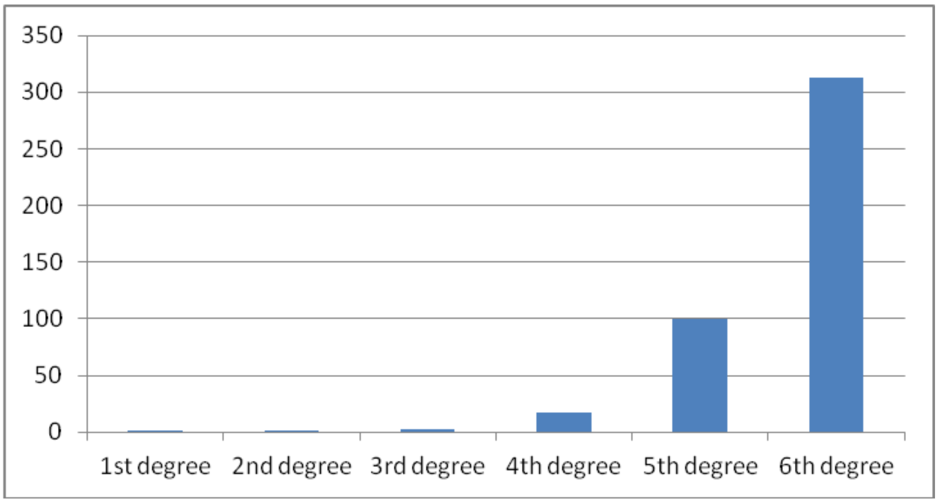


Figure 5. Average Time Taken to Find the Links for Various Degrees of Separation

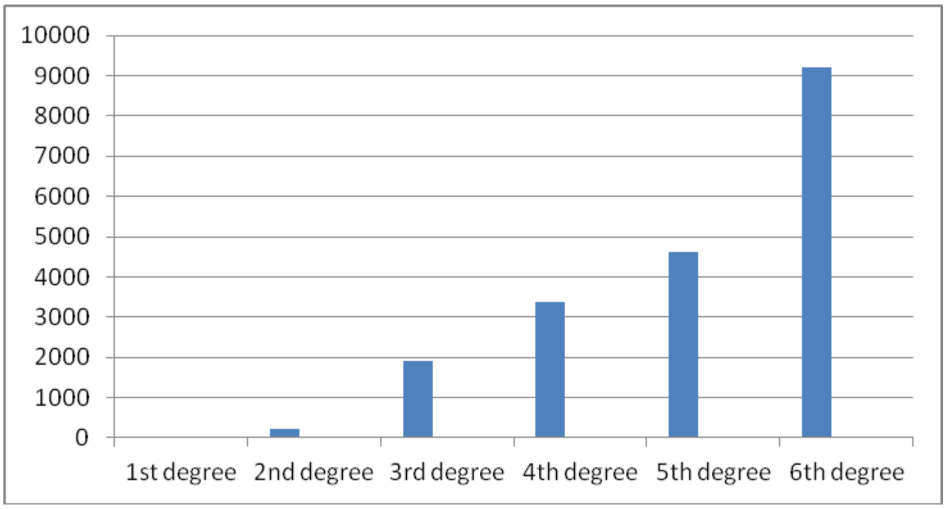


Figure 6. Average Nodes Visited to Ffind the Links for Various Degrees of Separation

The next graph shown below in Figure 7 gives us an average estimate of the number of nodes that the algorithm rejects at each degree of separation. The values for 1st and 2nd degree are negligible as expected since almost all the nodes are a potential target. At higher levels we can observe that the use of heuristic priority helps to eliminate a lot of nodes which could not have possibly led to the target. The use of look-back tables also come to play here as they eliminate all the nodes that could have led to the formation of cycles.

The experimental values we received look quite optimistic but we also have to keep in mind that our search domain was restricted to a small database. In situations when we are dealing with a larger database we might have to optimize our algorithm further to get similar results. The average value for the degree of separation for our database was found to be 5.21 which is in accordance with the theory of six degrees of separation.

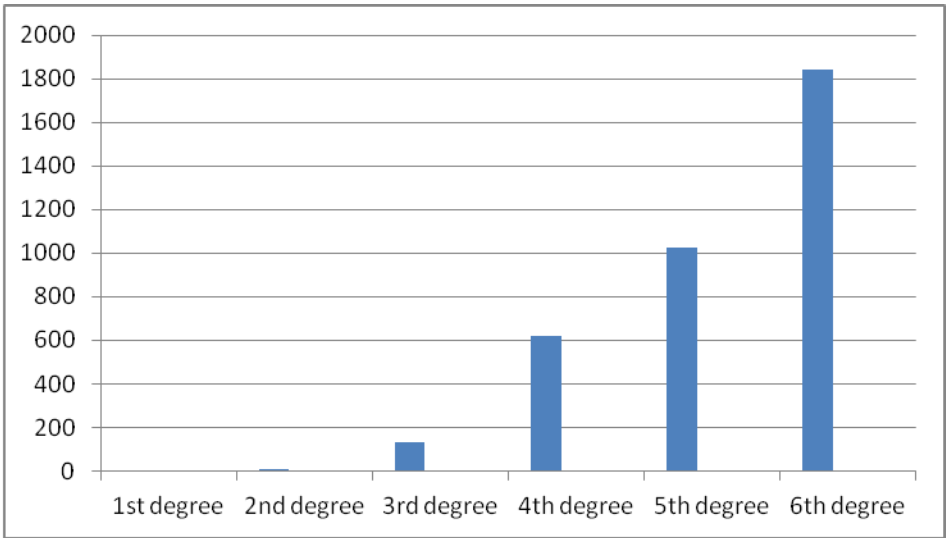


Figure 7. Average Nodes Rejected to Find a Link for Various Degrees of Separation

8. Conclusion

The idea that anyone in the world can be connected in six or less steps was enough to motivate us to look for its application in real life. Upon extensive research we came up with this idea to help people connect better with others using their common friends. Our application does a good job to achieve this but as the proper implementation of six degrees of separation requires a large search domain that is well connected, our application would fail to utilize the true meaning of "six degrees of separation" for every user without a well populated database.

Also the use of various optimizing techniques such as hash tables, look-back tables and heuristic priority enables our algorithm to get search results quite fast but this is only tested on a small database, which we feel may need a lot more optimization in future while dealing with a larger database.

9. Future Works

The present application that we have developed which is available on the internet makes use of a free database hosting facility. Since we have used a free service the number of transactions per second is very limited and often not enough to obtain a valid result before connection timeout. Therefore because of transaction limitations we would like to use dedicated database service for our application in the future. This would ensure that more number of users will be able to simultaneously use our application and query our database without having to face connection timeouts.

Also the java desktop application is not very appealing nowadays since we are living in an online age. Thus we would like to take our application further and make it available in the form of a website, which will make it even more convenient for people to access and use it.

In future we would also research into ways to make our algorithm obtain search results faster by implementing dynamic ranking techniques. These ranking techniques dynamically evaluate the best appropriate node that should be taken from the present node based on previous search results.

References

[1] Kevin Bacon, http://www.sixdegrees.org.

[2] E. Edwin Lawrence and R. Latha, "Analysis of Six degrees of separation in Facebook using Ant colony optimization", IEEE proceedings of International Conference on Circuit, Power and Computing Technologies (ICCPCT), 19-20 March 2015, Nagercoil, pp. 1-5, (**2015**).

[3] Ke Xiao-hua, "A Social Networking Services System based on the "Six Degrees of Separation" Theory and Damping Factors", IEEE proceedings of Second International Conference on Future Networks (ICFN-2010), 22-24 January 2010, Sanya, Hainan, **(2008)**, pp. 438-441.

[4] Reza Bakhshandeh, Mehdi Samadi, Zohreh Azimifar and Jonathan Schaeffer(2011), "Degrees of Separation in Social Networks", Proceedings of The Fourth International Symposium on Combinatorial Search (SOCS), 15-16 July 2011, Barcelona, Spain, pp. 18-23.

[5] Lei Zhang and Wanqing Tu (2009), "Six Degrees of Separation in Online Society", Proceedings of the WebSci'09, Society On-Line, 18-20 March 2009, Athens, Greece, pp. 1-5.

[6] Hakan Kardes, Abdullah Sevincer, Mehmet Hadi Gunes, and Murat Yuksel (2012), "Six Degrees of Separation among US Researchers", IEEE/ACM Proceedings of International Conference on Advances in Social Networks Analysis and Mining, 26-29 August 2012, Kadir Has University, Istanbul, Turkey, pp. 654-659.

[7] Facebook, https://www.facebook.com.

[8] SixDegrees.com, http://www.sixdegree.com.

[9] LinkedIn, https://www.LinkedIn.com.

Authors

Shubham Kumar, he is currently pursuing his bachelor's degree in computer science and engineering from VIT University, Vellore-632014, Tamil Nadu, India. His areas of interest are Data mining, Big data, Algorithm designing, Networking and Cloud computing.

Shivank Sharma, He is currently pursuing Bachelors of Technology in Computer Science and Engineering at the Vellore Institute of Technology, Vellore, Tamil Nadu, India. His major interest work area is Networking, Cloud Computing, Big data, Data warehousing and data mining.

N. Ch. S. N. Iyengar, he is a Senior Professor at School of Computing Sciences and Engineering at VIT University, Vellore, TN, India. His research interests include Security in Networks, Information, Cloud, ,Agent computing,e-services and Fluid Dynamics (Porous Media). He had 30yrs of teaching and 20 yrs of research experience with a good number of publications in reputed International Journals & Conferences. He chaired many Intl. Conf. and delivered Key note lectures, served as PC Member/Reviewer. He is Editor in chief and also Editorial Board member for many Int'l Journals.