I.J. Intelligent Systems and Applications, 2014, 12, 48-54
Published Online November 2014 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2014.12.07

Vague Logic Approach to Disk Scheduling

Priya Hooda
Dept. of Computer Science Engineering, ITM University, Gurgaon, India
Corresponding Author Email: priya.26hooda@gmail.com

Supriya Raheja
Dept. of Computer Science Engineering, ITM University, Gurgaon, India
Email: supriya@itmindia.edu

Abstract— Vague sets theory separates the evidences in favour and against of an element in a set which provides better mechanism to handle impreciseness and uncertainty. This research paper aims to handle the incompleteness and impreciseness of data associated with the disk access requests. Here, we propose a new disk scheduling algorithm, Vague Disk Scheduling (VDS) Algorithm, based on vague logic. The proposed framework includes Vague-Fuzzification Technique, Priority Expression, and VDS Algorithm. The Vague-Fuzzification Technique is applied to the input data of each disk access request and generates a priority for each request in the queue. Based on the priority allotted the requests are serviced. Finally work is evaluated on different datasets and finally compared with Fuzzy Disk Scheduling (FDS) Algorithm. The results prove that VDS algorithm performs better than FDS Algorithm.

Index Terms— Disk Scheduling, Hard Disk, Vague Logic, Vague-Fuzzification, Computer Science.

I. Introduction

All modern computers use hard disks to store large amount of data and information. For these computer systems, hard disk drives provide the bulk secondary storage [1]. In a multiprogramming environment with many processes, to provide control and uniform memory access to each process, the concept of disk scheduling is used. A disk drive consists of a number of platters. Each platter has two surfaces. Each surface of a platter is made up of concentric circles, called tracks. A single surface contains thousands of tracks. Typically a surface consists of 500-2000 tracks. Given a queue of disk access requests, to service a request, firstly the disk arm head is moved to the desired track, called seek distance. Then the disk rotates to the desired sector until it is under the read/write head. The time taken in the rotation caused is called rotational delay. The rate of rotation is termed as RPM, Rotations per Minute. The typical range of RPM lies in between 3600-7200 RPM. Various disk scheduling algorithms are available that uses either one of the above stated factors or both of them to improve the overall disk access time.

Because processor speed and memory capacity are increasing faster than the hard disk drive speed, we need methods for using disk drives more efficiently. Apart from traditional disk scheduling algorithms, with the advent in research and technology, new disk scheduling techniques and algorithms are proposed and developed using fuzzy logic [2] [3], providing a better understanding and a better way to deal with uncertainty associated with the disk access requests.

In this research paper we are extending the research in the field of disk scheduling using vague logic. The expression of human judgment often lacks precision and the confidence levels on the judgment contribute to various degrees of uncertainty [4]. Vague Logic is an extension of fuzzy logic that deals with uncertainty, impreciseness, and ambiguity associated with the information in a better way. In our work, we are implementing a new algorithm for scheduling disk access requests, Vague Disk Scheduling (VDS). This algorithm uses Vague-Fuzzification [5] technique based on vague sets theory. The algorithm computes a priority for each request in the queue and based on this priority the next request is serviced in a way that provides an optimal access time.

This paper is organized in 7 sections. Section 2 throws light on past related work performed so far. Section 3 describes traditional disk scheduling algorithms and disk scheduling algorithm using fuzzy logic. Vague logic and Vague-Fuzzification technique is discussed in section 4. Section 5 contains proposed framework and Vague Disk Scheduling Algorithm. Results are evaluated and compared with disk scheduling algorithm based on fuzzy logic in section 6. Finally, section 7 includes conclusion of the research done.

II. Related Work

A lot of research is being performed in both the fields, i.e. disk scheduling, and applications based on vague logic.

In 2009, Mohammad Sofian Abu Talip [3] gave a new way to disk scheduling using the concepts of fuzzy logic and fuzzy inference system. For disk scheduling optimization fuzzy logic can be applied in two categories: the servo controller that controls the arm head of the movable-disk and disk scheduling policy. The focus of this paper is to apply fuzzy logic to a disk scheduling policy. Using fuzzy inference system and IF-THEN rules on two inputs, seek time and arrival time, he gave a new approach for scheduling disk requests. The author has proposed a disk scheduling algorithm based on First

Come First Serve Policy using MATLAB Fuzzy Logic Toolbox. The validity of the proposed algorithm is proved by performing a comparison with the traditional disk scheduling algorithms.

In research paper [2] the author has designed a different approach by developing an FIS based on two inputs: seek distance and rotational delay for each disk access request. Based on the 9 IF-THEN rules the designed algorithm generates a Priority as output for every request in the queue. The highest priority request is chosen for servicing. After servicing the current request, the again the seek distance and rotational delay are calculated and the procedure is repeated until the queue becomes empty. This algorithm is proved to provide a better understanding and results.

The notion of vague logic is relatively new in the field of engineering and computer science, but in the past few years researchers have improved existing algorithms and application using vague logic. Wang Hong-xu [6], proposed a Vague Optimize algorithm to choose location of tailings dam. In the novel research work, by converting original data to vague set data, using similarity measures between vague sets and vague sort the best location of tailings dam is chosen. This algorithm proposed the following steps: establishing the index set, screening out the locations of tailings dam, creating the vague environment by transforming the original data of an expected location of tailings dam into vague data, calculation of similarity measures between vague set of locations of tailings dam and vague set of expectation location of tailings dam, and application of vague optimize sort. The concept of vague data mining is also used while incorporating similarity measures between vague sets.

Li Guxin [7] keeping in mind the increasing use and applicability of vague sets in engineering applications gave a notion to construct a vague environment in engineering applications. The motive behind construction of vague environment is that we can study, analyze, and solve practical problems based on vague sets only in a vague environment. Earlier there were methods which first convert raw data to fuzzy data and then fuzzy data to vague data. A notion is provided in this research paper to create vague environment by transforming raw data to vague data.

FuJin Zhang [8] has used vague set similarity measures on three-dimensional representation for pattern recognition in the research paper "A Class of Similarity Measures between Vague Sets and Its Applications to Pattern Recognition", 2010.

Ritu Aggarwal [9] has described a Vague Set based Controller Power System Stabilizer (VCPSS). This controller is evaluated on a single machine infinite bus power system. The results provided shows the superiority and robustness of a vague controller power system stabilizer as compare to conventionally tuned controller and fuzzy logic controller to enhance system dynamic performance over a wide range of operating conditions [8].

Yong Liu [10] studied and analyzed the relationship between fuzzy sets and vague sets in this paper. It is emphasized that vague sets are more accurate in describing vague information than fuzzy sets. A general model for transformation of vague sets into fuzzy sets is proposed and implemented. This transformation establishes a mathematical relationship between vague set theory and fuzzy set theory. This general model is said to be useful in applications which involve imprecise information processing and vague information process.

III. Disk Scheduling

When processes running on a machine have multiple requests to read data from or to write data to disk and an optimum order is needed to access them effectively, disk scheduling is used. A hard disk drive is composed of several concentric, rapidly-rotating platters, where data may be written to both sides of each platter [11]. Each platter is logically divided into several circular tracks. These tracks are further divided into equal sized sectors, where each sector can hold 512 bytes of data.

For a particular disk request, when a read/write head accesses a particular sector, the head suffers two kinds of delays. First is seek time or seek distance, which is the time required to move the head to desired track. The second is rotational delay, after setting on a particular track the head rotates on the track to the desired sector to read/write data and service a particular disk access request.

To select a request for servicing from the queue of disk requests, different algorithms such as First Come First Serve, Shortest Seek Time First, LOOK, C-LOOK, SCAN, C-SCAN. These are traditional disk scheduling algorithms that aim to minimize seek time and are optimized for aggregate throughput.

While FCFS uses a FIFO queue and selects requests sequentially to serve them. It is easy to implement and fair in the sense that once a disk access request has arrived, its place in the schedule is fixed [12]. SSTF gives priority for servicing to the request which has shortest seek time/distance. In SCAN algorithm the head starts from one end, travels to the other end servicing the disk requests and then redirects itself.

The LOOK algorithm works in a way similar to SCAN algorithm but the head does not reach the extreme end of disk, but changes its direction when it reaches the largest and smallest requested track. Recognition of the dynamic request of the queue leads to the LOOK algorithm [13]. The C-SCAN algorithm is a variation of SCAN algorithm, in which the head moves in one direction servicing the requests. After reaching the end of disk the head reverses and continues.

C-LOOK algorithm, a variation of LOOK algorithm, works in same way as C-SCAN does with the difference that here the head does not moves to the end of disk, it moves back from the last serviced request and continues. The C-LOOK algorithm looks for a request before continuing in a given direction [1].

A. Disk Scheduling Using Fuzzy Logic

In order to provide an efficient disk scheduling algorithm, we have implemented disk scheduling using fuzzy logic [2]. A Mamdani type Fuzzy Inference System (FIS) took two inputs: seek distance and rotational delay. The membership functions used for both the inputs and the output are of trimf type with different range parameters. The Defuzzification technique used is Centroid of area. This FIS is composed of 9 IF-THEN Rules. These rules map the two inputs to a single output. The output parameter is termed as priority. The next request to be serviced is chosen on the basis of this priority value.

The fuzzy based algorithm, Fuzzy Disk Scheduling FDS [2] is designed to service the pending disk access requests in the queue based on the output priority yielding better results than traditional disk scheduling algorithms. In order to understand uncertainty associated with the disk access requests with more preciseness, this research paper extends the research from fuzzy logic to the next level using Vague Logic. This FIS is implemented using MATLAB Fuzzy Tool Box as shown in fig. 1.

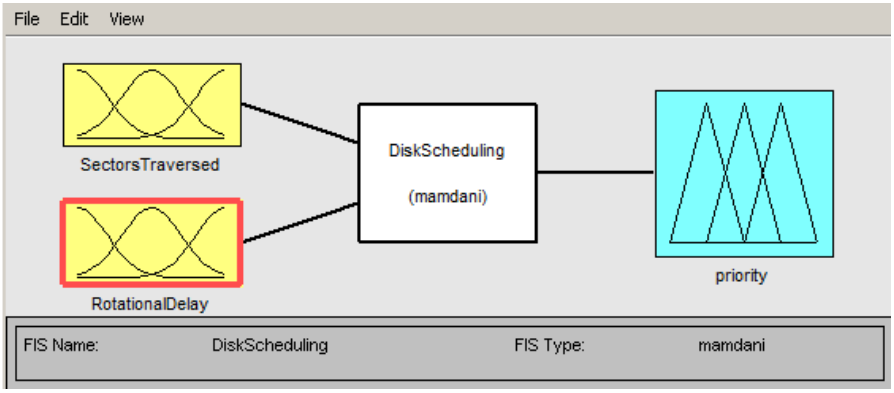

Fig. 1. FIS DiskScheduling

VI. Vague Logic

A vague set is a set of objects, each object having a grade of membership value in a continuous subinterval of [0, 1] [14]. A vague set is a generalization of a fuzzy set where instead of point based membership, an interval based membership is used [15]. In a space of objects, U, and u be the generic element of U, a vague Set V in U is characterized by two membership functions.

First is truth-membership function denoted by t_v, is a lower bound on the grade of u obtained from the evidence in favor of u, and second is false-membership function denoted by f_v, is lower bound on negation of u obtained from the evidence against u.

The grade of membership $\mu_v(u)$ in a vague set is bounded by $t_v(u) \le \mu_v(u) \le 1-f_v(u)$, where $t_v(u)+f_v(u) \le 1$. The uncertainty in a vague set is represented by the difference $1-f_v(u)-t_v(u)$. If this value is small, our knowledge is relatively precise and if the value is large our knowledge is little [5]. A vague set is graphically shown in fig. 2.

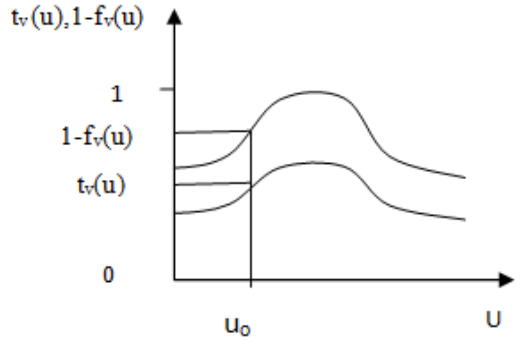

Fig. 2. Vague Set

A. Vague-Fuzzification

To incorporate the benefits of vague set theory in handling vagueness associated with information, a new technique Vague-Fuzzification is designed to implement fuzzification using vague sets [5]. The traditional fuzzification approach of fuzzy set theory uses different types of membership functions and shapes that use a point based membership assignment, whereas Vague-Fuzzification Technique uses vague sets and assigns grade of membership to each element of the set using interval based membership. This technique works in two steps:

1) First it uses the Positive Ordered Transforming Formula (POTF). This formula takes an index of single valued data and converts each value to corresponding vague valued data.

$$\left[V_u\left(c_n = [t_{mn}, 1-f_{mn}]\right) = \left[\frac{c^i_{mn} - c^i_{n.\min}}{c^i_{n.\max} - c^i_{n.\min}}, 1 - \frac{c_{n.\max} - c_{mn}}{c_{n.\max} - c_{n.\min}} \right] \right] \quad (1)$$

Where c_{mn} is single-value to be transformed, $c_{n.\min}$ is the minimum value from the index set and $c_{n.\max}$ is maximum value of the index set, and $i = 2, 3, 4\ldots$

2) In the second step it uses the General Transforming Model [8] to convert the vague valued data to the fuzzified values.

$$\mu_v = t_v(u) + \left(\frac{1}{2} \left[1 + \frac{t_v(u) - f_v(u)}{t_v(u) + f_v(u) + 2p} \right] \left[1 - t_v(u) - f_v(u) \right] \right) \quad (2)$$

Where $t_v(u)$ is truth membership value of vague set *u*, $f_v(u)$ is false membership value of the vague set *u*, and *p* is a positive real number.

The MATLAB implementation of the Vague-Fuzzification Technique is shown in fig. 3.

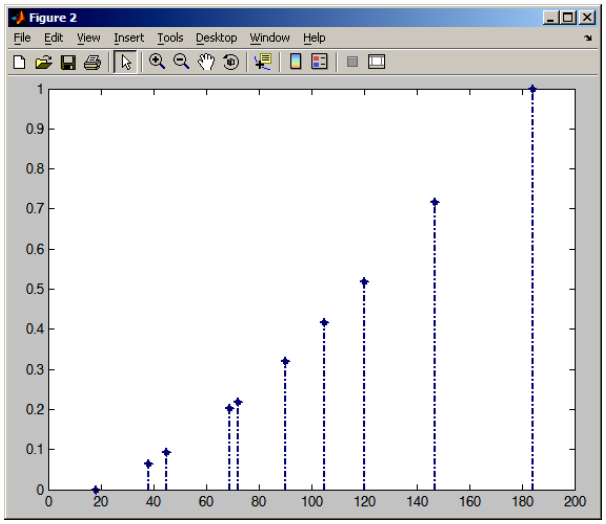

Fig. 3. Vague-Fuzzification Implementation in MATLAB.

V. Vague Disk Scheduling (VDS) Algorithm

In this section we are discussing the proposed architecture, as shown in fig. 4, and the proposed algorithm.

The proposed architecture follows three main steps:

a. Vague-Fuzzification,
b. Priority Expression,
c. Vague Disk Scheduling (VDS) Algorithm.

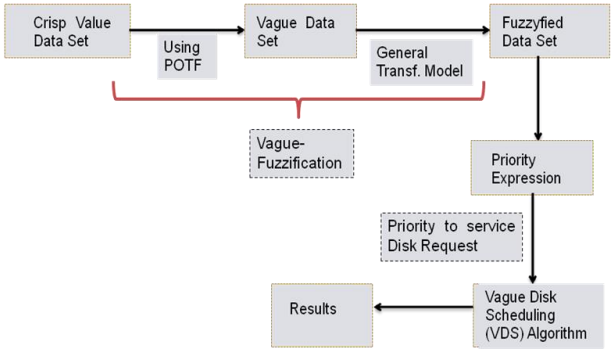

Fig. 4. The Proposed Architecture.

In the proposed architecture the Vague-Fuzzification step transforms the input crisp values, seek distance and rotational delay, to the vague-fuzzified values. The Priority Expression takes as input the vague-fuzzified values and computes the Priority for each disk access request. And the VDS Algorithm unit includes the step by step procedure to schedule the queue of disk access requests on the basis of the calculated priority such that the pattern of serviced disk requests provides optimal access time.

A. Proposed Algorithm

In the proposed VDS algorithm, we are using vague logic to select the disk request to be serviced by the scheduler. The algorithm computes a list of Priorities for each request in the queue and selects the one with highest priority. Taking Seek Distance and Rotational Delay as inputs, the Vague-Fuzzification technique computes the corresponding vague-fuzzified values. The Priority Expression takes the vague-fuzzified values as input and calculated optimized priority for each request. Based on the criteria, next request to be serviced is chosen. Finally, on the basis of the pattern of serviced disk access requests, total seek distance and rotational delay is calculated.

Assumptions

Disk Storage Capacity:	80 GB
No. of Heads:	16
No. of Cylinders:	158816
No. of Sectors:	63
Rotations per Minutes (RPM):	7200
Maximum Seek Time:	19 ms
Maximum Rotational Delay:	8.33 ms

Algorithm VDS

Inputs: VDS [Estimated Seek Distance, Rotational Delay].

Output: [Total Seek Distance, Rotational Delay].

1. Calculate the estimated Seek Distance (SD) from the current head position over the track/cylinder for each request forming the queue using the given relation:

   SD=|Current head position - next request position|

2. Apply Vague-Fuzzification Technique on Seek Distance to calculate fuzzified values.
3. Calculate the Rotational Delay (RD) from the current head position over the sector for each request forming the queue using the given relation:

   RD=8.33*No. of Sectors causing Rotation/63

4. Apply the Vague-Fuzzification Technique on Rotational Delay and calculate fuzzified values.
5. Calculate the priority, for each request using the fuzzified values of the Seek Distance and Rotational Delay, to schedule next request using the given relation:

   P = [(2SD)*RD]/2

   Assume [0] as the highest priority.
6. Sort the queue in the descending order on the basis of priority calculated.
7. Service the request with highest priority in the queue.
8. After servicing each request, update the head position and repeat steps 1 to 8 until the queue is empty.
9. Based on the pattern of serviced requests calculate the total seek distance and rotational delay as a measure of total access time accordingly.

VI. Performance Analysis

For getting clear results of the proposed algorithm, we have implemented the VDS Algorithm using MATAB. The performance of VDS Algorithm is compared against Fuzzy Disk Scheduling (FDS) Algorithm [2] in terms of Total Seek Distance and Rotational Delay 4 different cases based on different data sets are studied and

analyzed. The comparison made is tabularized in tables and illustrated graphically using graphs. The statistics calculated shows that VDS algorithm performs better than the FDS Algorithm. However, in some cases the performance of both the algorithms is equivalent.

A. Case Study 1

Assume the following queue of disk access requests: 55, 58, 39, 184, 90, 160, 150, 38, 18, with head currently positioning at track 100. Implementing the VDS Algorithm the disk requests are scheduled based on their computed Priorities. The results are summarized and compared with results of FDS Algorithm in table 1.

Table 1. Results and comparison using VDS and FDS algorithm.

Algorithm	Seek Distance (Tracks)	Rotational Delay (ms)
VDS	248	28.477
FDS	280	35.82

B. Case Study 2

Consider a queue of disk access requests: 98, 183, 37, 122, 14, 124, 65, 67, with head currently positioning at track 53. Based on VDS Algorithm and FDS algorithm the results are summarized in Table 2.

Table 2. Results and comparison using VDS and FDS algorithm.

Algorithm	Seek Distance (Tracks)	Rotational Delay (ms)
VDS	299	28.29
FDS	350	25.18

C. Case Study 3

Assume a queue of disk access requests to be serviced: 23, 89, 132, 42, 187, and head presently servicing request on track 100. The results using VDS Algorithm are tabularized in Table 3, with comparison against results computed using FDS Algorithm.

Table 3. Results and comparison using VDS and FDS algorithm.

Algorithm	Seek Distance (Tracks)	Rotational Delay (ms)
VDS	241	19.83
FDS	305	19.83

D. Case Study 4

Taking a queue of disk access requests as 8 147 177 94 150 102, 130 and head positioning at track 14. Evaluated results of VDS Algorithm are summarized in tabular form in table 4, comparing with FDS Algorithm.

Table 4. Results and comparison using VDS and FDS algorithm.

Algorithm	Seek Distance (Tracks)	Rotational Delay (ms)
VDS	335	15.8
FDS	352	15.8

The overall performance comparison of case 1, case 2, case 3, and case 4 are illustrated graphically in using charts as shown in fig. 5, fig. 6, fig. 7, and fig. 8 respectively. In the charts depicted below, the term SD, measured in track seek, stands for Seek Distance and RD, measured in milliseconds, stands for Rotational delay.

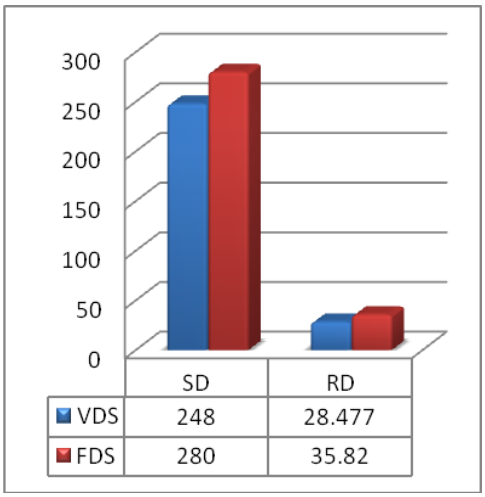

Fig. 5. Comparison based on case study 1.

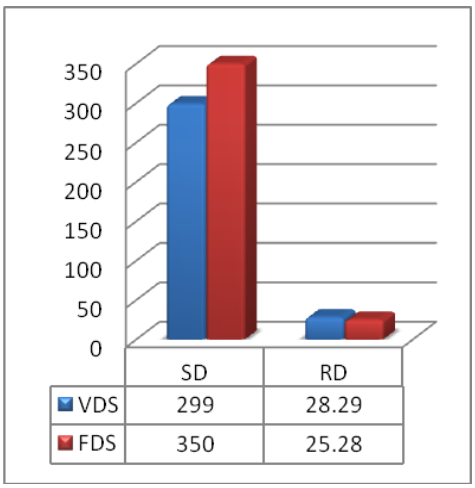

Fig. 6. Comparison based on case study 2.

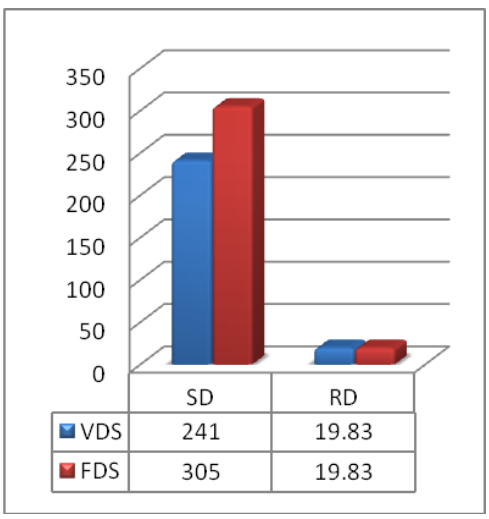

Fig. 7. Comparison based on case study 3.

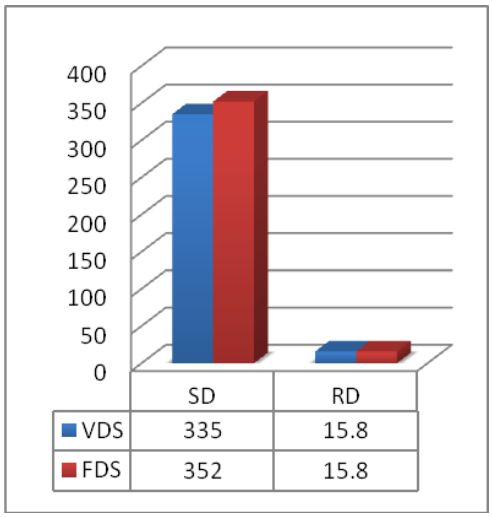

Fig. 8. Comparison based on case study 4.

VII. Conclusion

The fuzzy logic based algorithm for disk scheduling lacks the capability of dealing with the situation when a disk access request stands in an abstained condition. In this research paper the concepts of vague logic are introduced and applied for scheduling a queue of disk access requests. An algorithm, VDS Algorithm, is proposed that intelligently removes the abstained condition. This algorithm uses the technique Vague-Fuzzification to fuzzify the inputs. The proposed algorithm takes on two inputs, Seek Distance and Rotational Delay, and computes a priority, using priority expression, for each request for disk access. This priority is then used for scheduling all the disk access requests, thus yielding an improved overall access time. To show the validity of the research performed different cases are studied and analyzed. The results shows clear improvement in performance as compared against Fuzzy Logic based Disk Scheduling Algorithm.

Acknowledgment

I would like to thank my mentor Ms. Supriya Raheja and my institution for guiding and helping me in the research performed.

References

[1] Silberschatz, Galvin and Gagne "Operating Systems Concepts", 8th Edition, Wiley, 2009.

[2] Priya Hooda, Supriya Raheja, "A New Approach to Disk Scheduling Using Fuzzy Logic", *Journal of Computer and Communication*, Vol. 2, No. 1, Jan 2014, pp. 1-5. http://dx.doi.org/10.4236/jcc.2014.21001.

[3] M. S. A. Talip, A. H. Abdalla, A. Asif and A. A. Aburas, "Fuzzy Logic Based Algorithm for Disk Scheduling Policy," *International Conference of Soft Computing and Pattern Recognition*, 2009, pp. 746-749. doi.ieeecomputersociety.org/10.1109/SoCPaR.2009.151.

[4] Dug Hun Hong and Chang-Hwan Choi, "Multi-criteria Fuzzy Decision Making Problems based on Vague Set Theory" Fuzzy Sets and Systems, 114, 2000, pp. 103-113.

[5] Priya Hooda, and Supriya Raheja, "Implementation of Vague-Fuzzification using Vague Sets", *International Journal of Computer and Applications*, Vol. 87, No. 11, Feb 2014, pp. 14-17.

[6] Wang Hong-xu, "VO Algorithm and It's an Application for the Locations of Tailings Dam", *Information Technology Journal*, Vol. 11, No. 4, 2012, pp. 554-556. doi.10.3923/itj.2012.554.556.

[7] Li Guxin, Wang Hong-xu, and Zhang Chengyi, "Constructing Vague Environment", *Fuzzy Information and Engineering*, AICS 78, pp. 711-715.

[8] FuJin Zhang, and Hongxu Wang, "A Class of Similarity Measures between Vague Sets and Its Applications to Pattern Recognition", *International Conference on Educational and Network Technology*, 2010, pp. 366-367.

[9] Ritu Aggarwal, Shailja Shukla, and S.S. Thakur, "Comparative Analysis of Vague Controller and Fuzzy Controller for Single Machine Infinite Bus System", *International Conference on Fuzzy Systems*, July 2013, pp. 1-5.

[10] Yong Liu, Guoyin Wang, and Lin Feng, "A General Model for Transforming Vague Sets into Fuzzy Sets", *Transactions on Computer Science II*, LNCS 5150, 2008, pp. 133-144.

[11] Matthew Andrews, Michael A. Bender, Lisa Zhang, "New Algorithms for the Disk Scheduling Problem", *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, 1996, pp. 550-559.

[12] Manish Kumar Mishra, "An Improved FCFS (IFCFS) Disk Scheduling Algorithm" *International Journal of Computer Applications*, Vol. 47, No. 13, June 2012, pp 20-24.

[13] A. Thomasian and C. Liu, "Disk Scheduling Policies with Lookahead," *ACM Sigmetrics Performance Evaluation Review*, Vol. 30, No. 2, 2002, pp. 33. http://dx.doi.org/10.1145/588160.588165

[14] Wen-Lung Gau, Daniel J. Buehrer, "Vague Sets", *IEEE Transactions on Systems, Man, and Cybernatics*, Vol. 23, No. 2, March/April 1993, pp. 610-614.

[15] An Lu, Wilfred Ng, "Vague Sets or Intuitionistic Fuzzy Sets for Handling Vague Data: Which One Is Better?" *Proceedings of the 24th International Conference on Conceptual Modeling*, Oct. 2005, pp. 401-416. 10.1007/11568322_26.

Authors' Profiles

Priya Hooda was born on December 26, 1990. She received the B.Tech degree from Maharishi Dayanand University in 2012. Presently she is pursuing M.Tech. degree from ITM University.

Her thesis topic is Disk Scheduling Using Vague Logic. Her representative published research papers lists as follow: A New Approach to Disk Scheduling Using Fuzzy Logic (Journal of Computer and Communication 2014), Implementation of Vague-Fuzzification using Vague Sets (International Journal of Computer and applications, 2014). Her research interests include Fuzzy Logic, Vague Logic, object oriented programming, operating system, and disk scheduling.

Supriya Raheja was born on March 13. She is currently Assistant Professor (Senior Grade) in Department of CSE & IT,

School of Engineering & Technology, ITM University. She has about 7 years of Teaching Experience.

She is currently pursuing Ph.d. in Computer Science. Her areas of interest include Object Oriented Programming, Rapid Application Development, Operating System, Networks, Fuzzy Logic and Vague Logic. She has written 5 Laboratory Manuals. She had guided 6 M.Tech Projects and more than 25 B.Tech Projects. She has Published 20 Papers in International Journals with good indexing and in IEEE International Conferences. She is the reviewer and assistant editor of various international journals. She is the Faculty Coordinator for the ISTE Student Chapter. Along with this she is the Chief Coordinator of department magazine REFLECTION.

How to cite this paper: Priya Hooda, Supriya Raheja,"Vague Logic Approach to Disk Scheduling", International Journal of Intelligent Systems and Applications(IJISA), vol.6, no.12, pp.48-54, 2014. DOI: 10.5815/ijisa.2014.12.07

