Summary* of
Minimal-Total-Processing-Time Drum and Disk Scheduling Disciplines**

Samuel H. Fuller
Depts. of Computer Science and Electrical Engineering
Carnegie-Mellon University
Pittsburgh, Pennsylvania

Abstract

This article investigates the application of minimal-total-processing-time (MTPT) scheduling disciplines to rotating storage units when random arrival of requests is allowed. Fixed-head drum and moving-head disk storage units are considered and particular emphasis is placed on the relative merits of the MTPT scheduling discipline with respect to the shortest-latency-time-first (SLTF) scheduling discipline. The results of the simulation studies presented show that neither scheduling discipline is unconditionally superior to the other. For most fixed-head drum applications the SLTF discipline is preferable to MTPT, but for intra-cylinder disk scheduling the MTPT discipline offers a distinct advantage over the SLTF discipline. An implementation of the MTPT scheduling discipline is discussed and the computational requirements of the algorithm are shown to be comparable to SLTF algorithms. In both cases, the sorting procedure is the most time consuming phase of the algorithm.

1. Introduction

First-in-first-out (FIFO) and shortest-latency-time-first (SLTF) scheduling disciplines for drum and disk storage units have received considerable attention [cf. 1, 2, 3, 4, 8]. In [5] however, a new scheduling discipline, the minimal-total-processing-time (MTPT) discipline, is introduced for devices with rotational delays, or latency. The MTPT discipline finds a schedule for an arbitrary set of I/O requests that minimizes the total processing time for the set of requests. Moreover, if we let N be the number of I/O requests to be serviced, the original article presents an MTPT scheduling algorithm that has a computational complexity of O(NlogN), the same complexity as an SLTF scheduling algorithm. Our purpose here is to investigate the practical implications of the MTPT drum scheduling discipline.

Although the MTPT scheduling algorithm is known to enjoy a computational complexity of O(NlogN), nothing has been said about the absolute amount of time required to compute MTPT schedules. MTPT scheduling disciplines will be of little practical interest if it takes NlogN seconds to compute MTPT schedules, when current rotating storage devices have periods of revolution on the order of 10 to 100 milliseconds. In the next section we present the computation time required for a specific implementation of the MTPT scheduling algorithm.

In this summary the MTPT scheduling discipline is applied to the two most common classes of rotating storage devices: the fixed-head drum and the moving-head disk. The essential characteristics of a fixed-head drum is that there is a read-write head for every track of the drum's surface. Furthermore, it allows information to be stored in records of variable length and arbitrary starting addresses on the surface of the drum.

The only difference between a moving-head disk and a fixed-head drum is that a particular read-write head of a moving-head disk is shared among several tracks, and the time associated with repositioning the read-write head over a new track cannot be ignored. A set of tracks accessible at a given position of the read-write arm is called a cylinder.

2. Implementation of the MTPT Drum Scheduling Discipline

In this section we add some quantitative substance to the significant, but qualitative, remark that a MTPT drum scheduling algorithm has a computational complexity of O(NlogN).

MTPT0, a Fortran implementation of the original MTPT scheduling algorithm, is listed in [6]. (Fortran was used because a good set of compilers existed locally to support it.) Expressions (2.1) and (2.2) show the computation time required by MTPT0 as a function of N, the queue size. These times were collected on an IBM 360/91 and the subroutines were compiled by the IBM Fortran H compiler with maximum code optimization requested. The following expression is the expected execution time of the entire MTPT0 procedure.

$$100N + 50 \text{ microseconds} \qquad (2.1)$$

for $N < 8$. For $N \geq 8$ the sorting algorithm begins to exhibit its greater than linear growth rate. The next expression

---

*This is a summary of a paper that has been submitted for publication to the Comm. of the ACM.

**This work was conducted while the author was a Hertz Foundation Graduate Fellow at Stanford University, Stanford, Calif., and was partially supported by the Joint Services Electronics Program, U.S. Army, Navy, and Air Force under Contract N-00014-67-A-0112-0044. Computer Time was provided by the Atomic Energy Commission under Contract AT-(04-3)515.

approximates the expected execution time for MTPT0 minus the time it spends sorting.

$$50N + 50 \text{ microseconds.} \qquad (2.2)$$

These expressions suggest a practical implementation of MTPT0 might maintain a sorted list of the initial and final addresses of the I/O requests at all times. Then a schedule could be found more quickly after it is requested since there would be no need to execute the costly sorting step.

These expressions can only be used as a rough guide to execution times since it is very sensitive to implementation and MTPT0 was written to maximize clarity, not efficiency. Anyone considering implementing the MTPT algorithm is encouraged to read [6] where a more detailed performance analysis of MTPT0 is presented. The MTPT0 algorithm is not the only MTPT scheduling algorithm that may be of practical significance; for example, consider Fig. 2.1. Application of the MTPT0 scheduling algorithm shows that the schedule it constructs is

4, 3, 5, 1, 2.

This is an MTPT schedule, but then so are the sequences

5, 3, 4, 1, 2;
4, 1, 3, 5, 2; and
5, 1, 3, 4, 2.

Two of the MTPT sequences for the example of Fig. 2.1 share a distinct advantage over the MTPT sequence constructed by MTPT0. The last two sequences process record 1 on the first revolution while the sequence constructed by MTPT0, as well as the second sequence, overlook record 1 on the first revolution, even though they are latent at the time, and process it on the second revolution. Any reasonable measure of drum performance

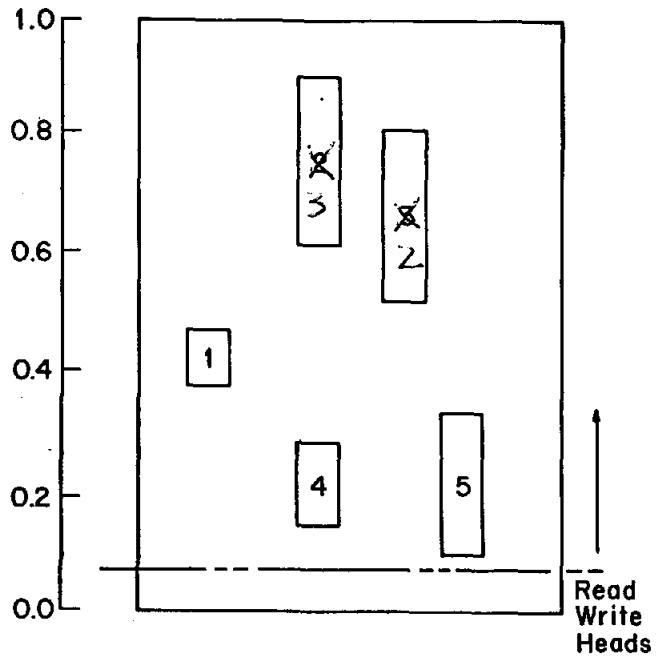

Figure 2.1. An example with four MTPT sequences.

will favor the last two MTPT sequences over the first two. For this reason two other MTPT algorithms have been implemented, MTPT1 and MTPT2. They are an improvement over MTPT0 and the cost of using them is increased computation time. For details on these other MTPT algorithms see[6].

3. Random Arrivals and Fixed-Head Drums

Several assumptions are required to specify the models of drum and record behavior that are used in the simulations. First, recent measurement of an operational computer system has shown that the starting addresses of successive I/O requests can be realistically modeled as independent random variables uniformly distributed around the circumference of the drum, and the length of the records can be approximated by exponentially distributed random variables with a mean of one third of the drum's circumference [4]. The remaining assumptions are: the arrival of I/O requests form a Poisson process ; the time required to compute the scheduling sequence is assumed to be insignificant; the endpoints are allowed to be real numbers in the interval [0,1); the period of revolution of the drum will be assumed time-invariant; no distinction is made between reading and writing on the drum; and no attempt is made to model the time involved in electronically switching the read-write heads.

Figure 3.1 shows the mean I/O waiting times for a fixed-head drum servicing record with lengths drawn from an exponential distribution with a mean of 1/2. The mean waiting time is shown as a function of ρ, where ρ is the ratio of the expected record transfer time to the expected interarrival time. The figure shows that the SLTF and MTPT2 curves are identical to within the accuracy of the simulation and MTPT1 and MTPT0 perform progressively poorer than MTPT2 and SLTF as the arrival rate of I/O requests is increased. The observation that MTPT0 and MTPT1 are poorer scheduling disciplines than the SLTF disciplines for heavily loaded drums is not too surprising. It is rare for large ρ that all the current requests will be processed before any new request arrives. When an additional request does arrive, a new MTPT sequence must be calculated and the non-robust nature of the MTPT0 algorithm suggests there will be little resemblance between the original and new sequences.

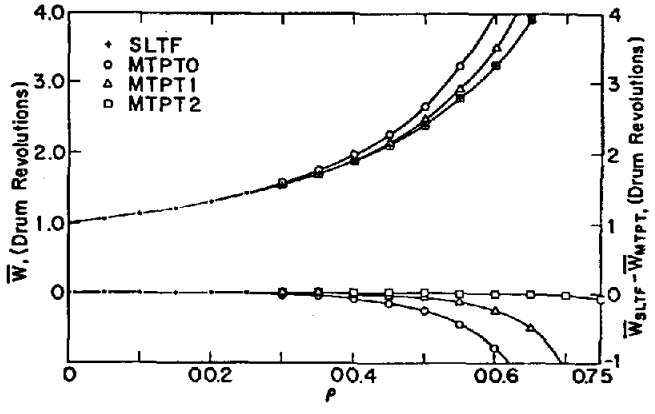

Figure 3.1. The expected waiting time when the records are exponentially distributed records with a mean of 1/2.

Figure 3.2 shows the mean waiting time for a drum with exponentially distributed records with a mean of 1/6. This figure is indicative of a general trend; the performance of the MTPT disciplines become worse with respect to the SLTF discipline as the mean record size decreases. The reader is referred to [6] for more complete details.

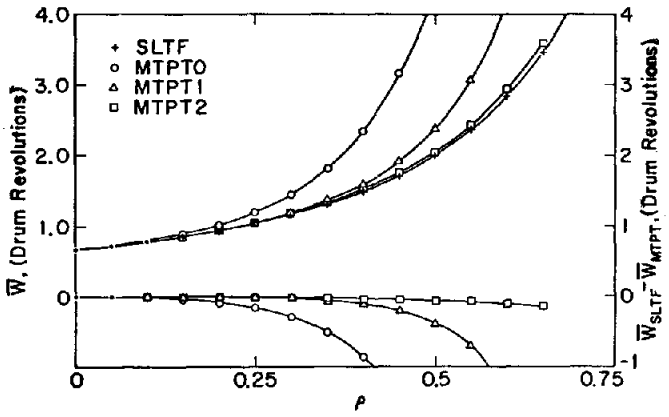

Figure 3.2. Standard deviation of the waiting time for exponentially distributed records with a mean of 1/2.

Figure 3.3 shows the performance of a drum with records exactly 1/2 of a drum revolution in length. In conjunction with Fig. 3.1, this figure illustrates that the relative performance of the MTPT schedules improves with respect to the SLTF schedule as the variance in record length decreases. Again, for further details of this aspect of the MTPT-SLTF comparison see [6].

Figure 3.3 includes another curve in addition to four curves already discussed. This curve shows the expected waiting time with the drum organized as a paging drum, with 2 pages per track [2,3]. The paging drum shows a pronounced improvement over any of the four scheduling disciplines discussed in this article, and if a drum is only going to service fixed-size records, Fig. 3.3 indicates the pronounced advantage in organizing the drum as a paging drum.

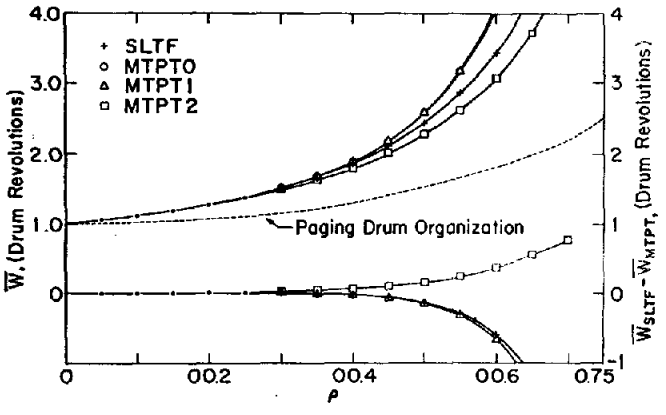

Figure 3.3 The expected waiting time when all the records are 1/2 the drum's circumference in length.

4. Random Arrivals and Moving-Head Disks

A few more remarks must be made on the simulations in order to completely specify conditions leading to the results of this section. Let C be the distance, in cylinders, that the head must travel, then the following expression roughly models the characteristics of the IBM 3330 disk storage unit [7]:

$$\text{seek time} = 0.6 + .0065\ C\ . \qquad (5.1)$$

The inter-cylinder scheduling discipline chosen for this study is SCAN[4]. The cylinder address for I/O requests are independent and uniformly distributed across the cylinders. (Conventional disk storage units have from 200 to 400 cylinders per disk but for any given set of active jobs, only a fraction of the cylinders will have active files. Therefore, the results of this section for disks with 5 and 10 cylinders is likely to be a good indication of the performance of a much larger disk that has active files on 5 to 10 cylinders.) Finally, in all the situations studied here, the records are assumed to be exponentially distributed with a mean of 1/2.

Figures 4.1 and 4.2 plot the mean I/O waiting time for the SLTF and MTPT scheduling disciplines applied to a disk and show quite a different result than for fixed-head drums. As the number of cylinders increases, the MTPT disciplines show more and more of an advantage over the SLTF scheduling discipline and just as importantly the difference between the MTPT disciplines decreases as the number of cylinders increases.

The reasons for the results seen in Figs. 4.1 and 4.2 are straightforward. The waiting time on an I/O request is made up of three types of intervals: the time to process the I/O requests on other cylinders, the time to move between cylinders, and the time to service the I/O request once the read-write heads are positioned over the I/O request's own cylinder. By their definition, MTPT disciplines minimize the time to process the set of I/O requests on a cylinder and hence minimze one of the three types of intervals in an I/O request's waiting time. All three MTPT disciplines process the I/O requests on a cylinder in the same amount of time, barring the arrival of a new request, and so the difference in expected waiting times between the three implementations can be expected to be reduced as the number of cylinders increases.

5. Conclusions

For fixed-head drums, in situations where: the variance in record sizes is small, the variance in waiting times must be minimized, or the mean duration of busy intervals must be minimized MTPT disciplines offer modest gains. It is crucial, however, to implement as good a MTPT discipline as possible, and unfortunately only the MTPT0 algorithm has been shown to enjoy an efficient implementation.

The MTPT scheduling discipline has been shown to be a promising intra-cylinder disk scheduling policy. For heavy loads significant improvements over the SLTF discipline are consistently achieved and just as importantly this improvement is relatively independent of the MTPT discipline used. In other words, MTPT0, which has an efficient implementation, offers very nearly as much of an improvement over SLTF as does MTPT1 and MTPT2 when five or more cylinders are actively in use.

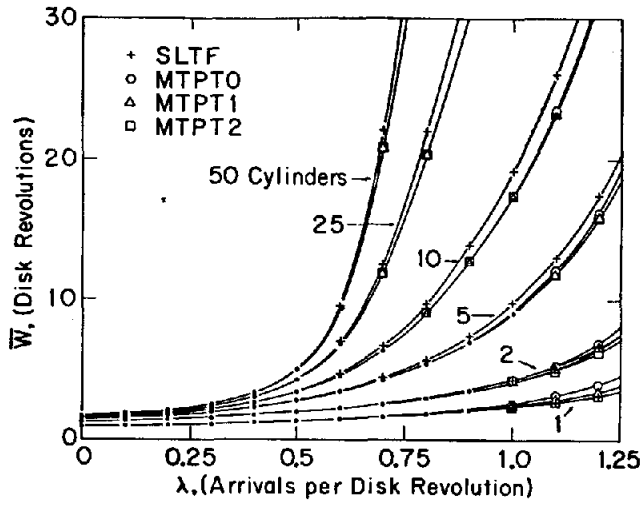

Figure 4.1. The expected waiting time of moving-head disks.

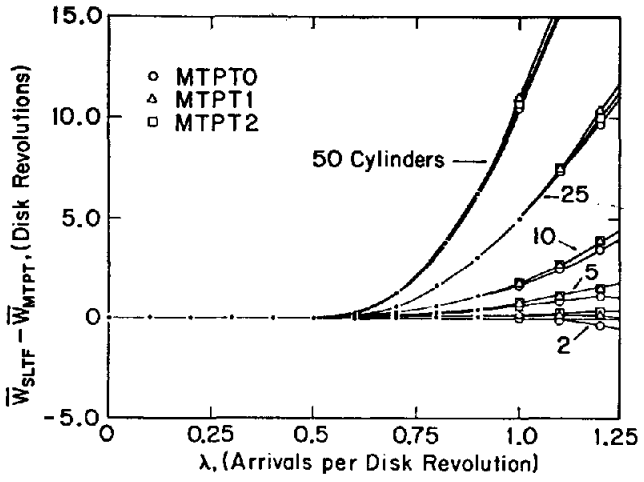

Figure 4.2. The difference between the expected waiting time of a moving-head disk when using the SLTF discipline and a MTPT discipline.

References

1. Abate, J., Dubner, H., and Weinberg, S. B., Queueing analysis of the IBM 2314 disk storage facility. J. ACM 15, 4 (Oct., 1968), 557-589.

2. Coffman, E.G., Analysis of a drum input/output queue under scheduling operation in a paged computer system. J. ACM 16, 1 (Jan., 1969), 73-90.

3. Denning, P. J., Effects of scheduling on file memory operations. Proc. AFIPS SJCC 30 (1967), 9-21.

4. Fuller, S. H. and Baskett, F., An analysis of drum storage units. Tech. Report 26, Digital Systems Laboratory, Stanford Univ., Stanford, Calif. (Aug. 1972). To appear in J. ACM.

5. Fuller, S. H., An optimal drum scheduling algorithm. IEEE Trans. on Computers C-21, 11 (Nov., 1972), 1153-1165.

6. Fuller, S. H., Random arrivals and MTPT disk scheduling disciplines. Tech. Report 29, Digital Systems Lab., Stanford Univ., Stanford, Calif. (Aug., 1972).

7. IBM 3830 storage control and 3330 disk storage. Order No. GA26-1592-1.

8. Teorey, T. J. and Pinkerton, T. B., A comparative analysis of disk scheduling policies. Comm. ACM 15, 3 (March, 1972), 177-184.