Alcom-FT Technical Report Series
ALCOMFT-TR-02-31

Random Arc Allocation and Applications*

Peter Sanders and Berthold Vöcking

Max Planck Institut für Informatik
Saarbrücken, Germany
[sanders,voecking]@mpi-sb.mpg.de

Abstract. The paper considers a generalization of the well known random placement of balls into bins. Given n circular arcs of lengths $\alpha_1, \ldots, \alpha_n$ we study the maximum number of overlapping arcs on a circle if the starting points of the arcs are chosen randomly. We give almost exact tail bounds on the maximum overlap of the arcs. These tail bounds yield a characterization of the expected maximum overlap that is tight up to constant factors in the lower order terms. We illustrate the strength of our results by presenting new performance guarantees for several application: Minimizing rotational delays of disks, scheduling accesses to parallel disks and allocating memory to limit cache interference misses.

1 Introduction

Randomly assigning tasks or data to computational resources has proved an important load balancing technique for many algorithmic problems. The model we study here was motivated by three different applications one of the authors became involved with where the lack of an accurate analysis of such a simple system became an obstacle. Section 3 gives more details on these applications: minimizing rotational delays in disk scheduling, scheduling parallel disks, and memory allocation to limit cache conflicts. The common theme there is that jobs (disk blocks, strings of disk blocks, memory segments) have to be allocated to a contiguous resource that wraps around (disk tracks, striped disks, memory locations mod cache size). In all three applications randomization is used to make worst case situations unlikely.

The following model describes all three applications. It is so simple that we expect further applications. An *arc allocation* describes the arrangement of n circular arcs that are pieces of a *unit circle*, i.e., a circle with circumference 1. We represent points on this circle by numbers from the half open interval $[0, 1)$. Let $0 \leq a_i < 1$ denote the starting point of *left endpoint* of arc i and α_i the *arc length* of arc i. Arc i spans the half open interval $[a_i, a_i + \alpha_i \bmod 1)$ where $x \bmod 1$ denotes the fractional part of x and where $[a, b)$ for $a > b$ denotes the set $[a, 1] \cup [0, b)$ in this paper. Let $\alpha = \sum_i \alpha_i / n$ denote the *average arc length*. If all arc length are identical, $\alpha_i = \alpha$ for all $0 \leq i < n$, we have a *uniform* arc allocation. In a *random* arc allocation, the starting points are chosen independently and uniformly at random. Let $L(x)$ denote the number of arcs

* Partially supported by the Future and Emerging Technologies programme of the EU under contract number IST-1999-14186 (ALCOM-FT).

containing x. Let $L = \sup_{x\in[0,1]} L(x)$ denote the *maximum overlap* of an arc allocation. In this paper, we estimate the expectation $\mathbf{E}[L]$ of the maximum overlap and derive tail bounds for L.

Let us go back to the interpretation of arcs as jobs to be executed/allocated. The maximum overlap L is important because all jobs can be executed using between L and $L+1$ trips around the circle if all jobs are known in advance (see Section 3.1). Furthermore, there is a natural online allocation algorithm that needs at most $2L$ trips around the circle (see Section 3.3).

1.1 New Results for Random Arc Allocation

In this paper we present almost exact tail bounds on the maximum overlap for random arc allocation. These tail bounds yield a complete characterization of the expected maximum overlap. Let $\Delta = L - \alpha n$ denote the difference between maximum and average load. We are able to describe $\mathbf{E}[\Delta]$ almost exactly in terms of Lambert's W function [5]. This function is discussed in more detail below. The tail bounds imply the following estimates for $\mathbf{E}[\Delta]$.

A) If $\alpha \leq \frac{\ln n}{n}$ then $\mathbf{E}[\Delta] = \mathcal{O}\left(\alpha n \exp\left(\mathrm{W}\left(\frac{\ln(1/\alpha)}{\alpha n}\right)\right)\right)$.

B) For $\alpha \in [\frac{\ln n}{n}, \frac{1}{2}]$, $\mathbf{E}[\Delta] = \mathcal{O}\left(\sqrt{\alpha n \ln\left(\frac{1}{\alpha}\right)}\right)$.

C) For $\alpha \in [\frac{1}{2}, 1 - \frac{\ln n}{n}]$, $\mathbf{E}[\Delta] = \mathcal{O}\left(\sqrt{(1-\alpha) n \ln\left(\frac{1}{1-\alpha}\right)}\right)$.

D) If $\alpha \geq 1 - \frac{\ln n}{n}$ then $\mathbf{E}[\Delta] = \mathcal{O}\left((1-\alpha) n \exp\left(\mathrm{W}\left(\frac{\ln(1/(1-\alpha))}{(1-\alpha) n}\right)\right)\right)$.

Our estimates on $\mathbf{E}[\Delta]$ in all four cases are essentially tight in the sense that they describe the case of uniform arc length exactly up to constant factors. Observe that the cases D) and C) are symmetric to the cases A) and B) in α and $1-\alpha$, resp. In fact, these bounds are derive using a simple symmetry arguments treating holes (i.e., the uncovered pieces of the circle) like arcs. In case A) it holds $\mathbf{E}[L] = \mathbf{E}[\Delta]$. Exact estimates for this case can be derived relatively easily using Chernoff bounds. More interesting is case D. To obtain tight estimates in this case, we need to combine Chernoff bounds with a random walk analysis.

Now let us come to a discussion of Lambert's W function. This function is defined to be the unique positive solution to the equation $\mathrm{W}(x) \cdot \exp(\mathrm{W}(x)) = x$ for $x > 0$. Of particular interest for us is the function $\exp(\mathrm{W}(x))$. Asymptotically, this function can be estimated by $\lim_{x\to\infty} \exp(\mathrm{W}(x)) = x/\ln(x)$. For example, consider the estimate of $\mathbf{E}[\Delta]$ for the subcase that arcs are very short, say $\alpha = \mathcal{O}(1/n)$. In this case, the characterization above gives $\mathbf{E}[L] = \mathbf{E}[\Delta] = \mathcal{O}((\ln n)/\ln\ln n)$. Furthermore, if $\alpha = \Theta(\frac{\log n}{n})$ then we obtain $\mathbf{E}[L] = \mathbf{E}[\Delta] = \mathcal{O}((\ln n)/\ln\ln n)$.

Finally, in some applications there might be arcs wrapping around the circle several times, i.e., $\alpha_i > 1$. Clearly, in this case our bounds for $\mathbf{E}[\Delta]$ transfer immediately when using $\alpha' = \frac{1}{n}\sum_i \alpha_i \bmod 1$ instead of α. In Section 2 we prove these bounds for the uniform case. The generalization to variable arc lengths is deferred to the full paper.

1.2 Results for Chains-into-bins

Many applications in computer science also require a discrete variant of arc allocation where the circle is subdivided into M equal bins and where arc end points are multiples of a bin size. We note that our proof techniques and hence our upper bounds directly transfer to this discrete model. Observe that discrete arc allocation is equivalent to the following *chains-into-bins* problem: $N = \sum_i \alpha_i$ balls connected into n chains are allocated to M bins that are arranged in a circle. A chain is allocated by throwing its first ball into a bin chosen independently, uniformly at random and by putting its remaining balls into adjacent bins in a round robin fashion. In this notation, the bounds in A) and B) become

$$\mathbf{E}[L^{(\mathrm{cb})}] = \Theta\left(\frac{N}{M}\,W^*\!\left(\frac{\ln\left(M\frac{n}{N}\right)}{N/M}\right)\right) \text{ if } N \geq M \ , \tag{1}$$

$$\mathbf{E}[\Delta^{(\mathrm{cb})}] = \Theta\left(\sqrt{\frac{N}{M}\ln\left(M\frac{n}{N}\right)}\right) \text{ if } N = \Omega(M\ln(Mn/N)) \text{ and } \alpha \leq M/2 \tag{2}$$

where $L^{(\mathrm{cb})}$ is the number of balls in the fullest bin and $\Delta^{(\mathrm{cb})} = L^{(\mathrm{cb})} - N/M$. In the way one can translate the results in the cases C) and D).

Let us compare our results for chains-into-bins to the well known results for balls-into-bins processes. These processes are among of the most intensively studied stochastic processes in the context of algorithm analysis (e.g., [10, 17, 12]). The simplest balls-into-bins process assumes that N balls are placed at random into M bins [10, 17]. Balls-into-bins are the special case of chains-into-bins where all chains consist of a single ball, i.e., $n = N$. We get

$$\mathbf{E}[L^{(\mathrm{bb})}] = \Theta\left(\frac{N}{M}\,W^*\!\left(\frac{\ln M}{N/M}\right)\right) \text{ if } N \geq M \ , \tag{3}$$

$$\mathbf{E}[\Delta^{(\mathrm{bb})}] = \Theta\left(\sqrt{\frac{N}{M}\ln M}\right) \text{ if } N = \Omega(M\ln M) \ . \tag{4}$$

The Bounds (3) and (4) are well known although other papers [10, 17, 13] use a different, slightly more complicated notation that yields more information about constant factors.

Another instructive perspective is that arc allocations are related to balls-into-bins systems with $1/\alpha$ bins. Our analyses for L and Δ will give further insights into the relationship between the two different random processes.

1.3 Previous Results

Barve et al. [2] introduce the chains-into-bins problem and show why several tail bounds for the case $N = n$ also apply to the general case. Apparently, $\mathbf{E}[L^{(\mathrm{cb})}]$

can only grow if chains are atomized into individual balls (although this is not proven yet). Our bounds improve these results by showing that $\Delta^{(\mathrm{cb})}$ can be much smaller if $n \ll N$, i.e., if chains are long.

Chains-into-bins have been analyzed asking what is the expected number of bins with at least a balls [11]. This measure was needed to estimate the number of cache misses in executing a class of cache-efficient algorithms. Refer to Section 3.3 for more details.

Arc allocations have been studied in mathematics under the aspect of when the arcs cover the circle (e.g., [15]). This is related to the minimum overlap which seems to be more important for most computer science applications. We have adopted the convention from these papers to measure arc lengths between 0 and 1 rather than 0 and 2π in order to avoid notational overhead.

An arc allocation defines a *circular arc graph* [9,8] with n nodes where there is an edge between nodes i and j if the corresponding arcs overlap. A set of overlapping arcs defines a clique of the circular arc graph. In this terminology, we are studying the size of the maximum overlap clique of a random circular arc graph. But note that the maximum overlap clique is not necessarily maximum clique of a circular arc graph [3].

2 Uniform Arcs

In this section we assume that all arcs have the same length. The following tail bound imply the expectation Bounds A) and B) respectively.

Theorem 1. *Suppose n arcs of length $\alpha \leq \frac{1}{2}$ are placed at random onto the unit circle. Let $\mu \geq \alpha n$ denote an upper bound on the average overlap. Then, for every $\epsilon > 0$,*

$$\mathbf{Pr}[\Delta \geq \epsilon\mu + 1] \leq n \left(\frac{\mathrm{e}^{\epsilon}}{(1+\epsilon)^{1+\epsilon}}\right)^{\mu} \tag{5}$$

$$\mathbf{Pr}[\Delta \geq 5\epsilon\mu] \leq \frac{6}{\alpha} \left(\frac{\mathrm{e}^{\epsilon}}{(1+\epsilon)^{1+\epsilon}}\right)^{\mu} . \tag{6}$$

Bound (5) that is best suited for short arcs is derived by bounding the maximum overlap by the overlap at the discrete set of starting positions of arcs. We defer the analysis to the full paper since simple Chernoff bound arguments are sufficient in this case.

Perhaps the most interesting case are rather long arcs with $\alpha < 1/2$. Section 2.1 derives Bound (6) that is a up to a factor $\Theta(n)$ more tight in this case. The proof combines Chernoff bound arguments with random walk arguments that may be interesting for other applications too. Bounds C) and D) for even longer arcs can be proven using an almost symmetric argument on the *minimum* overlap of non-arcs or *holes*. The proof is deferred to the full paper. In the full paper we furthermore argue that our results are essentially tight by giving lower bounds in terms of a balls-into-bins process considering $1/\alpha$ equally spaced positions on the circle. Section 2.2 reports simulation results that even give some hints as to what the constant factors in Bounds B) and C) might be.

2.1 Proof of Bound (5)

Proof. Define $\kappa = \lceil 1/\alpha \rceil \geq 2$. Let $x_0, \ldots, x_{\kappa-1}$ denote κ points on the circle that decompose the circle into κ intervals $X_i = [x_i, x_{i+1})$ of identical length $\lceil \frac{1}{\kappa} \rceil \approx \alpha$. (Here and in the following $i+1$ abbreviates $(i+1) \bmod \kappa$.) Observe that every arc has at most one endpoint in each interval. Define $\Delta_i = L(x_i) - \alpha n$ and $\Delta_i' = \sup_{x \in X_i}(L(x) - L(x_i))$. In this way, the maximum overlap in interval X_i is exactly $\alpha n + \Delta_i + \Delta_i'$. Our argument is based on the following two claims:

$$\forall \epsilon > 0 : \mathbf{Pr}[\Delta_i \geq \epsilon\mu] \;\leq\; \left(\frac{\mathrm{e}^{\epsilon}}{(1+\epsilon)^{1+\epsilon}}\right)^{\mu} \tag{7}$$

$$\forall \epsilon > 0 : \mathbf{Pr}[\Delta_i' \geq \epsilon(2+2\epsilon)\mu \mid \Delta_i + \Delta_{i+1} \leq 2\epsilon\mu] \;\leq\; 2\left(\frac{\mathrm{e}^{\epsilon}}{(1+\epsilon)^{1+\epsilon}}\right)^{(1+\epsilon)\mu} \tag{8}$$

Let us show that, in fact, these claims imply the theorem. First suppose $\epsilon \geq 1$. The maximum overlap in interval X_i is bounded above by $L(x_i) + L(x_{i+1}) = 2\alpha n + \Delta_i + \Delta_{i+1}$ because every arc overlapping with interval X_i covers x_i or x_{i+1}. This implies the inequality a1: $\Delta \leq \max_i\{\alpha n + \Delta_i + \Delta_{i+1}\}$ so that we obtain

$$\begin{aligned}\mathbf{Pr}[\Delta \geq 3\epsilon\mu] &\overset{(a1)}{\leq} \mathbf{Pr}[\exists_{i=0}^{\kappa-1} : \alpha n + \Delta_i + \Delta_{i+1} \geq 3\epsilon\mu] \overset{(a2)}{\leq} \mathbf{Pr}\left[\exists_{i=0}^{\kappa-1} : \Delta_i \geq \epsilon\mu\right] \\ &\overset{(a3)}{\leq} \frac{1}{\kappa}\left(\frac{\mathrm{e}^{\epsilon}}{(1+\epsilon)^{1+\epsilon}}\right)^{\mu} \overset{(a4)}{\leq} \frac{2}{\alpha}\left(\frac{\mathrm{e}^{\epsilon}}{(1+\epsilon)^{1+\epsilon}}\right)^{\mu} ,\end{aligned}$$

where inequality (a2) follows from $\alpha n \leq \epsilon\mu$, (a3) follows from Claim (7), and (a4) follows from $\kappa = \lceil \frac{1}{\alpha} \rceil$.

Now assume $\epsilon < 1$. Observe that this implies b1: $\epsilon \geq \epsilon(3+2\epsilon)/5$. Furthermore, we apply b2: $\Delta = \max_i\{\Delta_i + \Delta_i'\}$ and obtain

$$\begin{aligned}\mathbf{Pr}[\Delta \geq 5\epsilon\mu] &\overset{(b1)}{\leq} \mathbf{Pr}[\Delta \geq \epsilon(3+2\epsilon)\mu] \\ &\overset{(b2)}{\leq} \mathbf{Pr}[\exists_{i=0}^{\kappa-1} : \Delta_i \geq \epsilon\mu \vee \Delta_i' \geq \epsilon(2+2\epsilon)\mu] \\ &\overset{(b3)}{\leq} \sum_{i=0}^{\kappa-1} \mathbf{Pr}[\Delta_i \geq \epsilon\mu] + \mathbf{Pr}[\Delta_i' \geq \epsilon(2+2\epsilon)\mu \mid \Delta_i + \Delta_{i+1} \leq 2\epsilon\mu] \\ &\overset{(b4)}{\leq} \kappa\left(\frac{\mathrm{e}^{\epsilon}}{(1+\epsilon)^{1+\epsilon}}\right)^{\mu} + 2\kappa\left(\frac{\mathrm{e}^{\epsilon}}{(1+\epsilon)^{1+\epsilon}}\right)^{(1+\epsilon)\mu} \leq \frac{6}{\alpha}\left(\frac{\mathrm{e}^{\epsilon}}{(1+\epsilon)^{1+\epsilon}}\right)^{\mu} .\end{aligned}$$

The following basic fact from probability theory implies inequality (b3). For events X, X', and Y with $X' \subseteq X$ it holds $\mathbf{Pr}[X \vee Y] \leq \mathbf{Pr}[X] + \mathbf{Pr}[Y \setminus X'] \leq \mathbf{Pr}[X] + \mathbf{Pr}[Y|X']$. Furthermore, inequality (b4) follows from the Claims (7,8).

It remains to prove the two claims. Observe that $\mathbf{E}[L(x_i)] \leq \mu$ so that the bound in Claim (7) follows directly from a Chernoff bound. Hence it only remains to show the Claim (8). We will estimate Δ_i' by investigating the following random walk. Recall that each arc has at most one endpoint in interval X_i. Let m denote the number of those arcs that have an endpoint in X_i. As we condition on $\Delta_i + \Delta_{i+1} \leq 2\epsilon\mu$, we can assume

$$m \;\leq\; L(x_i) + L(x_{i+1}) \;=\; 2\mu + \Delta_i + \Delta_{i+1} \;\leq\; (2 + 2\epsilon)\mu \ .$$

For the time being, assume that m is fixed. Let $y_1, \dots, y_m$ denote the endpoints in X_i, sorted from left to right. If we are given these endpoints without further information then the *orientation* of the corresponding arcs (i.e., whether y_j is a left or right endpoint) defines a random experiment that can be described in terms of binary random variables as follows. Let $s_1, \dots, s_m$ denote random variables with

$$s_j \;=\; \begin{cases} +1 \text{ if } y_j \text{ is a left endpoint, and} \\ -1 \text{ if } y_j \text{ is a right endpoint.} \end{cases}$$

The only assumption that we made about the allocation of the arcs is $\Delta_i + \Delta_{i+1} \leq 2\epsilon\mu$. As this assumption does not affect the arc's orientation, the variables $s_1, \dots, s_m$ are independent and $\mathbf{Pr}[s_j = 1] = \mathbf{Pr}[s_j = -1] = \frac{1}{2}$, for $1 \leq j \leq m$.

Now let us define $S_j = \sum_{k=1}^{j} s_j$, for $0 \leq j \leq m$. Notice that the sequence $S_0, S_1, \dots, S_m$ corresponds to a random walk in which a particle starts at position 0 and goes up or down by one position in each step with probability $\frac{1}{2}$ each, that is, $\Delta_i' = \max_{0 \leq i \leq m}(S_i)$. Hence, we can estimate Δ_i' by analyzing this random walk. Applying Theorem III.7.1 from [6] we can derive the following probability bound. For every $r \geq 0$,

$$\mathbf{Pr}[\exists j\{1, \dots, m\} : S_j \geq r] = \sum_{k=r+1}^{\infty} \mathbf{Pr}[S_m = k] + \mathbf{Pr}[S_m = k+1] \leq 2\mathbf{Pr}[S_m > r].$$

Next we observe that the random variable $X_m = S_m/2 + m/2$ follows the binomial distribution and, hence, can be estimated using a Chernoff bound. In this way, we obtain

$$\mathbf{Pr}[S_m > \epsilon m] \;=\; \mathbf{Pr}[X_m > (1+\epsilon)m/2] \;\leq\; \left(\frac{\mathrm{e}^{\epsilon}}{(1+\epsilon)^{1+\epsilon}}\right)^{m/2} ,$$

for every $\epsilon > 0$. As a consequence,

$$\mathbf{Pr}[\Delta_i' > \epsilon m] \;\leq\; 2\mathbf{Pr}[S_m > \epsilon m] \;\leq\; 2\left(\frac{\mathrm{e}^{\epsilon}}{(1+\epsilon)^{1+\epsilon}}\right)^{m/2} .$$

Clearly, Δ_i' is monotonically increasing in m. Therefore, the worst-case choice for m is $m = (2 + 2\epsilon)\mu$, and we obtain,

$$\mathbf{Pr}[\Delta_i' > \epsilon(2 + 2\epsilon)\mu] \;\leq\; 2\left(\frac{\mathrm{e}^{\epsilon}}{(1+\epsilon)^{1+\epsilon}}\right)^{(1+\epsilon)\mu} ,$$

which proves Claim (8). Thus Theorem 6 is shown.

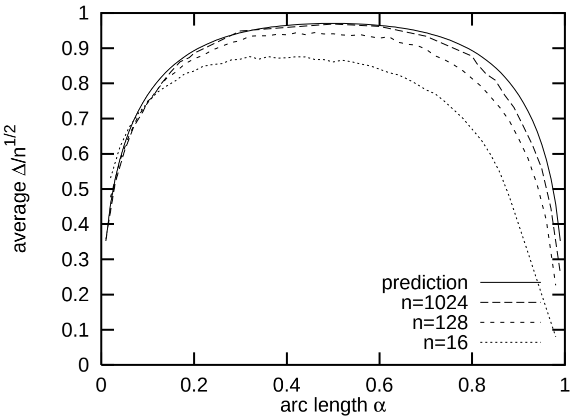

Fig. 1. Comparison of the theoretical prediction $\Delta = \sqrt{en \cdot \alpha(1-\alpha) \ln \frac{1}{\alpha(1-\alpha)}}$ with simulations for different n. The measurements are averages of 10000 repetitions for $n \leq 128$ and 1000 repetitions for $n = 1024$.

2.2 Simulations

Fig. 1 shows simulations for three different values of n and compares them with our analytic bound from the case B) and C) of our characterization for $\mathbf{E}[\Delta]$. Merging the bounds in these two cases we obtain the elegant estimate $\mathbf{E}[\Delta] \approx \sqrt{\alpha(1-\alpha) n \ln\left(\frac{1}{\alpha(1-\alpha)}\right)}$.

In fact, the measured and predicted curves are quite close together when we choose an appropriate constant factor (namely $\sqrt{e}$) in front of this estimate. Even for $n = 16$, where a significant influence of lower order terms can be expected, Δ is fairly well approximated. An interesting phenomenon is that the measured graphs for $\Delta(\alpha)$ are not completely symmetric around $\alpha = 1/2$. This asymmetry is not reflected by our theoretical analysis since we use the same Chernoff bounds to estimate the overlap of arcs in case of $\alpha \leq \frac{1}{2}$ and the overlap of holes in case of $\alpha > \frac{1}{2}$. In fact, however, the deviation of holes below the mean can be expected to be slightly less than the deviation of arcs above the mean, which explains the asymmetry that can be observed in the experiments.

3 Applications

We now present three examples, where our bounds on arc allocation yield performance guarantees. In Section 3.1, Bound (2) guarantees that rotational delays in accessing a single disks eventually become small compared to data access times. Section 3.2 gives a different disk scheduling application where load balancing data accesses over several disks is the objective. Whereas the first two example concern execution time, the last example in Section 3.3 bounds the memory consumption of a technique for reducing cache interference misses.

In all three applications very bad worst case behavior is avoided using randomization. The price paid is that the best case behavior gets worse. Since best case behavior may sometimes not be far from real inputs, it is crucial for our performance bounds to demonstrate that this possible penalty is very small. This is a quite strong incentive to study lower order terms in the performance bounds rather than bounds that leave the constant factors open.

3.1 Disks and drums

One of our main motivations for studying arc allocations was the desire to find disk scheduling algorithms that take rotational delays into account. For example, consider a data base query selecting the set $S = \{x \in r : x.a = y\}$ from a relation r. Assume we have an index of r with respect to attribute a that tells us a set of small disk blocks where we can find S. In this situation, the access time for retrieving S is dominated by rotational delays and seek times rather than the access or transmission time of the data [14]. Unfortunately, simultaneously minimizing seek times and rotational delays is NP-hard [1]. On the other hand, the explosive growth of disk storage densities in the last years suggests to consider the case where the accessed file fits into a narrow zone on the disk. In this case, seek times can be bounded by a constant that is only slightly larger than the overhead for request initiation, data transmission, and settling the head into a stable state. Such constant overheads can be absorbed into the size of the blocks and we end up with a problem where only block lengths and rotational delays matter. Interestingly, this reasoning leads to a model logically identical to *drums* — rotating storage devices from the 50s [4] that are technologically outdated since the 70s. Drums have a separate read/write head for every data track and hence suffer no seek delays.

To read a block on a drum one just has to wait until its start rotates to the position of the read head and then read until the end of the block. Suppose we want to read a batch of n blocks efficiently. Each block can be modeled as an arc in an arc allocation problem in the obvious way. Obviously, L is a lower bound for the number of drum rotations needed to retrieve all n blocks. Fuller [7,16] found optimal and near optimal drum scheduling algorithms that can retrieve the blocks in at most $L+1$ drum rotations. One such algorithm, *Shortest Latency Time First (SLTF)*, is very simple: When the drum is at point x, pick the unread block i whose starting point a_i is closest, read it, set $x = a_i + \alpha_i \bmod 1$ and iterate until all blocks are read.

The question arises, how good is an optimal schedule. In the worst case, n rotations may be needed. For example, if all blocks have the same starting point. In this case, even a good scheduling algorithm is of little help. Our results provide us with very attractive performance guarantees if the starting points are randomized. We need time $n\alpha + O(\sqrt{n})$ rotations and hence for large n, almost all of the schedule time is spent productively reading data.

Performance guarantees for random starting points need not merely be predictions for the average case if we randomize the mapping of logical blocks to physical positions. Here is one scheme with particular practical appeal: Start with a straightforward non-randomized mapping of logical blocks to physical positions by filling one track after the other. Now rotate the mapping on each track by a random amount. This way, accesses to consecutive blocks mapped to the same track remain consecutive. A technical problem is that starting points are not completely independent so that our analysis does not strictly apply. However, starting points of two blocks in a batch of blocks to be read are either independent or the two blocks do not overlap (we merge consecutive blocks on the same track). Therefore it seems likely that our upper bounds still apply.

3.2 Parallel disk striping

Assume a file is to be allocated to M parallel disks numbered 0 through $M-1$ so that accesses to any consecutive range of data in the file can be done efficiently. A common allocation approach is *striping* — block i of the data stream is allocated to disk $i \bmod M$. The situation gets more complicated if n files are accessed concurrently. Barve et al. [2] propose *simple randomized striping (SR)* where block i of a file f is mapped to disk $r_f + i \bmod M$ where r_f is a random offset between 0 and $M-1$. The number of I/O steps needed to access N blocks from the n data streams and M disks is the variable $L^{(\mathrm{cb})}$ in the corresponding chains-into-bins problem. Our results improve the performance bounds shown in [2] for the case that $n \ll N$.

3.3 Cache efficient allocation of DRAM memory

Many algorithms that are efficient on memory hierarchies are based on accessing n arrays in such a way that accesses in each array are sequential but where it is unpredictable how accesses to different arrays are interleaved. In [11] it is shown that these algorithms even work well on hardware caches where we have no direct control over the cache content provided that the starting points of the arrays modulo the cache size M are chosen at random. (Essentially the SR disk striping from [2] is applied to words in the cache.) Now the question arises how to allocate the arrays. Assume we have a large contiguous amount of memory available starting at address $x \bmod M$. A naive approach allocates one array after the other in arbitrary order by picking a random location $y \bmod M$ and allocating the array so that it starts at point $x + (y - x \bmod M)$. On the average this strategy wastes $nM/2$ of storage. A better way is by applying the simple SLTF algorithm we have seen for drum scheduling. This way we need space ML on the average wasting only $M\Delta$ of memory.

This bound for the offline algorithm also translates into a performance guarantee for an online allocation algorithm where requests for memory segments arrive one at a time. The following greedy algorithm can be easily proven to be 2-competitive: Keep a list of free memory intervals sorted by starting address. Allocate a new segment in the first free interval that has room at the right address offsets. Hence, space $2M(L+\Delta)$ memory suffices to fulfill all requests. In the final version of [11] citing our result replaces a lengthy derivation that shows a bound of $4eML$ for the online algorithm. No result on the offline case was available there.

Acknowledgements

We would like to thank Ashish Gupta and Ashish Rastogi for fruitful cooperation on disk scheduling algorithms.

References

1. M. Andrews, M. A. Bender, and L. Zhang. New algorithms for the disk scheduling problem. In IEEE, editor, *37th Annual Symposium on Foundations of Computer Science*, pages 550–559. IEEE Computer Society Press, 1996.
2. R. D. Barve, E. F. Grove, and J. S. Vitter. Simple randomized mergesort on parallel disks. *Parallel Computing*, 23(4):601–631, 1997.
3. B. Bhattacharya, P. Hell, and J. Huang. A linear algorithm for maximum weight cliques in proper circular arc graphs. *SIAM Journal on Discrete Mathematics*, 9(2):274–289, May 1996.
4. A. A. Cohen. Technical developments: Magnetic drum storage for digital information processing systems (in Automatic Computing Machinery). *Mathematical Tables and Other Aids to Computation*, 4(29):31–39, January 1950.
5. R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth. On the lambert W function. *Advances in Computational Mathematics*, 5:329–359, 1996.
6. W. Feller. *An Introduction to Probability Theory and its Applications*. Wiley, 3rd edition, 1968.
7. S. H. Fuller. An optimal drum scheduling algorithm. *IEEE Trans. on Computers*, 21(11):1153, November 1972.
8. M. R. Garey, D. S. Johnson, G. L. Miller, and C. H. Papadimitriou. The complexity of coloring circular arcs and chords. *SIAM Journal on Algebraic and Discrete Methods*, 1(2):216–227, June 1980.
9. V. Klee. What are the intersections graphs of arcs in a circle? *Amer. Math. Monthly*, 76:810–813, 1969.
10. V. F. Kolchin, B. A. Sevatyanov, and V. P. Chistiakov. *Random Allocations*. V. H. Winston, 1978.
11. K. Mehlhorn and P. Sanders. Scanning multiple sequences via cache memory. *Algorithmica*, 2002. to appear.
12. M. Mitzenmacher, A. Richa, and R. Sitaraman. The power of two random choices: A survey of the techniques and results. In P. Pardalos, S. Rajasekaran, and J. Rolim, editors, *Handbook of Randomized Computing*. Kluwer, 2000.
13. M. Raab and A. Steger. "balls into bins" – A simple and tight analysis. In *RANDOM: International Workshop on Randomization and Approximation Techniques in Computer Science*. LNCS, 1998.
14. C. Ruemmler and J. Wilkes. An introduction to disk drive modeling. *Computer*, 27(3):17–28, March 1994.
15. A. F. Siegel and L. Holst. Covering the circle with random arcs of random sizes. *J. Appl. Probab.*, 19:373–381, 1982.
16. H. S. Stone and S. F. Fuller. On the near-optimality of the shortest-access-time-first drum scheduling discipline. *Communications of the ACM*, 16(6), June 1973. Also published in/as: Technical Note No.12, DSL.
17. J. S. Vitter and P. Flajolet. Average case analysis of algorithms and data structures. In *Handbook of Theoretical Computer Science*, volume A: Algorithms and Complexity, chapter 9, pages 431–524. Elsevier, 1990.