Power Optimization in Disk-Based Real-Time Application Specific Systems

Inki Hong and Miodrag Potkonjak
Computer Science Dept., University of California, Los Angeles, CA

ABSTRACT

A flurry of modern communications, video, and DSP applications, such as world wide web, interactive high resolution television, video-on-demand, video conferencing, and wireless communications, are mainly data management oriented. In this type of systems, magnetic disk performance is quickly becoming a primary bottleneck which dictates key design metrics of the overall system. At the same time, technology trends imply reduced importance of classical design objectives, such as area and throughput, and together with consumers demand for portability promote power consumption as the principal design metric.

While numerous power optimization techniques have been proposed at all levels of design process abstractions for electronic components, until now, power minimization in mixed mechanical-electronic subsystems, such as disks, has not been addressed. We first analyze optimization degrees of freedom for power minimization in disk-based application specific systems. Next, we propose a conceptually simple, but realistic power consumption model for disk drives. We also developed six conceptually different sets of techniques for power minimization in disk-based systems.

The core of the paper are simulated annealing and traveling salesman problem based heuristics for optimization of power consumption in several common hard real-time disk-based design systems. We show how to coordinate tasks scheduling and their disk data assignment, so to minimize power consumption in both electronic and mechanical components of used disks. Extensive experimental results indicate significant power reduction ability of the proposed techniques and algorithms.

1.0 Introduction

1.1 Motivation: Why Synthesis of Disk-based Systems? Why Low Power?

In the last three decades, there have been constant stream of many massive storage alternatives, including charge-coupled devices (CCD), bubble memories, optical and magneto-optical disks, semiconductor disks, and holographic storage. Nevertheless, magnetic disks are the de-facto standard for providing high volume memory capacity in modern computer systems. Disks provide superior trade-off with respect to common design metric such as cost, memory capacity, latency, and data bandwidth.

Until recently, disks have been used mainly in general purpose computing systems. However, convergence of several application and technological trends, resulted in the last couple of years in the rapidly increasing importance of massive storage systems in application specific systems. On one side, there is rapid growth in applications such as internet-based servers (e.g. world wide web), video-on-demand, interactive television, video conferencing and wireless communications, all of which have as dominating components large volume data management. On the other side, all technological trends indicate that key design metrics of modern designs, such as speed, power, and cost, are increasingly more dominated by massive storage elements. For example, while modern microprocessor, ASIC, and semiconductor memory technologies are advancing at exponential rates of more than 50% compounded annually, performance of disks has been improving less than 10% compounded annually.

Both conceptual analysis and experimental measurement indicate that magnetic disk is already a bottleneck in current computing systems. Disk timing performance is dependent on the speed at which its sizable mechanical components physically move, and magnetic disks are inherently significantly slower than other system components, such as the processor, which are electronic in nature.

Another equally pronounced consequence of the current application and technological trends is increasing importance of power minimization. The length and level of operation for many types of portable communication, computation, and consumer electronic devices is limited by the weight of batteries. In addition cost of providing energy and even more associated cooling, reduced reliability and increased packaging cost of high power designs, additionally emphasize importance of power.

Our main strategic goal is to give impetus for research and development of synthesis techniques for design of massive storage application specific systems. We have three main technical goals:

1. To establish an accurate and computation-efficient power consumption model for disk-based systems.
2. To identify effective ways to reduce power in disk-based application specific systems.
3. To develop a practical approach and optimization synthesis algorithms for a scheduling and assignment of disk-based real-time systems.

Traditionally, both behavioral and system-level synthesis have been data-path and microprocessor focused. Only a few efforts addressed design of background memory [Cat94]. This is the first effort, to the best of our knowledge, which addresses power optimization of disk-based application specific systems.

1.2 Motivational Example, Replication Example, and Multiple-Disk Example

In disk-based real-time application specific systems, tasks may require data from a disk for its execution. Given N tasks with a specified access sequence to M disk data blocks, the total disk seek time depends on how the N tasks are scheduled and how the M blocks are stored. Given the throughput requirement, if the N tasks can be finished earlier than the deadline, the disk read and write time can be slowed down to reduce power consumption. Because more accurate power consumption model of disks is described in Section 4.1, its simplified version is explained here. Disks consist of the mechanical part (the spindle system) and the electronic part which account for roughly half of the power consumption, respectively. There are two degrees of freedom to reduce the power consumption by slowing down the disk read and write time. The first degree is the supply voltage of the electronic part which has a quadratic dependence on power consumption. The voltage can be lowered up to the point that the slowdown of the operation speed due to lower voltage still allows to meet the deadline, i.e., the reduced operation speed matches the reduced disk read and write time. The other degree of freedom is the motor speed of the mechanical part which has a linear dependence on power. We assume that the disk read and write time is inversely proportional to the spindle motor speed. The spindle motor speed can be slowed down to match the reduced disk read and write time.

The Figure 1 illustrates a simple instance of the power optimization problem. We consider a small example of 10 tasks where each task accesses one or two disk blocks among 10

	Disk Block		Disk Block
Task	Access Sequence	Task	Access Sequence
1	3	6	2, 3
2	6	7	5
3	3	8	4, 1
4	9, 1	9	5, 6
5	7	10	8, 1

Initial Power Dissipation(PD) = 1.51 W

Initial PD by the Spindle System = 700 mW

Initial PD by the Electronic Part = 810 mW

Initial Supply Voltage = 5.0 V

Initial Spindle Motor Speed = 5000 RPM

	Solution 1	Solution 2
Disk Block Assignment	<1,5,6,10,9,2,7,4,8,3>	<3,7,2,6,5,4,9,1,8,10>
Task Schedule	<2,8,6,9,5,10,1,7,4,3>	<10,4,7,9,2,6,1,3,5,8>
Disk Read/Write Time(ms)	30.0	30.0
Disk Seek Time(ms)	110.82	38.54
Schedule Length(ms)	140.82	68.54
Deadline(ms)	140.82	140.82
Scaled Schedule Length(ms)	140.82	140.82
Scaled Supply Voltage(V)	5.0	1.98
PD by the Electronic Part(mW)	810.0	126.9
Disk Read/Write Time(ms)	1.0	3.41
Spindle Motor Speed(RPM)	5000	1466
PD by the Spindle System(mW)	700.0	311.3
Total Power Dissipation(mW)	1510.0	438.2

Figure 1 A simple instance of the power optimization problem. PD - Power Dissipation.

available disk blocks for its execution. We assume that a single disk is used for the storage of all 10 data blocks and each block requires one disk track for its storage. We also assume that each disk access to a block involves a disk read and write. The seek time model and power consumption model described in the next sections have been used. Since Solution 2 enables that the execution of schedule finishes earlier than the deadline, disk read/write time can be slowed down. In the consequence, the voltage of the electronic part and the spindle motor speed of the mechanical part can be reduced to save power consumption. Even on this small example, a properly selected task schedule and disk assignment can improve the resulting power consumption by a factor of 3.45.

When the throughput requirement is so tight, the effective power reduction techniques used in the previous example can not be applied unless the disk seek time is further reduced. When disk writes are much fewer than disk reads, the replication of data blocks can be used to further reduce seek time with the expense of larger disk. We only consider the case that all the disk accesses are disk reads. The problem we consider is that given a task schedule, find a best disk assignment while replicating some of disk data blocks. As documented in the

experimental result section, careful replication of data blocks significantly reduces the disk seek time and so it does the power consumption. For an example of 200 tasks with 220 tracks, our heuristic reduced the power consumption by 39.3% while using 10% more disk space.

It is common that the multiple disks are used for the storage of data blocks to exploit economies of scale. Some video-on-demand storage server contains more than 100 disk drives. We consider the following scenario. We are given a task schedule of long period. In the task schedule, only some of the data blocks are accessed. The blocks to be accessed in the schedule are called as active blocks. Because only a few blocks are active, the disks are spinning idle for the most of the time consuming a lot of power. We can put most of the disks in sleep mode by moving active data blocks into one or a few disks during the initial transient period while serving the disk accesses at the same time. In the example of 5 disks with 250 blocks, given a random disk assignment and a random schedule of period 1000 which accesses 30 blocks so that the active blocks can be moved to a single disk, the power consumption is reduced by a factor of 3.49 by using this technique.

1.3 Paper Organization

The rest of the paper is organized in the following way. We first present the background material about the disk technology and several disk timing models. Section 3 summarizes the related work. In Section 4 we discuss power optimization degrees of freedom for application specific disk-based systems, propose six techniques for power reduction and introduce our approach for power optimization. Section 5 contains synthesis and optimization backbone of the paper, where technical details of the approach for reducing power in hard real-time application specific systems are presented. After presenting extensive experimental results in Section 6, we conclude the paper by restating main contributions.

2.0 Background Material

In this section we first provide an overview of disk technology and power consumption source in a disk. After that, we briefly discuss the most popular timing models of a magnetic disk. We conclude the section, by explaining the selected hardware and computational models.

2.1 Disk Technology

Magnetic disks have become the de-facto standard for providing non-volatile high volume memory in modern computer systems. Disks are inherently much slower than electronic system components, such as the processor. Disk performance times are measured in milliseconds rather than nanoseconds. Therefore disk performance has become a major bottleneck in the performance of the entire system. Since frequent accessing of a disk can have profound effects on performance, developing efficient means of ordering and controlling disk access is crucially important. Any improvement in the time required to satisfy requests sent to the disk is generally mirrored by a proportional increase in overall performance. Note that this is not necessarily true for other system components which may spend a significant amount of time "waiting" on slower devices, such as the disk.

The major components of a magnetic disk are the platters and the read/write heads. The platters are circular surfaces with a metallic coating which allows them to retain a magnetic charge. Multiple platters are placed directly over each other, forming a cylindrical object. A gap is left between each of the platters to accommodate a read/write head. The read/write head floats just above the surface of the platter and, as the platter rotates beneath it, the head may detect (read) or alter (write) the charge of specific areas on the platter. All read/write heads are attached to an arm which moves the heads either toward the center or toward the edge of the platters.

Each platter is composed of a series of concentric circles called tracks. A track can be defined as the total area of a rotating platter that passes beneath a stationary read/write head. Thus, a head may read or write to any position within the current track without needing to move. Because track size decreases as you move toward the center of the platter, the memory capacity per track also decreases. The following values can be considered "typical" for most modern magnetic disks:

Storage capacity	1 - 2 gigabytes
Number of platters	14 -18
Platter size	1.3 - 8 inches (3.5, 5.25 inches most popular)
Number of cylinders	2000 - 2600
Seek time	9 - 15 milliseconds
Rotational latency	6 - 8 milliseconds
Platter rotation speed	3600 - 7200
Data transfer rate	50 - 70 megabits per second
Average access time	20 - 35 milliseconds (for 8k data block)

So why does accessing such a simple device take such a considerable amount of time? This problem arises because accessing a specific sector on the disk requires that the read/write head move to the correct track (seek time) and that the platter rotate until the desired sector is beneath the head (rotational latency). Only after this overhead has been paid can the actual transfer of data begin. The amount of time required to physically move the head and the platter is quite significant and can generally only be decreased to a certain point.

2.2 Power Consumption and Timing Characteristics of Magnetic Disks

Power required by a hard disk drive is consumed by its many different components. Mechanical devices, such as the motors used to spin the platters and move the arms, and electronic devices, used to control the disk's operation, demand specific amounts of power in order to function properly. To complicate matters even further, the power requirements of each component will vary with the current operational mode of the disk. Common operational modes with different power requirements are:

- Start-up: Platters accelerate from rest to full speed.
- Seek: Arms move read/write heads to a new cylinder.
- Read/Write: Data transfer occurs between a read/write head and a platter surface.
- Idle: Platters spinning at full speed. No head movement or data transfer.
- Standby: Platters stop spinning and heads are parked.
- Sleep: Platters stopped, heads parked and electronics operating at minimal level.

The power requirements of a typical disk drive are given in Table 2. Detailed explanation of power consumption mechanisms is given in [Har95] and [Wag92].

A seek moves the disk head from track to track. Recently, several techniques have been proposed for analytic and empirical modeling of access speed[Rue94, Pat90]. The common denominator in all of them is that longer distance which arm has to travel corresponds to larger time overhead. Typical seek times for an IBM disk are given in Table 1 [Pat90].

seek distance range [tracks]	seek time
1 - 50	$1.9 + \sqrt{\text{distance}}$ - distance / 50
51 - 100	8.1 + 0.044 * (distance - 50)
101 - 500	10.3 + 0.025 * (distance - 100)
501 - 884	20.4 + 0.017 * (distance - 500)

Table 1 Typical seek time for an IBM disk.

2.3 Computational and Hardware Models: Hard Real-Time Disk-based Application

Specific System

Computational and hardware models were selected mainly for the goal to cover as large as possible set of modern and future disk-based application specific systems. Figure 2 shows the view at the highest level of abstraction on a typical system. The system has three components: disk, cache memory, and processor. Processor by itself can have multiple processors and/or ASICs. Since for power minimization in both memory and processor several approaches are readily available [Sin95], we focus our attention on disk's power optimization. We assume that a disk is a separate unit, as it is always the case in industrial practice.

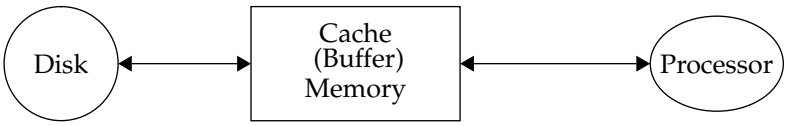

Figure 2 Architecture of a disk-based application system

We assume that each of tasks follows synchronous data flow semantics and syntax individually [Lee95]. Therefore, each task is periodic. Each task is fully specified by its period and its running time of one iteration of a given processor. We assume, with no loss of generality that all tasks have identical periods. When this is not the case, a simple preprocessing step, application of the least common multiple (LCM) theorem [Law81], in polynomial time transforms an arbitrary set of periods to an equivalent instance of this design scenario. Since context switching overhead is usually high for modern systems, we assume no task preemption. Note that this preemption restriction, actually does not impact any of the proposed methods, since in all discussed design cases non-preemptive policies yield superior results in comparison with preemptive policies.

Furthermore, each task has a need to read or read and write data to the disk. We assume that for each task a sequence of disk blocks to be accessed for its execution is given. Time to serve one read or write request is the sum of seek time and data transfer time. Seek time is proportional to distance which disk's head has to travel, and read and write time is proportional to amount of data which has to be transferred. The goal is to properly schedule all tasks and their required data transfers, so that all timing constraints are satisfied and disk's power consumption is minimized.

3.0 Related Work

In this section we first survey information about magnetic disks. Next, we summarize the most effective approaches for disk-scheduling and disk organization in general purpose computing and multimedia. We also briefly survey power optimization techniques. We conclude the section by presenting the pointers to the references which provide a detailed treatment of the algorithmic techniques which we used for optimization.

Although there have been constant stream of proposed alternative massive storage technologies, magnetic disks have dominated secondary storage since the mid sixties. Detailed description of magnetic disks can be found in many books [Hoa91, Jor96]. Another brief, but excellent exposition are papers [Rue94] and [Har95]. Relative comparison of magnetic disks with other massive storage alternatives is given on [Dou94].

The early disk-related research in operating systems has been focused on development of scheduling algorithms for efficient use of disks in time-shared mainframes [Teo72]. Later, operating systems researchers developed new disk scheduling algorithms for new general-purpose computing platforms assuming increasingly more realistic and complex disk models [Sel90]. While for long time disk scheduling algorithms were studied mainly in connection with time-sharing general purpose computing more recently focus has been on disk-based multimedia scheduling algorithms [Dai94]. Disk modeling recently attracted a great deal of interests [Pat90, Rue94, Har95]. While there is an abundance of macroscopic models for disks, there are surprisingly few detailed quantitative descriptions of mechanical, electrical, and magnetic components of a disk [Wag92].

Recently a number of synthesis and compilation techniques for power optimization at all levels of abstractions during design process have been proposed [Cha95, Sin95]. Although, power optimization is most effective at the higher levels of abstractions, until recently majority of power minimization techniques were proposed at logic synthesis and physical design phases of design [Sin95].

At the behavioral level Chandrakasan et al. were the first to show the effectiveness of transformations by showing an order of magnitude reduction in several DSP examples. Design trade-offs related to semiconductor background memory are used for power reduction by Catthoor et al. [Cat94]. A good survey of low power storage alternatives for general purpose mobile computing is given in [Dou94]. Several groups of authors proposed

approaches for power management in disks of mobile computers by using spin-down strategy and reported exceptionally high saving, with relatively little impact on perceived performances [Li94, Dou94]. Design issues which impact power consumptions are often confronted by designers of disk drives. References [Wag92] and [Har95] provide solid starting points to study key power-performance trade-offs during disk design.

In this paper, two effective combinatorial optimization techniques are used; the simulated annealing (SA) algorithm and the traveling salesman problem (TSP) heuristic.

SA has been proposed by Kirkpatrick et al. [Kir83] who combine iterative optimization with concepts and analogies from statistical mechanics; the resulting heuristic algorithm explores the configuration space of the optimization problem by allowing the hill climbing moves through probabilistic acceptance of disimproving moves. Since this seminal paper, SA has been a favorite approach for the optimization of hard combinatorial optimization problems mostly due to its simple implementation and wide applicability, the proof of optimality and the extensive empirical performance evaluation studies [Ree93].

The TSP problem, which is known to be NP-complete [Gar79], is probably the most widely studied problem in combinatorial optimization [Law85]. Through the extensive studies of the past on the symmetric TSP, a number of efficient and effective solution techniques for the symmetric TSP that achieve solutions within a few percentage points of optimal in practice have been developed [Joh90], including 3-Opt and Lin-Kernighan [Lin73] algorithms. Reinelt [Rei92] has reduced the running times of the algorithms with little loss of the solution quality by considering only the *k* nearest neighbors of each city. Martin et al. [Mar92] have given an effective way to improve the quality of the solutions on the earlier successful heuristics, which is called as the large-step markov chain(LSMC) heuristic. LSMC starts with an arbitrary initial local minimum and keeps improving on it by “perturbation + local optimization” cycles. LSMC can be seen as a variant of simulated annealing which moves among local minima only. Although a few solution methods for the asymmetric TSP exist, there do not exist enough experimental data to judge their performance [Law85].

4.0 Power Optimization Degrees of Freedom for Disk-based Application Specific Systems

In this section, we first introduce an elaborate power consumption model for a magnetic disk. After that, we propose six approaches for power minimization in disk-based

application specific systems. We conclude this section by describing our approach for power minimization in disk-based systems.

4.1 Magnetic Disks: Power Consumption Model

The first step in establishing a viable and relevant new research area is the development of proper models which accurately, but in tractable way, abstract key design metrics. However, though numerous models for electronics system are readily available, until recently, the disk drive studies have been using relatively simple and simplistic models. Recent article by Ruemmler and Wilkes [Rue94] clearly indicates the importance of using a calibrated, high quality disk drive model instead of simple first order model. They show that overall error by a factor of 14 is induced when non-adequate disk model is used.

After conducting extensive study of recent literature [Rue94, Har95, Jor96] and technical literature from major disk drive manufacturers [e.g. Sea96], we adopted the following disk drive power consumption model. During the selection of specific disk drive parameters, we tried to use the latest information for state-of-the-art disk in a conservative way; we always accepted specific data so that final resulting improvement in power are pessimistic.

The model separately considers two subparts: mechanical and electronic subsystems. Those two parts have two sharply different power dependencies. Figure 3, adopted from [Har95], shows five main components of modern disk drive. Four components except spindle motor, driver and controller are fully electronic parts.

The electronic part follows standard power trade-offs of CMOS-based designs. The sources of power consumption in a CMOS integrated circuit are due to four types of currents: leakage, standby, short-circuit, and capacitive. All the currents except capacitive can be reduced to a relatively low percentage of the total design power by a combination of proper design techniques [Sin95] and are mainly independent from the synthesis tasks related to architectural and application design of disks electronic subsystems. Therefore, the power consumption can be quantified using the following widely quoted equation: $P_{elect} = \alpha * C * V_{dd}^{2} * f$ where α is the activity factor, C is average capacitance switched per cycle, V_{dd} is the supply voltage, and f is the cycle frequency, assuming that V_{sw} the switched voltage is equal to the supply voltage. For power-delay dependency, we use the 6th order Nevine's rational polynomial approximation proposed and experimentally verified by Chandrakasan et al. [Cha95].

The spindle (motor) part is modeled using a simple linear model. Note that usually more

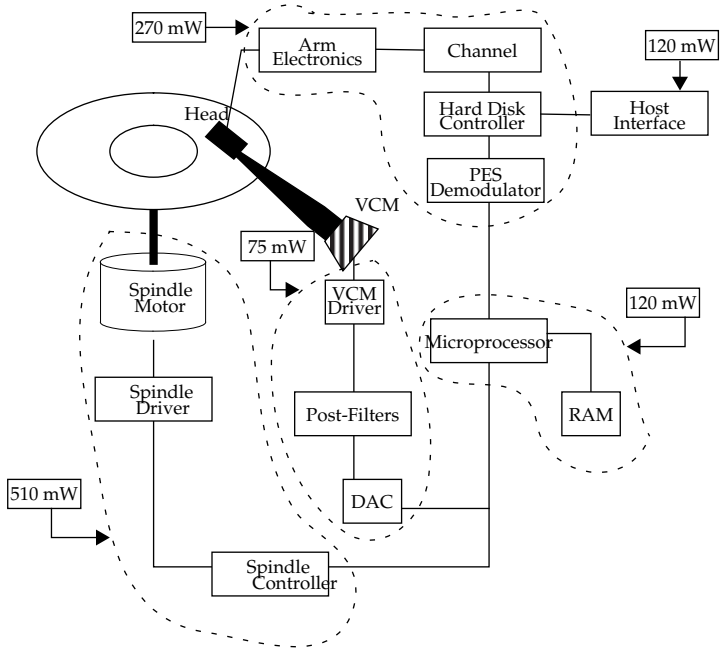

Figure 3 Schematic diagram showing the contributions to idle-mode power dissipation;

than 50% of power of a disk is spent in four electronics components. Even in spindle, controller is fully electronic part which usually consumes nontrivial percentage of power. There are two different sets of spindle controllers and motors, and they have very different power distribution, since more powerful and flexible motors, which require more energy, require less accurate controllers and vice versa. Since our optimization methodology better performs on systems which have higher percentage of power consumption in electronics parts, we will assume the most pessimistic scenarios, where all energy of spindle mechanism and control is spent in mechanical parts. Elaborate measurements [Har95] show linear dependency between rotational spindle motor speed and power. In particular, we use the following formula, derived from [Har95]: $P_{disk} = P_{fs} - \gamma^* (nrs - ors)$, where P_{disk} is power consumption of the disk which operates on operating rotations speed, denoted *ors*, γ is constant scaling coefficient, P_{fs} is power consumption at nominal rotational operating speed, denoted by *nrs*.

Again, we selected parameters in this formula, to follow our conservative estimation of improvements in power consumption. We used the following values in our experimentations: $P_{fs} = 700\ mW$; $\gamma = 110\ mW/1000\ rpm$; and $nrs = 5000\ rpm$.

4.2 Power Reduction Approaches in Disk-based Application Specific Systems

In this subsection we outline six ways to reduce power in a disk-based application specific system. While the first five approaches do not require redesign of available disks, the last is based on principles which can be used only during design phase of a disk. Note that approaches are not fully orthogonal; often the most effective way to reduce power is to combine optimization mechanisms of two or more approaches.

4.2.1 Reducing/Eliminating Read/Write Operations

The first approach is using reduction of data transfers from and to disk. Since disk consumes usually less than half energy in idle mode than in read/write mode, the reduced traffic results in reduced power. Scheduling. transformations and large buffer memory are key techniques which can explore this optimization degree of freedom. Several transformational techniques have significant potential for reducing average required number of data transfers from or to disk. For example, retiming enables more efficient block processing which regularly results in significantly reduced read/write requirements.

4.2.2 Reducing Seek Time through Head Movement Reduction

Again, this optimization principle is based on exploring low power consumption of a disk in operation modes where certain disk's functionality is not used. Modern disks spend almost twice less power in idle mode than in seek mode. In order to reduce seek time, system-level designer has to carefully arrange data assignment to tracks of disk as well as to properly order data access pattern. Therefore, task-level scheduling is most relevant synthesis task for exploring this trade-off. The advantage of this method is that no hardware alternations are required. However, as it is shown in this paper once the seek time is reduced it is usually more beneficial to reduce disk read/write so that the voltage and the spindle motor speed can be reduced (see Section 4.2.4) than to exploit this trade-off.

4.2.3 Spinning Down and Up Techniques

Spinning down and up disk is based on observation that disk in standby and sleep mode consumes approximately an order of magnitude less power than in seek and read-write modes. Even when compared to idle mode, this mode consumes significantly less power.

The spinning down/up approach is the basis for several power optimization approaches in general purpose mobile computers [Li94, Dou94]. The key idea is to cluster several requests for data, so to reduce frequency of disk accesses when typical user sessions are analyzed.

However, this technique has a limited potential in application specific systems. Since the spin down/up technique induces a significant timing and power overhead, the application domain of the approach is restricted only to systems with very flexible timing constraints, mainly general purpose computers. It is difficult to envision single disk-based real-time systems which can tolerate latency of several seconds and which do rarely access data. When multiple disks are used, however, the spin down/up technique combined with the clustering of active data blocks into a subset of disks is an effective approach to power optimization in this paper.

4.2.4 Lowering Voltage in Electronic Subsystems and Reducing Rotational Speed in Mechanical Subsystem

It is well known that due to quadratic dependency, often the most effective technique for power minimization in CMOS designs is increasing throughput of ASIC designs, so that voltage can be reduced [Cha95]. As it is explained in Section 4.1, lowering voltage in electronic parts and reducing speed in mechanical parts of a disk result in proportional power reduction, but also in slower data transfer from and to disk, and slower data processing. In order to enable proper functionality and timing of overall application specific systems under new conditions, as enabling step the system design has to reduce required speed of the disk. This can be done either by eliminating read/write operations or by reducing seek time. In this paper, we explore several effective and efficient approaches to reduce seek time, and therefore enable disk functioning under reduced speed requirements, what eventually significantly reduces power in the disk as well as in the overall application specific system.

4.2.5 Disk Sizing Techniques

The power consumption for a given generation of disk drives is proportional to their storage capacity. Table 3 provides a specific set of data for 1995 generation of IBM's disks. Note that the largest disk provides 18 times more storage capacity (720 MB vs. 40 MB), but consumes only 2.6 times more power (4.2 W vs. 1.6 W). If all data is accessed sequentially, there is no strict throughput constraints, and data is placed so that no seek time overhead is induced, obviously larger disks are better choice. In more realistic pattern usage scenarios, however, it is easy to see that multiple smaller disk can provide better performance-power trade-off.

	3.5-in	2.5-in	1.8-in	1.3-in
Capacity(MB)	720	340	85	40
Start-Up(W)	8.0	4.7	3.0	2.2
Seek(W)	4.5	2.0	1.5	1.7
Read/Write (W)	4.2	2.8	1.725	1.6
Idle(W)	2.5	1.35	0.75	1.0
Standby(W)	0.8	0.4	0.035	0.5
Sleep(W)	0.1	0.2	0.02	0.015

Table 2 Power requirements of hard disk drives in operation modes.

4.2.6 Alternation during Design and Assembly of a Disk Drive

Finally, during design of a disk drive, one has a number of opportunities to significantly impact the power-performance curve of the disk. There is a very intricate and complex trade-off between power consumption and speed when different volumetric, magnetic, and current constraints are applied [Wag92] during the design of moving coil actuator. Moving coil actuators (voice coil motors) are used for positioning the magnetic read and write heads in modern disks.

We decided not to pursue this avenue for power reduction due to the following two reasons. The first is that it requires an extensive alternation during disk manufacturing process, which reduces chances of application in practice. The second is that disk's technological trends are rapidly changing and with them qualitatively different trade-offs result in between electrical, magnetic and mechanical characteristics of moving coil actuators, key components which impact disk access time [Wag92].

4.3 Our Approach for Power Minimization

We now summarize our approach to power minimization. The key idea is to minimize seek time using proper scheduling and data assignment algorithms so that disk read/write time can be slowed down to result in the opportunities of exploiting power optimization degrees of freedom; the voltage of the electronic components can be reduced and the spindle motor speed of the mechanical component can be slowed down. In addition to the task scheduling and disk assignment, the replication of data blocks in the context of read-only disk accesses has been employed to further reduce the disk seek time. When the throughput requirement is so tight and the task scheduling and disk assignment technique can not generate slack for the disk read/write time reduction, the replication technique effectively reduces the disk seek time so that the resulting disk read/write time reduction reduces power consumption. Because it is not uncommon that application specific systems use multiple disks for the economies of scale, the power optimization problem for multiple

disks has also been studied using such effective techniques as disk spin-down of subset of disks and disk seek time minimization by disk assignment in the context of a disk-cluster where only a subset of all data is required at any given moment.

5.0 Power Optimization Problem

In this section, we discuss the power optimization under throughput requirement in two different contexts; (i) a single disk is used for the storage of all the data blocks of the tasks. (ii) multiple disks are used to store the data blocks of the tasks.

5.1 Single Disk: Problems Formulation, Graph-Theoretic Abstraction, Complexity, and Optimization Strategy

First, we formulate the three different versions of the power optimization under throughput requirement using the disk seek time minimization, spindle motor speed scaling and voltage scaling problem, and then show how to reformulate them into graph-theoretic problems. Next, we establish the computational complexity of the problems, by showing that all the problems are NP-complete. Lastly, we provide our optimization strategies for the problems.

5.1.1 Problems Formulation

We present the formulations of three different versions of the task scheduling and disk assignment problem to optimize power consumption in the standard Garey-Johnson [Gar79] format.

Problem: The Power Optimization Under Throughput Requirement Using Disk Seek Time Minimization, Spindle Motor Speed Scaling and Supply Voltage Scaling Problem.

Version 1: The Task Scheduling Problem

Instance: Given a set of M tasks described by the disk block access sequence, an initial voltage V, an initial spindle motor speed S, a disk block assignment in a disk and positive constants D and P.

Question: Are there a static periodic schedule of the M tasks, a new spindle speed S' and a new voltage V' such that the disk seek time + read/write time is at most D and the power consumption is at most P?

Version 2: The Disk Assignment Problem

Instance: Given a set of M tasks described by the disk block access sequence, an initial voltage V, an initial spindle motor speed S, a static periodic schedule of the tasks and positive constants D and P.

Question: Are there a disk block assignment, a new spindle motor speed S′ and a new voltage V' such that the disk seek time + read/write time is at most D and the power consumption is at most P?

Version 3: The Task Scheduling and Disk Assignment Problem

Instance: Given a set of M tasks described by the disk block access sequence, an initial voltage V, an initial spindle motor speed S and positive constants D and P.

Question: Are there a disk block assignment, a static periodic schedule of the tasks, a new spindle motor speed S' and a new voltage V' such that the disk seek time + read/write time is at most D and the power consumption is at most P?

5.1.2Graph-Theoretic Abstraction

The disk seek time minimization problem, which is a subproblem of the power optimization problem can be reformulated using the graph-theoretic abstraction. The task scheduling problem is transformed into an instance of the asymmetric TSP problem. Because all the possible task schedules have the same disk seek time within tasks, we need to optimize only the seek time between tasks. The mapping from the problem to the TSP is straightforward. For each task, a city is introduced. The disk seek time matrix between tasks becomes the distance matrix for the TSP. Then, the minimum tour of the TSP is the best schedule of the problem. The disk assignment problem is reformulated using the graph-theoretic abstraction in the following way. The given task schedule of disk block accesses can be described by a undirected weighted graph $G(V,E)$ of blocks. The weight $W(i,j)$ of an edge between blocks i and j is the number of times that the two blocks i and j are adjacent in the schedule. Let $D(i,j)$ denote the disk seek time between the blocks i and j in a disk. The total disk seek time for a given schedule is the sum of $W(i,j)*D(i,j)$ over all (i,j) in E. When a graph G is not weighted, each block requires only one track for its data and the distance $D(i, j)$ is the number of tracks between the blocks i and j, the problem is reduced to the Optimal Linear Arrangement problem which is NP-complete [Gar79]. For a special case when the graph of n nodes is a tree, there exists an optimal algorithm whose running time is O(nlogn)[Ado73].

5.1.3Computational Complexity

Since the power optimization problem includes the disk seek time minimization problem as a subproblem, the computational complexity of the three versions of the disk seek time minimization problem is discussed in this section. We proved that all three problems, using the standard Karp's reduction technique, are NP-complete. Due to the space limit, only the proof for the disk assignment problem is described in the following, as an illustration.

The proof uses the standard approach for establishing NP-completeness of a problem [Gar79]. The first part of the proof that the restricted problem belongs to the class of NP problems is straightforward. For the second part of the proof we use the Karp's polynomial reduction technique. We transform a known NP-complete problem, Optimal Linear Arrangement [Gar79], into an instance of our problem.

Problem: Optimal Linear Arrangement

Instance: Graph $G = (V, E)$, positive integer K.

Question: Is there a one-to-one function $f: V \rightarrow \{1, 2, ..., |V|\}$ such that $\sum_{\{u, v\} \in E} |f(u) - f(v)| \leq K$?

Suppose an arbitrary graph $G = (V, E)$ *is given*. We first polynomially transform the graph to a schedule for our problem. For each node in V, the schedule contains a corresponding block. The idea is that for each edge in E the corresponding two blocks are adjacent each other four times in the schedule and for any other edge of a complete graph not in E the corresponding two blocks are adjacent twice. Starting from node 1 to node $|V|$, all the edges in a complete graph are considered in the order of the node numbers so that any edge (u, v) with $u < v$ is not considered. For each edge, a corresponding part of the schedule is generated in the following way. If the edge (u, v) is in E and $u+1 = v$, then u,v,u is added in the schedule. If the edge (u, v) is in E and $u+1 \neq v$, then u,v,u,v,u is added in the schedule. If the edge (u, v) is not in E and $u+1 \neq v$, then u,v,u is added in the schedule. At the end of the schedule, $|V|$-1, $|V|$-2, ..., 2, 1 is added in the schedule. When $\sum_{\{u, v\} \in E} |f(u) - f(v)|$ is K for a one-to-one function f, the disk seek time for the transformed schedule and the same disk assignment solution as the function f is $2K + \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} |i - j| = 2K + |V| * (|V| - 1)$, when each block is stored in a track and the disk seek time between two blocks i and j is the number of tracks

between them. It is now easy to see that if we are able to achieve a disk assignment solution that has the resulting seek time of $2K + |V|(|V| - 1)$ then a one-to-one function f which is exactly the same as the disk assignment solution satisfies $\sum_{\{u, v\} \in E} |f(u) - f(v)| := K$. Therefore, the disk assignment problem is NP-complete.

5.1.4Optimization Strategy

We solve the power optimization problems in two steps. First, we find a task schedule and/or a disk assignment such that the disk seek time is minimized. Next, a voltage scaling and a spindle motor speed scaling are performed such that the throughput requirement is met.

Since the computational complexity of the disk head movement minimization problem forbids an exact or optimal solution, effective heuristic methods have been developed for the problem. The task scheduling problem is transformed into a TSP problem and an efficient and effective TSP heuristic is applied to the transformed problem. For the disk assignment problem, the simulated annealing (SA) algorithm has been used. The task scheduling and disk assignment problem employs a reiterative heuristic which repeatedly solves the task scheduling problem and the disk assignment problem separately using their TSP and SA heuristics until no improvement is achieved.

As shown in the earlier section, the task scheduling problem is transformed into an instance of the asymmetric TSP problem. Due to the current status of the available asymmetric and symmetric TSP heuristics, we transform an asymmetric TSP into a symmetric one by the technique proposed by Jonker and Volgenant [Jon83]. This allows us to apply the efficient and effective symmetric TSP heuristics directly. Let TSP(C,D) denote an asymmetric TSP with a set of n cities $C = \{1, 2, ..., n\}$ on a distance matrix D. Let TSP(C′, D′) denote the transformed symmetric TSP problem. Each city i in C is converted into two cities, i and $i + n$, in C'. If $|i-j| = n$, i.e., the two cities i and j in C' are from the same city in C, $D'[i][j] = - M$, where M is a very large positive number. If ($i \le n$ and $j \le n$) or ($i > n$ and $j > n$), $D'[i][j] = \infty$. Otherwise, $D'[i][j] = D[i][j]$ if $i > j$ and $D'[i][j] = D[j][i]$ if $i < j$. They show that any optimal solution of the transformed symmetric TSP corresponds to an optimal solution of the original asymmetric TSP.

The LSMC heuristic with the Reinelt's nearest neighbor-based fast implementation ($k = 25$ was chosen) of the 3-Opt local optimization engine has been used. The quality of the TSP

tour can be traded off with the running time by adjusting the number of the 3-Opt local optimization algorithms applied. In this paper, we used 100 iterations of 3-Opt local optimization algorithms for all the examples.

For the disk assignment problem, a simulated annealing (SA) algorithm has been employed. The SA algorithm is described in the following pseudo-code.

```
Generate a starting solution S.
Set the best solution S* = S.
Determine a starting temperature T.
while not yet frozen,
  while not yet at equilibrium for the current temperature,
    Choose a random neighbor S' of the current solution.
    if cost (S') < cost(S)
      Set S = S'
      if cost(S) < cost(S*)
        S* = S'
    else
      Generate a random number r uniformly from [0, 1].
      if r < e^(-(cost(s') - cost(s))/T)
        S = S'
  Update the temperature T.
Return the best solution S*.
```

The actual implementation details are presented for each of the following areas; the cost function, the neighbor solution generation, the temperature update function, the equilibrium criterion and the frozen criterion. Firstly, the disk seek time + read/write time for the given task schedule and disk assignment has been used as the cost function in the simulated annealing algorithm. Secondly, the neighbor solution is generated by the interchange of two blocks. Thirdly, the temperature is updated by the function $T_{new} = \alpha(T_{old}) * T_{old}$. For the temperature $T > 200.0$, α is chosen to be 0.1 so that in high temperature regime where every new state has very high chance of acceptance, the temperature reduction occurs very rapidly. For $100.0 < T \leq 200.0$, $10.0 < T \leq 100.0$ and $1.0 < T \leq 10.0$, α is set to 0.8, 0.95 and 0.98 respectively so that the optimization process explores this promising region more slowly. For $T \leq 1.0$, α is set to 0.8 so that T is relatively quickly reduced to firmly converge to a local minimum. The initial temperature is set to 4,000,000 for the disk assignment problem and 1.0 for the task scheduling and disk assignment problem. The initial temperature for the latter problem is set very low because the initial solution given by a TSP heuristic is already very good, so the low initial temperature allows

the fast running time without losing solution quality. Fourthly, the equilibrium criterion is specified by the number of iterations of the inner loop. The number of iterations of the inner loop is set to 20 times of the number of blocks. Lastly the simulated annealing algorithm terminates when the temperature falls below 0.1.

For the disk assignment problem, replications of the data blocks are considered to further reduce the disk seek time with the expense of larger disk. We only consider the case that each block can be replicated only once. When a disk write follows every disk read, the replication does not reduce the disk seek time. The reduction of the seek time due to disk reads can not compensate the increase due to disk writes because both copies of a block need to be written. When disk writes are much fewer than disk reads, the replication can reduce the disk seek time. We assume that all the disk accesses are disk reads, common scenario in many applications. The disk seek time can be different depending on the initial head position of a first block in the schedule if the block has been replicated. In this case, the cost used for the simulated annealing algorithm is computed in the following way. The seek times are computed twice with the different initial head position. If the initial head position of next period with the first copy as an initial head position is different from that with the other copy as an initial head position, the cost is the average of the two seek times because the disk head moves by the same pattern every two periods or the disk head follows either pattern depending on the initial head position. Otherwise, the cost is the seek time of the pattern of which the initial head position of next period is the same as that of the current period because no matter where it starts, the disk head moves by the pattern after the first transient period.

We need to decide carefully which blocks to replicate because the selection significantly affects the final results. The selection process starts with an ordering of the nodes of the graph constructed from a given schedule. For each node i in the graph, a number is assigned to the node i which is the disk seek time between the node i and the nodes incident on the node i for the assignment of the blocks following the pipe organ algorithm. The pipe organ algorithm is one that the node i is placed in the center and the other terminal node of the edge with the highest weight is placed on the left end position from the node i, the terminal node of the edge with the next highest weight is placed on the right end position from the node i and the same process repeats for the remaining nodes. All the nodes with the degree higher than 2 are sorted in a decreasing order by the assigned number. The first k nodes

(data blocks) are selected for replication when there are spaces for k data blocks. If there are no more candidates for replication, the remaining disk space is not used. Experimental data have shown that this heuristic significantly outperforms a random replication algorithm which randomly chooses blocks to replicate.

For the task scheduling and disk assignment problem, a reiterative heuristic which repeatedly solves the task scheduling problem and disk assignment problem, separately. The heuristic is described using the following pseudo-code.

```
Generate a random disk assignment.
Apply the TSP heuristic to find a task schedule given the random disk
assignment.
Set the current schedule and assignment to the best-so-far solution.
Repeat
  Apply the SA algorithm to find a disk assignment given the current
  task schedule.
  If the new assignment does not improve upon the best-so-far, stops
  the loop and return the best-so-far.
  Apply the TSP heuristic to find a task schedule given the current
  disk assignment.
  If the new schedule does not improve upon the best-so-far, stops the
  loop and return the best-so-far.
```

5.2 Multiple Disks: Problem Formulation, Complexity, and Optimization Strategy

The power optimization problem using the spin-down and disk seek time minimization is formulated, the computational complexity of the problem is discussed and the optimization strategy for the problem is provided. First, we formally define the problem in the Garey-Johnson format.

Problem: The Power Optimization Under Throughput Requirement Using the Spin-Down and Disk Seek Time Minimization

Instance: Given K disks, N/K blocks stored per disk, a block schedule with access to M blocks, where $M < N$, a disk assignment and positive constants D and P.

Question: Are there a new disk assignment and a spin-down decision such that the disk seek time + read/write time is at most D and the power consumption is at most P?

The computational complexity of the problem is NP-complete because a special case of the problem, the disk seek time minimization problem for single disk, is NP-complete.

Since the computational complexity forbids an exact solution, heuristics have been developed. There are two issues to be resolved. The first is to determine the k target disks that just fit in all the active data blocks. Because it is too computation-intensive to get an optimal solution, our heuristic chooses the k disks with the maximum number of active blocks, and minimizes data block swaps. The second issue is how to partition the active blocks to the target disks and how to position them in each assigned disk. The simulated annealing algorithm with pairwise interchanges both within a disk and between disks as neighbor solution generation functions is used to simultaneously solve the two problems. Because the active blocks which are originally in the target disks should not be moved, only the interchanges between any other blocks are allowed.

6.0 Experimental Results

6.1 Single Disk without Replication

We have generated random examples by varying the number of tasks. The number of blocks and the schedule period are chosen to be the same as the number of tasks. We have tried the examples of 50, 100, 150, and 200 tasks. Each task accesses either one or two blocks. Each disk block access involves a disk read and write. The disk seek time + read/write time of the best random solution is used as a deadline. The initial power dissipation (PD), the initial PD by the spindle system, the initial PD by the electronic part, the initial supply

Task	Task Scheduling Problem				Disk Assignment Problem				Task Scheduling and Disk Assignment Problem			
	Random		Optimized		Random		Optimized		Random		Optimized	
	Ave.	Best	Ave.	Best	Ave.	Best	Ave.	Best	Ave.	Best	Ave.	Best
50	1424.9	1149.2	623.7	623.1	1481.6	1151.1	531.6	529.8	1501.0	1294.3	363.0	334.9
100	3541.6	3156.4	1715.7	1708.3	3376.5	2986.9	1129.7	1126.7	3555.5	3126.5	916.0	811.8
150	4574.1	4138.7	2223.5	2220.0	4499.7	3960.3	1816.3	1794.4	4598.1	4085.4	1145.4	1113.7
200	5896.7	5466.1	2914.4	2907.8	5449.5	4881.9	2407.1	2365.6	5910.1	5416.4	1579.4	1555.9

Table 3 The results for the disk seek time + read/write time minimization.

Task Scheduling Problem			Disk Assignment Problem		Task Scheduling and Disk Assignment Problem	
Task	Random	Optimized	Random	Optimized	Random	Optimized
50	1510.0	370.7	1510.0	346.9	1510.0	297.4
100	1510.0	336.6	1510.0	307.7	1510.0	286.4
150	1510.0	347.1	1510.0	331.9	1510.0	297.2
200	1510.0	351.6	1510.0	353.6	1510.0	302.9

Table 4 The results for the power minimization using voltage scaling and spindle motor speed scaling.

voltage, the initial spindle motor speed and the initial disk read/write time has been set to

1.51 W, 700 mW, 810 mW, 5.0 V, 5000 RPM and 1.0 ms, respectively. The Tables 3 and 4 illustrate the effectiveness of the power optimization using disk seek time minimization, voltage scaling, and spindle motor speed scaling. The power consumption reductions by factors of 4.30, 4.51 and 5.10 are achieved for the task scheduling problem, the disk assignment problem and the task scheduling and disk assignment problem, respectively.

6.2 Single Disk with Replication

We make the same assumptions as the Section 6.1 except that each disk access involves only a disk read. The deadline for the problem is the disk seek time + read/write time obtained by the simulated annealing algorithm without replication. Only in the context of the disk assignment problem, the replication is considered. We have used an example of 200 blocks. Tables 5 and 6 illustrate the effectiveness of the replication by showing the power consumption reductions by 33.9% and 54.2% are achieved with the replication heuristic while the disk size has increased by 10% and 25% respectively. The SA with the replication heuristic outperforms that with random replication by 27.4%.

Tracks	Random Solution		SA solution with random replication		SA solution with replication heuristic	
	Ave.	Best	Ave.	Best	Ave.	Best
200	4853.16	4188.57	2116.10	2074.60	2116.10	2074.60
220	4815.93	4269.17	2080.97	2035.53	1955.69	941.87
250	4522.11	4269.26	2012.65	1961.76	1831.16	1800.79

Table 5 The results for the disk seek time + read/write time minimization with replication.

Tracks	SA solution with random replication	SA solution with replication heuristic
200	1510.0	1510.0
220	1244.0	917.1
250	966.0	691.4

Table 6 The results for the power optimization with replication.

6.3 Multiple Disks

We assume that 5 disks are used to store the 250 data blocks and each block requires one disk track for its storage. We randomly assign and position the blocks to the disks. We have generated the random schedules which access only to 30 blocks by varying the schedule period. We have tried schedule periods of 1000, 2000 and 3000. We have used the following data to compute relative power consumption amount. The power requirements for seek, read/write, idle and sleep are 2.0, 2.8, 1.35 and 0.2 respectively [Har95]. The Table 7 shows that the power consumption reduction by a factor of 3.45 using the spin-down has been achieved.

Period	Without spin-down	With spin-down
1000	122347.87	35052.48
2000	242484.48	69047.59
3000	358746.44	107351.33

Table 7 The results for the power optimization with spin-down for multiple disks.

7.0 Conclusion

We studied a new problem of power optimization in disk-based application specific systems. We analyzed optimization degrees of freedom for power minimization in disk-based application specific systems and proposed a conceptually simple, but realistic power consumption model for disk drives. We developed six conceptually different sets of techniques for power minimization in disk-based systems.

The highlight of the paper is the new approach for power minimization in disk-based systems. Simulated annealing and traveling salesman problem heuristics are used as optimization mechanisms for power minimization in several common hard real-time disk-based systems design scenarios. We demonstrated how to coordinate tasks scheduling and their disk data pattern access and assignment, so to minimize power consumption in both electronic and mechanical components of used disks. Extensive experimental results indicate significant power reduction ability of the proposed techniques and algorithms.

8.0 References

[Ado73] D. Adolphson, T. C. Hu, “Optimal Linear Ordering”, SIAM J. Appl. Math., Vol. 25, No. 3, pp. 403-423, 1973.

[Cat94] F. Catthoor et al., “Golabal Communication and Memory Optimizing Transformations for Low Power Signal Processing Systems, VLSI SP Workshop, pp. 178-187, 1994.

[Cha95] A.P. Chandrakasan, M. Potkonjak, R. Mehra, J. Rabaey, R. Brodersen, “Optimizing Power Using Transformations”, IEEE Transactions on CAD, Vol. 14, No. 1, pp. 13-32, January 1995.

[Dai94] S.J. Daigle, J.K. Strosnider, “Disk Scheduling for Multimedia Data Streams”, Proc. SPIE High-Speed Networking and Multimedia Computing, Vol. 2188, pp.212-223, 1994.

[Dou94] F. Douglis, F. Kaashoek, B. Marsh, R. Caceres, K. Li, J.A. Tauber, “Storage alternatives for mobile computers”, First USENIX Symposium on Operating Systems Design and Implementation (OSDI), pp. 25-37. Berkeley, CA, 1994.

[Gar79] M. R. Garey, D. S. Johnson, “Computers and Intractability: A Guide to the Theory of NP-Completeness”, W. H. Freeman and Company, New York, 1979.

[Har95] E.P. Harris, S.W. Depp, W.E. Pence, S. Kirkpatrick, M. Sri-Jayantha, R.R. Troutman, “Technology Directions for Portable Computers”, Proc. of the IEEE, Vol. 83, No. 4, pp. 636-658, 1995.

[Hoa91] A. S. Hoagland, James E. Monson, “Digital magnetic recording” 2nd ed, John Wiley, New York, NY, 1991.

[Joh90] D. S. Johnson, “Local Optimization and the Traveling Salesman Problem”, In Proceedings of the

17th Intl. Colloquium on Automata, Languages and Programming, pp. 446-460, July 1990.

[Jon83] R. Jonker, T. Volgenant, "Transforming Asymmetric into Symmetric Traveling Salesman Problems", Operations Research Letters, Vol. 2, No. 4, 1983.

[Jor96] F. Jorgensen. "The complete handbook of magnetic recording", 4th ed, TAB Books, New York, NY, 1996.

[Kir83] S. Kirkpatrick, C. Gelatt, M. Vecchi, "Optimization by Simulated Annealing", Science, Vol. 220, No. 4598, pp. 671-680, 1983.

[Law81] E. L. Lawler, C.U. Martel, "Scheduling periodically occurring tasks on multiple processors", Information Processing Letters, Vol. 12, No. 1, pp. 9-12, 1981.

[Law85] E. L. Lawler, J. K. Lenstra, A. Rinnooy-Kan, D. Shmoys, "The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization", Wiley, Chichester, 1985.]

[Lee95] E.A. Lee, T.M. Parks, "Dataflow Process Networks", Proc. of the IEEE, Vol. 83, No. 5, pp. 773-799, 1995.

[Li94] K. Li, R. Kumpf, P. Horton, T. Anderson, "A Quantitative Analysis of Disk Drive Power Management in Portable Computers", USENIX Symposium on Operating Systems Design and Implementation pp. 279-291, 1994.

[Lin73] S. Lin, B. W. Kernighan, "An effective Heuristics Algorithm for the Traveling-Salesman Problem", Operations Research, Vol. 31, pp. 498-516, 1973.

[Mar92] O. Martin, S. W. Otto, E. W. Felten, "Large-Step Markov Chains for the TSP Incorporating Local Search Heuristics", Operations Research Letters, Vol. 11, No. 4, pp. 219-224, 1992.

[Pat90] D.A. Patterson, J.L. Hennessy, "Computer Architecture: A Quantitative Approach", Morgan Kaufmann, San Mateo, CA, 1990.

[Ree93] C. R. Reeves, "Modern Heuristic Techniques for Combinatorial Problems", John Wiley & Sons, Inc., New York, 1993.

[Rei92] G. Reinelt, "Fast Heuristics for Large Geometric Traveling Salesman Problems", ORSA Journal on Computing, Vol. 4, No. 2, pp. 206-217, 1992.

[Rue94] C. Ruemmler, J. Wilkes, "An introduction to disk drive modeling", IEEE Computer Magazine, Vol. 27, No. 3, pp. 17-28, 1994.

[Sea96] http://www.seagate.com/stor/table.html

[Sel90] M. Seltzer, P. Chen, J. Ousterhout, "Disk Scheduling Revisited", Proc. of USENIX, pp. 313-323, 1990.

[Sin95] D. Singh et al., "Power Conscious CAD Tools and Methodologies", Proc. of the IEEE, Vol. 83, No. 4, pp. 570-594, 1995.

[Teo72] T.J.Teorey, T.B. Tinkerton, A comparative Analysis of Disk Scheduling Policies, Communications of the ACM, Vol. 15, No. 3, pp. 177-184, 1972.

[Wag92] J.A. Wagner, "Access time minimization of a moving coil actuator with volumetric, magnetic and current constraints", IEEE Industry Applications Society Annual Meeting, pp. 148-155, 1992.