Scheduling of Biological Samples for DNA Sequencing

by

Yuwei Hu
B.Mgmt., Information Management and Information Systems
Zhejiang University, 2008

and

Chin Soon Lim
B.Eng, Electrical Engineering
National University of Singapore, 2006

Submitted to the School of Engineering
In Partial Fulfillment of the Requirements for the Degree of

Master of Science in Computation for Design and Optimization

at the

Massachusetts Institute of Technology

September 2009

Signatures of Authors ……...
School of Engineering
August 6, 2009

Certified by ……...
Stephen C. Graves
Abraham J. Siegel Professor of Management Science
Thesis Supervisor

Accepted by ……...
Jaime Peraire
Professor of Aeronautics and Astronautics
Director, Computation for Design and Optimization Program

Scheduling of Biological Samples for DNA Sequencing

by

Yuwei Hu and Chin Soon Lim

Submitted to the School of Engineering
on August 6th, 2009, in partial fulfillment of the
requirements for the degree of
Master of Science in Computation for Design and Optimization

ABSTRACT

In a DNA sequencing workflow, a biological sample has to pass through multiple process steps. Two consecutive steps are hydroshearing and library construction. Samples arrive randomly into the inventory and are to complete both processes before their due dates. The research project is to decide the optimal sequence of samples to go through these two processes subject to operational constraints. Two approaches, namely, heuristic and integer programming have been pursued in this thesis. A heuristic algorithm is proposed to solve the scheduling problem. A variant of the problem involving deterministic arrivals of samples is also considered for comparison purposes. Comparison tests between the two approaches are carried out to investigate the performance of the proposed heuristic for the original problem and its variant. Sensitivity analysis of the schedule to parameters of the problem is also conducted when using both approaches.

Thesis Supervisor: Stephen Graves
Title: Abraham J. Siegel Professor of Management Science

ACKNOWLEDGEMENTS

We are indebted to Professor Stephen C. Graves for his help, patience, time and guidance throughout the project. His thoughtful and powerful questions helped to expedite our problem solving process. His line of thought created a deep impression on us. Working with him has been an eye-opener for the both of us. His meticulous editing has vacuumed up all the errors. All remaining errors solely belong to the authors.

We owe an equally great debt to Karen Ponchner and Scott Steelman from the Broad Institute for their time and help. We would like to thank Karen for providing us with the opportunity to work on some of the interesting problems at Broad. Scott has patiently explained the whole sequencing process, answered our endless questions, generated data for us, tested our software, … . The list could go on and on. This thesis would not have existed without him.

Our next thank you goes to our sponsors, the Singapore-MIT Alliance and MIT. It has been a wonderful, enlightening, inspirational and truly fruitful experience to study at MIT.

We are thankful to the group of fun-loving students at CDO for bringing us joy and happiness during our stay in MIT. We thank Laura Koller for her help with the administration and hospitality during our stay in the United States.

The first author would like to express her gratitude to all of her classmates at CDO who gave her the possibility to complete this thesis. She wants to thank them for all their help, support, interest and valuable hints. Especially she is obliged to Jie Wang for her stimulating suggestions and encouragement. Finally, this thesis would not have been possible without the confidence, endurance and support of her family.

The second author is eternally grateful to his parents, sisters and brother for showing him unwavering support and patience throughout his life. Last but not least, he would like to show his appreciation to his wife, Jing Ma for her continued encouragement, love, understanding and belief in him.

Contents

List of Figures

List of Tables

1 Introduction

1.1 The Research Topic

In a DNA sequencing workflow, a biological sample has to pass through multiple process steps. Samples arrive randomly over time and are classified as paired end or fragment biological samples. Paired end samples have to pass through two consecutive steps in a sequencing workflow, namely, hydroshearing and library construction. Each sample is characterized by its type, its processing priority, its arrival date and its due date. Samples have to be sequenced before their due dates. At any point in time, there might be any number of samples of different types in process to be scheduled. This research project entails developing methods to schedule the biological samples for these two steps subject to operational constraints.

The work in this research has been done in collaboration with researchers at the Broad Institute. This chapter begins with a quick overview of the genome sequencing procedure at the Broad Institute and emphasizes the two genome sequencing processes that this research project focuses on. Finally, a section outlines the organization of this thesis.

1.2 The Broad's Genome Sequencing Platform

The Broad Institute is organized around scientific programs and platforms, and one such platform is Genome Sequencing Platform, which designs and carries out large-scale genome sequencing projects, together with groups throughout the Broad community. Genomes of interest include various organisms, such as human, mammals, fish, insects, fungi, plant, bacteria and viruses. The platform's major activities include high-throughput genome sequencing, genome finishing, sequencing informatics and project management, and the research project here aims at helping the project management team to keep the platform's many efforts organized and on track by applying operations research approaches and techniques. The next chapter will describe the specific research problem in more detail.

1.3 Work Flow of a Genome Sequencing Project

The genome of an organism is the hereditary information encoded in the DNA. A chromosome is an organized structure of DNA. A genome sequencing project seeks to determine the sequence of DNA in those chromosomes. A typical work flow of a Genome Sequencing Project at Broad is displayed in Figure 1.1:

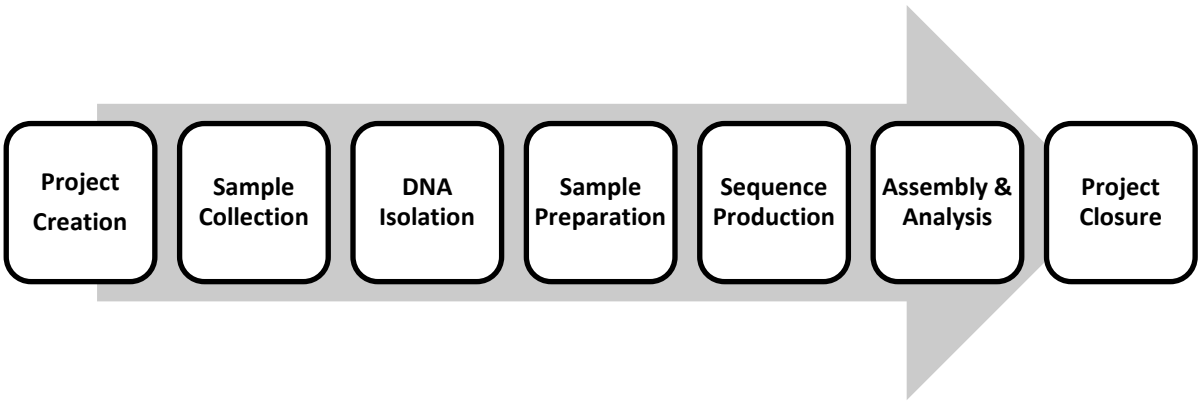

Figure 1.1 A typical work flow of a Genome Sequencing Project at Broad.

1.3.1 Project Creation

At the very beginning, scientists submit to a funding agency a white paper that describes the proposed study of the genome sequences of animals, plants or diseases. If the funding agency approves the project, a PASS will be created. A PASS is the detailed specification that documents all of the sequencing projects that are required to accomplish the overall proposed research project. The PASS includes the procedures for carrying out the projects of interest, including the expectations, plans and choices of technology etc. A PASS can consist of 50 sequencing projects; these projects are then allocated to various sequencing facilities, such as the Broad Institute. The focus of our research is on scheduling the sequencing projects that have been allocated to a sequencing facility, namely to the Broad Institute.

1.3.2 Sample Collection

Broad will liaise with scientists to collect the required DNA samples. Sequencing projects have to be completed within a time frame as stated in the PASS. DNA samples will arrive near the beginning of this time frame. Since DNA Sample Kits are collected by different collaborators, the time that the required samples arrive at Broad can vary a lot and is highly unpredictable.

A genome sequencing project can consist of many samples. A Work Request is generated for each sample and gives the detailed requirements on genome sequencing, including the types and amounts of the required DNA fragments and the determined technology.

1.3.3 DNA Isolation

DNA isolation is an extraction process of DNA from the various samples. After extracting the DNA from the samples, each extraction must undergo quality control (QC) before it goes to the next step, which is the preparation of samples for sequencing purpose.

1.3.4 Sample Preparation

Sample Preparation is necessary before the production of the genome sequence. Sample preparation ensures that the desired DNA fragments are separated and cloned. The actual procedure of sample preparation depends on the type of sample and on the choice of sequencing technology to be used. For example, the Roche-454 Sequencing System requires a specific preparation procedure called hydroshearing for a paired-end sample and, while a fragment sample can go to the next preparation step called library construction directly. If the constructed library does not pass quality control, a rework is required. Rework means that the sample has to undergo sample preparation again.

1.3.5 Sequence Production

In this step, the genome sequence is produced by the designated sequencing system, and the resulting sequence will undergo QC. A rework in this step might be required if the QC fails.

1.3.6 Project Closure

If the project result passes QC, the project and the corresponding Work Request will be closed, and the genome sequence will be uploaded to a website accessible by scientists.

1.4 Sample Preparation in Roche-454 System

The research problem arises in Sample Preparation in a Roche-454 sequencing system. The Roche-454 sequencing machine is the instrument that creates the final DNA sequence of the sample. Sample preparation consists of two procedures: hydroshearing and library construction. Hydroshearing is the process of fragmenting DNA samples using hydraulic action. A DNA library is a collection of DNA fragments and library construction refers to the process that creates the library.

Hydroshearing happens before library construction and it is required for paired-end samples, while fragment samples can skip hydro-shearing and go to library construction directly. In this problem, the focus is on the procedures for paired-end samples. DNA sequencing can be carried out from both ends of the molecules for a paired-end sample. More details about the DNA sequencing of paired-end samples can be found in [Illumina2008]. For the rest of this thesis, samples refer only to paired-end samples. Hence, these two words are used interchangeably throughout the rest of the thesis.

The research problem is to decide an optimal sequence of paired-end samples to pass through these two procedures while satisfying the different constraints in hydroshearing and library construction. The objectives that determine optimality and the different constraints are described in the next chapter.

1.5 Organization of Thesis

A detailed description of the problem with the objectives, constraints and assumptions are given in chapter 2. Chapter 2 also introduces some of the terminology and methods used in literature to tackle such a problem. A theoretical bound for the objective function under the relaxed problem is also provided in chapter 2. In chapter 3, a heuristic is proposed to solve the problem. The chapter will describe the algorithm in detail and provide an example of how the algorithm works. Chapter 4 touches on the integer programming method of the problem. Chapter 5 compares the performance and solution provided by the proposed heuristic with the integer programming method. Chapter 6 concludes this thesis with some suggested future work.

2 Problem Description

This chapter starts by giving an overview of similar scheduling problems that the research community has been trying to tackle. Next, some terminologies are introduced. Then, the scheduling problem that this project is trying to solve is described. After that, a section summarizes some of the specific techniques used to solve such a problem. Finally, the theoretical bound for achieving one of the objectives of the problem is derived.

2.1 Literature Review about Job and Batch Scheduling

In many manufacturing and assembly facilities, a number of operations have to be done on each job. Often these operations have to be done on all jobs in the same order implying that the jobs have to follow the same route. The machines are assumed to be set up in series, and the environment is referred to as a *flow shop* [Pinedo2002].

A somewhat more general machine environment consists of a number of stages in series with each stage having a number of machines in parallel. Each of these parallel machines is available to perform the same operation. Thus, each job has to be processed at each stage on only one of the machines. This machine configuration is often referred to as a *hybrid flow shop* [Caricato2007].

The hybrid flow shop scheduling problem has been widely discussed in the literature and has many industrial applications. The scheduling problem that leads to the minimum makespan tends to be NP-hard in general [Pinedo2002], with very few notable exceptions.

On the other hand, in many practical situations of contemporary manufacturing scheduling, it is either necessary or recommended to group jobs into batches for processing. In this context, there exist one or several *batch machines* or *batching processors*. A *batch machine* is a machine that can process a limited number of jobs simultaneously (Brucker1997). On the contrary, a *discrete processor* or a *discrete machine* can only process one job at a time (Ahmadi1992).

There are mainly two situations when batching can result in improved efficiency (Potts2000). The first situation is that a batch machine is capable of processing several jobs simultaneously. In this case, batches are formed according to overall production needs.

The second situation is that jobs may be batched if they share the same setup on a machine (such as their required tooling, color, container size, etc.)[Mosheiov2004]. This is often referred to as a *family scheduling model* [Wang2001], where jobs are partitioned into families according to their similarity. In this context, large batches have the advantage of high machine utilization because it reduces the time on setup. However, jobs with high priority in a different family may be delayed.

There are two variants of the family scheduling model depending on when the jobs become available. Under *batch availability*, a job only becomes available when the complete batch to which it belongs has been processed [Webster1995]. An alternative assumption is *job availability* (or known in the literature as item availability), in which a job becomes available immediately after its processing is completed [Potts2000]. Note that when the batch availability assumption is applied, the order of jobs in each batch does not affect job completion times. Due to the same reason, the processing time of a batch is determined by the longest processing time of the jobs in the same batch.

Usually, once the process begins, no job can be released from the batch machine until the entire batch is processed. Furthermore, each job has a certain size or capacity requirement, and the total size of the jobs in a batch cannot exceed the capacity of the batch machine.

Contemporary flow shop scheduling with batch machines frequently poses new challenges and difficulties in production planning. Therefore, solving these problems usually requires out-of-box thinking. This difficulty has motivated a number of solution methods. Frequently proposed approaches include variation or combination of sequencing rules [Smith1956], heuristics (such as genetic algorithms, simulated annealing) [Marimuthu2005], dynamic programming [Ng2007], mixed-integer programming [Stafford2002], and hybrids (methods that combine dynamic programming, or mixed-integer programming solvers with a heuristic). Some approaches such as TSP-based algorithms are also adapted to make them suitable for application to specific problems: an exhaustive literature review on TSP-based approaches for flow shop scheduling can be found in [Bagchi2006].

2.2 Terminology

A sample in this thesis refers to DNA samples that are to be sequenced. The basic unit of the sequencing process is a DNA sample. The research problem involves scheduling these samples for hydroshearing and library construction. All samples are labeled with unique sample identification numbers (*IDs*) and they are also classified under their different *types*. A type refers to a group of samples that share similar DNA characteristics. *Batch size* is the number of samples of a sample type for a project. Each sample goes through two different processes prior to DNA sequencing, namely, hydroshearing and library construction. *Work request date* refers to the date when the sample first arrives into the inventory prior to hydroshearing. Each sample has either a *high* or *standard* processing *priority*, which means that this sample must complete hydroshearing and library construction within two and three weeks for high and standard priorities respectively. The sample's *due date* is the work request date plus either two or three weeks depending on the priority of this sample.

A *hydroshearing schedule* consists of samples to be hydrosheared in sequence in the week. The number of *slots* of hydroshearing in each week is determined by the *capacity* for that week. One slot is required for each sample. A *library construction schedule* consists of *tasks*. Each task can contain up to 6 samples, and in effect corresponds to a

batch process. Each week library construction can complete a specified schedule of tasks. Samples are to be assigned to a specific task in the library construction schedule.

2.3 Hydroshearing and Library Construction Scheduling

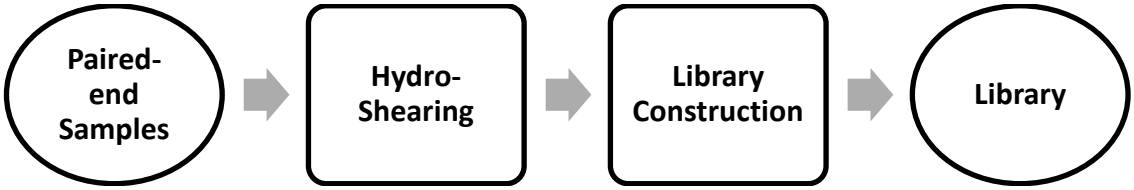

Figure 2.1 Procedures of Sample Preparation in a 454 system for paired-end samples.

In figure 2.1, the inputs of the Sample Preparation procedure are paired-end samples. The outputs of this system are libraries. A library is a collection of cloned DNA strains, usually from a specific organism.

The scheduling problem is (1) to decide the week that each sample should undergo hydroshearing and then (2) to assign each sample to a library construction task . This problem is further complicated by other issues. Firstly, samples are collected from various places around the world. Although the number of samples and the types that these samples belong to are known from the white paper, the actual arrival dates of samples are unpredictable. Moreover, samples of the same type do not necessarily arrive at the same time; instead, they can come in multiple batches. Secondly, hydroshearing is a shared resource for several different sequencing processes, and not just for the Roche-454 process. As a consequence, there is limited control over when hydrosheared samples are returned. Rather the Roche-454 process gets a capacity allocation from hydroshearing and then hydroshearing provides a one-week service time – that is, samples that arrive at the start of the week, up to the capacity limit, are processed by the end of the week.

This Sample Preparation system can be modeled as a two-stage flow shop manufacturing system. Thus hydroshearing is the first stage, while library construction is the second one. For each paired-end sample, there are two scheduled times: one for hydroshearing, and the other for library construction, which happens after the hydroshearing time. A schedule is a table that indicates the samples assigned for hydroshearing and library construction on every working day.

Since there are different sets of constraints in hydroshearing and library construction, the makespan, which is the total time taken for Sample Preparation (starting when the first sample begins hydroshearing until the last sample finishes library construction), can be affected significantly by how the samples are scheduled for these two procedures. This scheduling process has been done manually at Broad Institute.

Therefore, the objective of this research project is to develop an algorithm to assign each sample to hydroshearing and library construction so as to minimize the makespan, or

equivalently minimize the number of library construction tasks. In the following sections, the constraints, control parameters, objectives and assumptions are discussed in detail.

2.3.1 Constraints

Generally speaking, for any procedure in Sample Preparation, there are mainly four sets of constraints on the machines that perform Sample Preparation:
1) *Capacity constraints:* the maximum number of samples a machine can process in a week or day is limited. If the procedure processes several samples simultaneously (e.g. a batch machine), it also has a constraint on the batch size, which is the maximum number of samples that can be handled at a time. Note that this batch size constraint only exists when there are batch machines used during the procedures.
2) *Processing time* of a procedure: the time taken for a sample to finish a procedure.
3) *Time constraints*: each sample must complete both procedures before the due date.
4) *Compatibility constraints*: in order to avoid contamination, samples of the same type must be processed under strict constraints. If constraints cannot be met, some of the capacities might be wasted.

For this problem, the library construction procedure can be modeled as a batch machine, while the hydroshearing procedure is a discrete one (e.g. a classical machine that can only process one job at a time). Moreover, a sample is ready for scheduling as soon as the corresponding work request is generated.

Hydroshearing Constraints
In hydroshearing, samples from the same type cannot be processed consecutively. This is known as the compatibility constraint of the hydroshearing procedure. For example, 2 organisms of A cannot be processed consecutively, so if we have 4 samples for hydroshearing, 3 As and 1 B, the process should be A – B – A – W – A, where W means a wasted processing slot. As an example, assume that the hydroshearing capacity for the week is 20. Under the compatibility constraint, hydroshearing can process at most 20 samples if they can be alternated, or 10 samples if they are all from the same organism (in this case, there are 10 slots that are wasted).

At Broad Institute, booking of the hydroshearing capacity required for next week is done weekly. The specific samples that are going to undergo hydroshearing next week need not be specified when booking the capacity. The total number of remaining slots for the rest of this week on any day in this week is given by the difference between the capacity booked for this week and number of samples that have already been hydrosheared in this week. Realistically speaking, the number of remaining slots for the rest of the week is also affected by the day. For instance, if there are no hydroshearing of samples on the first 4 days of the week, the number of remaining slots on Friday is likely be less than the total capacity for the week. This is due to the fact that hydroshearing is a shared resource and the number of remaining slots on that Friday is also affected by the demand of other sequencing processes. Therefore, from Monday to Friday, the current capacity of hydroshearing for the rest of the week is constantly changing.

A sample can complete hydroshearing by the Friday in the week that it has been sent for hydroshearing. Moreover, samples can be sent to and collected back from hydroshearing team multiple times during a week given that the required slots are available.

Library Construction Constraints
In library construction, samples are grouped in tasks. A *task* is a combination of 6 samples. The compatibility constraint states that no two samples in each task should be of the same type. In this sense, the library construction procedure is a batch machine that can handle at most 6 samples of different types simultaneously. A full task refers to a 6-sample task, and each task takes 2.5 days to process, which is independent of the composition of it.

A technician is required to perform the library construction. If there are 3 technicians working on library construction and each of them can perform 2 tasks per week, the capacity of library construction is 6 tasks or 36 samples. Moreover, tasks will be assigned to each technician every week on Monday and Wednesday, and once a task begins, no changes can be made to the samples that are contained in that task.

The number of technicians that can perform library construction and the number of hydroshearing slots in a week are two input parameters of the problem.

Table 2.1 summarizes these constraints in hydroshearing and library construction.

	Hydroshearing	Library Construction
Capacity in each week	M samples	N tasks
Number of samples that can be of the same type	At most, M/2 samples in the week can be of the same type.	No two samples can be of the same type in each task.
Batch Size	NA	6 samples in each task
Processing Time for each sample	One week (from Monday to Friday)	2.5 days
Compatibility	Two samples of the same type cannot be processed consecutively.	All the samples in a task are of different types.

Table 2.1 Summary of Constraints in Hydroshearing and Library Construction.

2.3.2 Objectives

We are to decide the sequence that samples undergo hydroshearing and the assignment of samples to the tasks in library construction. There are two objectives that the solution should try to achieve. The first is to minimize the number of library construction tasks required. All samples should be assigned to a task and the total number of tasks to contain all these samples is to be minimized.

The second is to minimize the number of vacancies in the earlier tasks. This is required because once the day of library construction has passed, all unused capacity in that day will be wasted. Furthermore, by assigning as many samples as possible to the earliest

working days, there will be more vacancies in the later days of the schedule so that the new arrivals can be scheduled.

When the library construction tasks are created, the task processing time is independent of the composition of each task; consequently, the makespan is independent of the task sequence. Thus, the optimal library construction schedule should be the one that processes maximum number of samples in each task (maximum machine utility) and gives the minimum number of tasks. Therefore, in this problem, these two objectives are equivalent to minimizing the makespan of the set of samples.

2.3.3 Assumptions

The following assumptions are made about the problem:
1) Samples arrive dynamically, and the arrival dates are highly unpredictable. Hence, a proposed approach is to perform scheduling based on the samples that are in the inventory on each working day. There is no effort to predict or anticipate the arrivals of samples. This is a myopic scheduling in that we only schedule the work that has arrived into the system.

2) Although samples can be sent to and collected back from the hydroshearing team during different days of a week and some high priority samples can even finish the process in one day, we assume that samples to be processed in a given week are generally sent to hydroshearing on Monday, and collected on Friday. However, more samples can be sent during the week provided that there are still slots available. These samples will still be completed by the hydroshearing team by the Friday of the week.

2.4 Approaches Taken by the Research Community to Solve a Similar Problem

The problem as described in section 2.3 contains 2 main issues, namely, a stochastic arrival of samples to be processed and a resource-constrained scheduling of the existing samples.

The first issue is known as online scheduling in literature [Megow2006]. Online scheduling typically considers the problem of scheduling jobs that arrive over time to many identical machines. The second issue is known as resource-constrained project scheduling (RCPS) in literature [Brucker2001]. The most basic problem considered here is that of scheduling a deterministic set of activities with known duration to be processed by limited resources.

A few papers have tried to tackle problems that have some element of both issues. Elkhyari et. al. [Elkhyari2003] uses an explanation-based constraint programming technique with operation research algorithms to solve the timetabling problem. Elkhyari et. al. modeled the timetable problem as a RCPS but also provided an option to handle unexpected activities like a missing teacher or a slide projector breakdown. We have not pursued this direction because of the unfamiliarity with the technique used.

One paper that closely resembles the hydroshearing and library construction scheduling problem is written by Ruml et. al. [Ruml2005]. The problem involves on-line planning and scheduling jobs that arrive asynchronously over time. The objective is to minimize the total time required to finish all jobs. However, the approach used in the paper comes from the Artificial Intelligence domain and is out of the knowledge scope of this thesis's authors.

The next 2 papers relates to an approximate dynamic programming approach to solve problems of similar nature. Topaloglu and Powell [Topaloglu2006] deal with the dynamic resource allocation problem in the context of fleet management. The paper solves the problem of having to satisfy customers' demands for different kind of jets at different locations. A value function approximation is used in their approach. Choi et. al. [Choi2007] solves the problem of selecting and scheduling existing and potential research and development projects and their tasks using the Q-learning approach and Markov chain to model uncertain parameters. In another paper by the same authors [Choi2004], they use heuristics to generate a set of important states and solve the dynamic programming method over this confined state space. Besides papers from the Artificial Intelligence domain, the authors of this thesis do not know of any other papers that tackle an identical problem.

2.5 Comparison of Techniques

To solve this scheduling problem, various operations research approaches and techniques, including integer programming, dynamic programming (DP), approximate dynamic programming (ADP) and heuristics are explored.

The reasons behind using DP method are, firstly, decisions are made sequentially in time steps and DP is the natural way to go and secondly, ADP can handle the stochastic sample arrivals. A disadvantage of using DP method is that there is the problem of "curse of dimensionality". Another disadvantage is that the arrival dates of samples are hard to estimate. Hence, the proposed approach is based on the current set of samples and ignores arrivals of new samples. When new samples arrive, re-scheduling is done. This is the reason for making the first assumption in the section 2.3.

Applying a heuristic to solve the scheduling problem is another possible approach because of its simplicity. However, using a heuristic does not guarantee an optimal solution. Furthermore, a heuristic might not always create a feasible solution.

The static version of the scheduling problem can be formulated as an integer programming problem using binary decision variables. Binary variables are created for 1) each combination of samples and tasks and 2) each combination of samples and hydroshearing slot. These binary variables for each sample will take value 1 if the sample is assigned to that task or hydroshearing slot. As long as the integer programming method is feasible and can be solved, optimality of the solution is guaranteed. This is the advantage of using integer programming method as compared to using heuristic. If the

problem cannot be solved to optimality, a useful bound can also be found. The disadvantage of using integer programming is that the run time performance might suffer when the number of samples is increased or when the planning horizon of the scheduling is increased.

In this thesis, only the heuristic and integer programming approaches are pursued.

2.6 Theoretical Bound on the Minimum Number of Library Construction Tasks

This subsection derives the theoretical minimum number of library construction tasks required to contain all the samples.

Suppose that there are N types of samples, labeled as A, B, . . . , N. Furthermore, there are |A| number of type A, |B| number of type B, |C| number of type C and so on. Each of these samples must be assigned to a task. Each task can hold up to 6 samples and no two samples within the task can be of the same type.

Theorem 1. *The minimum number of tasks required to contain all these samples is given by*

$$\max\left\{|A|,|B|,\ldots,|N|,\left\lceil\frac{|A|+\ldots+|N|}{6}\right\rceil\right\}$$

Proof. Scenario 1
Assume that the minimum number of tasks required is |A|, i.e.

$$\max\left\{|A|,|B|,\ldots,|N|,\left\lceil\frac{|A|+\ldots+|N|}{6}\right\rceil\right\}=|A| \qquad (*)$$

This also implies that $|A| \geq |B|$, $|A| \geq |C|$, … , $|A| \geq |N|$ and

$$|A| \geq \left\lceil\frac{|A|+\ldots+|N|}{6}\right\rceil$$

This last inequality also means that

$$|A| \geq \frac{|A|+\ldots+|N|}{6}$$

or simplifying, $|B| + \ldots + |N| \leq 5|A|$.

For each sample of type A, a task is created. The first slot in each task is assigned to one sample of type A. We number these tasks 1, 2, |A|. Next, samples of type B are assigned

to the second slots of these existing tasks, starting with task 1. Since $|B| \leq |A|$, each sample of type B can be assigned to a different task. Continuing from where we left off, samples of type C are assigned to slots in the tasks, starting from the next available second slots of these existing tasks, namely task $|B| + 1$. If there are no more second slots available, we start with the third slot of the first task and continue to assign to the third slot. Again, no two samples of type C can end up in the same task because $|C| \leq |A|$. The process is repeated for all N types of samples. The very last type of sample will be able to fit into the existing available slots because $|B| + \cdots + |N| \leq 5|A|$.

Table 2.2 illustrates this assignment, where N = 9.

A	A	A	A	A	A
B	B	B	B	C	C
C	C	C	C	D	D
D	D	E	E	E	E
F	F	F	F	G	G
G	G	H	H	I	I

Table 2.2 An example illustrating assignment of samples to tasks.

If equation (*) holds, this is the minimum number of tasks required because if one of these tasks is removed and the samples in this removed task are assigned to the remaining tasks, the constraint that no two samples of the same type can be in the same task will be violated. Hence, we have proven that if equation (*) holds, the minimum number of tasks required is given by $|A|$. Using symmetry, the same argument applies for types B, C, . . . , N.

Scenario 2 Now, assume that

$$\max\left\{|A|,|B|,\ldots,|N|,\left\lceil\frac{|A|+\ldots+|N|}{6}\right\rceil\right\}=\left\lceil\frac{|A|+\ldots+|N|}{6}\right\rceil$$

For convenience, let

$$\left\lceil\frac{|A|+\ldots+|N|}{6}\right\rceil=n$$

Using the same methodology as before, n tasks are created. Next, the slots of the tasks are filled using the same method as mentioned earlier. As before, the constraint of no two samples of the same type can be in the same task is never violated because $|A| \leq n, \ldots, |N| \leq n$. Since the ceil of any number x is greater than or equal to x ($\lceil x \rceil \geq x$),

$$6\left\lceil\frac{|A|+\ldots+|N|}{6}\right\rceil\geq 6\left(\frac{|A|+\ldots+|N|}{6}\right)$$

$$6\left\lceil\frac{|A|+\ldots+|N|}{6}\right\rceil\geq |A|+\ldots+|N|$$

Hence, the number of slots available is greater than or equal to the number of samples and all samples can be packed into these tasks.

Table 2.3 illustrates this assignment, where N = 9, n = 3 and there are 14 samples.

A	A	B
B	C	C
D	D	E
E	F	G
H	I	

Table 2.3 Another example illustrating assignment of samples to tasks.

If one of the tasks is removed, there are not enough slots in the remaining tasks to accommodate the samples in the removed task. This is because even if there is no constraint on the types of samples in the same task, the minimum number of tasks required is n. The number of tasks cannot be decreased any further.

Therefore, we have proven that the minimum number of tasks required is given by

$$\max\left\{|A|,|B|,\ldots,|N|,\left\lceil\frac{|A|+\ldots+|N|}{6}\right\rceil\right\}$$

■

This proof establishes that the minimum number of tasks is given by the above expression and does not try to minimize the number of vacancies in the earlier tasks.

In addition, the proof does not consider due dates of the samples. The theoretical bound might not be reached when the due dates of the samples are taken into account. Consider the following example: suppose that there are 18 samples of different types that are due very soon and must be scheduled to the first day of library construction, which has 3 tasks. There are also 4 samples from one type that is different from those 18 samples. The theoretical minimum number of tasks required would be 4. The table below shows the library construction schedule using the theoretical minimum number of tasks required.

S	S	S	S
A	F	K	P
B	G	L	Q
C	H	M	R
D	I	N	
E	J	O	

However, because of the due dates, the minimum number of tasks required would be 7. The table below shows the library construction schedule when taking into account the due dates for this example.

A B C D E F	G H I J K L	M N O P Q R	S	S	S	S

2.7 Conclusion

This chapter has reviewed some of the relevant work carried out by other researchers. We define some terminology that will be important to understanding the rest of the chapters. The hydroshearing and library construction scheduling problem is described. Most crucially, the objectives, constraints and assumptions of the problem are listed. These objectives and constraints shape the development of the proposed heuristic that will be introduced in the next chapter.

3 Heuristics

3.1 Introduction

Traditionally, heuristics have been used to solve many optimization problems because of the intuitive nature and simplicity. Heuristics can find an approximate solution in a short time with minimal complexity. Memory and scaling issues that occur in other types of method do not occur when using a heuristic. Implementation of heuristic might not require sophisticated optimization software. Thus, applying heuristic methods might be cost effective.

On the other hand, applying heuristic methods to a problem does not necessarily give an optimal solution. Heuristics also might have trouble determining whether or not a feasible solution exists. In many cases, only a theoretical bound exists for the heuristic methods. A gap typically exists between the optimal objective value and the objective value produced by the heuristic.

In the scheduling problem here, the general idea of applying heuristic is to first plan the library construction schedule and then plan the hydroshearing schedule based on the library construction schedule. It is hard to perform both scheduling at the same time. Hence, it is simpler to fix one and create a schedule for the other. Since the library construction schedule has stricter and more complicated constraints, it makes sense to schedule the library construction tasks followed by the hydroshearing schedule. Once the library construction schedule is created, planning the hydroshearing schedule is simply to pick out samples in a chronological order in the library construction schedule.

This chapter is organized as follows. A terminology section introduces some of the terms used in the heuristic methods to solve the two stage scheduling problem. Following that, some of the possible heuristics to solve this problem are suggested and discussed. The next section introduces the two-phase heuristic approach. This section also illustrates the approach with an example. After that, a section describes the implementation of the two-phased heuristic in detail. Finally, a conclusion ends this chapter.

3.2 Terminology

This section lists the terminology required to understand the rest of this chapter in addition to that already defined in the problem description.

Besides the characteristics mentioned earlier, samples can thus be further identified by the *stages* that they are in. Figure 3.1 shows the three different stages of a sample. PRE_HYDROSHEARING stage implies that the sample has only just arrived and has not been hydrosheared. Samples in HYDROSHEARING stage refer to samples that have been sent for hydroshearing but have not returned from hydroshearing. Samples in POST_HYDROSHEARING stage are samples that have returned from hydroshearing.

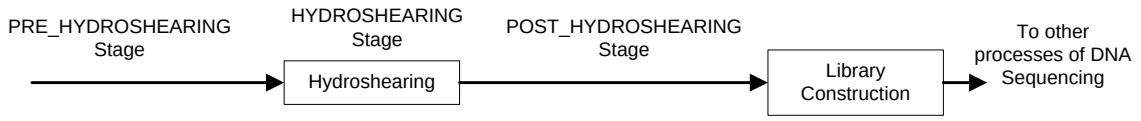

Figure 3.1 Possible stages that a sample can be in.

The *latest library construction date* is the date of the last library construction task by which the sample must be assigned before the sample is past due. The *earliest library construction date* is the date of the first library construction task that the sample can fit into after hydroshearing. As mentioned earlier in the section under assumptions, samples can be hydrosheared in the week that they arrive if the hydroshearing capacity and compatibility constraints are not violated.

A *library construction schedule by day* is a schedule that groups the samples according to the date when these samples will undergo library construction. A *library construction schedule by task* is a schedule that assigns samples in each day of the library construction by day into their respective tasks. A *hydroshearing week schedule* is a group of samples that will undergo hydroshearing in that week. Each sample will only occupy one *slot* of a hydroshearing week schedule. A *hydroshearing schedule* is a set of hydroshearing week schedules that specifies when samples that require hydroshearing are to undergo hydroshearing.

3.3 Simple Heuristics

Heuristics are simple to generate. Samples can be ordered by their work request dates or due dates in ascending or descending order. Scheduling this list of samples involves going down the list and assigning the samples to the earliest possible day. Or the samples can be ordered according to their types. Samples of the type with the largest batch size can head the list. And scheduling continues with this list of sample. The ordering method can also be the smallest batch size. The above mentioned heuristics start scheduling from the first library construction day. Another heuristic can be created by inverting the process. Scheduling can start from the latest library construction day. Samples are delayed until the very last moment before carrying out the library construction. This increases the probability of having newly arrived samples to find vacancies in the existing tasks. If there are already 6 samples in the task, there is no point in waiting and library construction should proceed for this task. Hybrid heuristic can also be created by combining some of these heuristics. For example, some of the 6 samples in the task can be chosen based on the earliest due date criterion and the rest by largest batch size. However, due to the complexity and uniqueness of the constraints in hydroshearing and library construction scheduling, some of these heuristics might not give a feasible schedule.

This section explains two of these simple heuristics that can be applied to the problem of library construction scheduling. The largest batch size heuristic has some similarities to the proposed solution in the next section. The feasibility, pros and cons of these two heuristics are also discussed. This section ends with an explanation of why the largest batch size heuristic minimizes the number of vacancies in the earlier library construction

tasks. This discussion explains why the sample of the type with the largest batch size is always chosen for assignment and shifting in the proposed heuristic. The proposed heuristic will be explained in the next section.

3.3.1 Earliest Due Date Heuristic

Samples are first sorted according to due dates in ascending order. The heuristic then goes down the sorted list starting from the top of the list and creates a new task based on the following rules:

Let S denote the number of samples, and we will use N as a variable to denote the total number of tasks

```
N := 1
For s = 1 to S

        n := 1
        While n <= N
                If s can be assigned to task n, then
                        assign s to task n, and go to next s
                Else, go to next n

        N := N+1
        Assign s to N

Next s
```

This heuristic shares all the benefits of applying heuristic methods to solving optimization problems as stated in the introduction to this chapter. However, this heuristic does not try to minimize the total number of library construction tasks required. A new task is required whenever a sample cannot fit into the earlier task. Hence, the order of the sample list will impact the number of tasks required. It merely tries to fulfill the due dates of the samples. A simple example will illustrate this point succinctly. Suppose that there are 8 samples, A B C D E F G G, ordered according to their due dates in ascending order. This heuristic will create 3 library construction tasks when 2 would suffice.

3.3.2 Largest Batch Size Heuristic

To understand this subsection better, we describe a library construction task again. A library construction task groups at most 6 different samples of different types. The capacity of the task is 6. The occupancy is the number of samples assigned to the task. The vacancy is the number of non-occupied spaces in the task that is not occupied. Hence, the vacancy equals the difference between capacity and occupancy.

Based on the list of samples, the largest batch size heuristic creates one task at a time using the following rules:
1) if there are at least 6 types of samples, we assign to the task one sample each from the 6 types with the largest number of remaining samples, else
2) if there are less than 6 types of samples, we assign to the task one sample from each type with remaining samples
3) Samples are removed from the list after being assigned a task. The above step is repeated until all samples have been grouped into tasks and there are no more samples left in the list.

Let the occupancy of task n be $d_n(numTypes_n)$ where $numTypes_n$ is the number of types of samples before grouping is carried out for task n. Since the number of types cannot increase after each task scheduling, $numTypes_n \geq numTypes_{n+1}$. Based on the previous rules for task scheduling, using the largest batch size heuristic will give the following occupancy in each task,

$$d_n(numTypes_n) = \min (6, numTypes_n)$$

Since $numTypes_n \geq numTypes_{n+1}$,

$$d_n(numTypes_n) \geq d_{n+1}(numTypes_{n+1})$$

This means that the occupancy of a previous task is always greater than or equal to the occupancy in a latter task. This achieves the objective of minimizing the number of vacancies in the earlier tasks.

The benefits of the largest batch size heuristic are that it is simple and easy for implementation. In addition, the library construction schedule by task created using this heuristic minimizes the number of vacancies in the earliest tasks as demonstrated in the previous paragraphs.

The con of the heuristic is that the due date might not be fulfilled. Suppose that there is a single sample of a type that is due soon and must be scheduled in the first task. By choosing the other samples with larger batch sizes, this sample cannot be scheduled into the first task. Hence, this heuristic might not create a feasible schedule. This disadvantage motivates an improvement on the heuristic.

3.4 Proposed Two-Phased Heuristic Approach

This section provides a general overview of the proposed heuristic. This heuristic improves on the simple heuristic outlined in the previous section because it creates a feasible schedule if one exists. It can be considered as a hybrid of the largest batch size heuristic and last minute heuristic. However, the difference is that there are two phases in this heuristic that seeks to improve on what these two simple heuristics cannot offer. More details required for implementation will be given in next section. This section will also illustrate the proposed heuristic through an example.

3.4.1 Brief Description of the Heuristic

Solving the problem of hydroshearing and library construction scheduling requires that each sample be allocated to a slot of hydroshearing and a library construction task. The constraints of this problem are time constraints, hydroshearing compatibility constraints, hydroshearing capacity constraints, library construction compatibility constraints and library construction capacity constraints. Time constraints refer to requirement that sample must complete library construction before a specific date. The other constraints have already been discussed in the section on problem description. The objective of the optimization problem is to minimize the vacancies in the earlier tasks and the total number of library construction tasks required. The solution should first satisfy all these constraints before trying to minimize the objective function.

The intuition behind the proposed heuristic is to break down the problem into subproblems. The first subproblem is to satisfy all the constraints and create a library construction schedule that is feasible under all the capacity constraints and time constraints. However, such a schedule should not be created randomly. A systematic approach is required. Hence, the sample is assigned to its latest library construction day where possible. If this is not possible because of a capacity constraint, the sample should be assigned to the day before its latest library construction day. The process is repeated until the sample is assigned a day. The reason for doing this is that by moving backward in time from the sample's latest library construction day, the due date constraint of the sample will not be violated. The sample will complete library construction before the due date since the assigned library construction date is always before the due date. Another benefit of going backwards in time is to check if a feasible schedule actually exists for this set of samples. A schedule might not exist when there are no days when the sample could be assigned to. More details of the feasibility of the problem are explained in the next and following section. Solving this subproblem is phase 1 of the heuristic.

After solving the first subproblem, the existing library construction schedule from phase 1 is feasible but not optimal. There might be vacancies in the earlier tasks because the heuristic has assigned samples to their latest library construction date. Capacity in the earlier tasks is wasted as a result. Hence, the second subproblem is to minimize the objective function subject to all the constraints. The objective calls for minimizing the vacancies in the earlier tasks and also minimizing the total number of tasks required. Hence, from the existing solution from phase 1, the heuristic tries to shift samples forward in time to the earliest possible day while continuing to satisfy all the compatibility and capacity constraints. The action of shifting forward in time is crucial to satisfying the time constraints. By shifting forward in time and never backward from its existing library construction date, there is no concern about violating the time constraints.

To summarize, the two-phased heuristic first assigns samples to their latest possible library construction day and then shifts these samples forward to their earliest possible library construction day. The two-phased heuristic plans the library construction schedule by day rather than by task. This is because each of the tasks in each day occurs

simultaneously. There is no difference in assigning the sample to the first or last task in that day. The only concerns are the compatibility and capacity constraints. Hence, this heuristic can only schedule samples into their library construction day and another function is required to schedule samples in each day into their tasks.

3.4.2 Illustration of the Two-Phased Heuristic Approach

This subsection illustrates the two-phase heuristic using an example. Suppose that today is Monday June 1, 2009. There are 3 tasks in each day. The total hydroshearing capacity for each week is 20. Hence, at most 10 of the samples in each hydroshearing week can be of an identical type. The set of samples is given in table 3.1. All samples are in the PRE_HYDROSHEARING stage. Each row of table 3.1 is interpreted as follows. For example, in row 1, there are 4 samples of type A with standard priority. Their work request dates are May 29. Since they are of standard priority, they must complete library construction at the end of three weeks. Thus, their due dates are 3 weeks from the work request date and are given by Jun 19.

Type	Number of samples	Priority	Weeks	Work Request Date	Due Date
A	4	Standard	3	May 29	Jun 19
B	3	Standard	3	May 27	Jun 17
C	2	Standard	3	May 28	Jun 18
D	2	Standard	3	May 22	Jun 12
E	2	High	2	May 27	Jun 10
F	1	High	2	May 27	Jun 10
G	1	High	2	May 27	Jun 10

Table 3.1 Samples for illustration of the two-phased heuristic.

The last due date is Jun 19. Hence, the planning horizon is 3 weeks. Table 3.2 below shows the associated dates of the hydroshearing weeks and library construction days over the planning horizon. There are no library construction tasks in the first week because samples have not undergone hydroshearing.

Hydroshearing Week (start date – end date)	1 (Jun 1 – Jun 5)						2 (Jun 8 – Jun 12)						3 (Jun 15 – Jun 19)					
Library construction day (start date – end date)							1 (Jun 8 – Jun 10)			2 (Jun 10 – Jun 12)			3 (Jun 15 – Jun 17)			4 (Jun 17 – Jun19)		
Tasks							1	2	3	4	5	6	7	8	9	10	11	12

Table 3.2 Dates of the hydroshearing weeks and library construction days for the example.

PHASE 1: Samples and their Latest Library Construction

In this phase, samples are assigned to the latest possible day. The sample of the type with the largest batch size is first chosen. In the event of a tie, the sample in an earlier stage is chosen. The rationale for choosing a sample according to these parameters will be given in the next section. In this example, the first sample to be assigned is of type A. The due date is Jun 19 which implies that the latest library construction day is 4. The table below shows the existing library construction schedule after this assignment. The number in the parenthesis behind the sample is to show that this sample is assigned to this day as the first assigned sample of phase 1.

Week	1		2		3	
Day			1	2	3	4
Date			Jun 8 – Jun 10	Jun 10 – Jun 12	Jun 15 – Jun 17	Jun 17 – Jun19
						A (1)

The list is updated after the previous assignment. Once again, the sample of the type with the largest batch size is chosen. Since there is a tie between samples of type A and B in both batch size and stage, sample of type A is chosen arbitrarily.

Type	Number of samples	Priority	Weeks	Work Request Date	Due Date
A	3	2	3	May 29	Jun 19
B	3	2	3	May 27	Jun 17
C	2	2	3	May 28	Jun 18
D	2	2	3	May 22	Jun 12
E	2	1	2	May 27	Jun 10
F	1	1	2	May 27	Jun 10
G	1	1	2	May 27	Jun 10

Week	1		2		3	
Day			1	2	3	4
Date			Jun 8 – Jun 10	Jun 10 – Jun 12	Jun 15 – Jun 17	Jun 17 – Jun19
						A (1) A (2)

The list is updated and now the next sample to be assigned is B.

Type	Number of samples	Priority	Weeks	Work Request Date	Due Date
A	2	2	3	May 29	Jun 19
B	3	2	3	May 27	Jun 17
C	2	2	3	May 28	Jun 18
D	2	2	3	May 22	Jun 12
E	2	1	2	May 27	Jun 10
F	1	1	2	May 27	Jun 10
G	1	1	2	May 27	Jun 10

Week	1		2		3	
Day			1	2	3	4
Date			Jun 8 – Jun 10	Jun 10 – Jun 12	Jun 15 – Jun 17	Jun 17 – Jun19
					B (3)	A (1) A (2)

Repeating the process for all samples, the table below shows the library construction schedule at the end of phase 1. Samples of type C have a due date of Jun 18 and hence, their latest library construction date is Jun 15. Since there are only 3 tasks in each day of library construction, only 3 samples of type A can fit into day 4. The remaining sample of type A has to be assigned to an earlier day, which is 3.

Week	1		2		3	
Day			1	2	3	4
Date			Jun 8 – Jun 10	Jun 10 – Jun 12	Jun 15 – Jun 17	Jun 17 – Jun19
			E (8) E (13) F (14) G (15)	D (7) D (12)	B (3) B (5) C (6) A (9) B (10) C (11)	A (1) A (2) A (4)

PHASE 2: Minimizing the Vacancies in the Earlier Days

In this phase, the heuristic tries to minimize vacancies in the earlier days by shifting forward the samples in the existing library construction schedule by day.

The sample of the type with the largest batch size is chosen for the shifting. In the event of a tie, the sample with the earliest due date is chosen. The rationale for choosing this sample will be explained in the next section. The original table of samples is repeated here for convenience.

Type	Number of samples	Priority	Weeks	Work Request Date	Due Date
A	4	Standard	3	May 29	Jun 19
B	3	Standard	3	May 27	Jun 17
C	2	Standard	3	May 28	Jun 18
D	2	Standard	3	May 22	Jun 12
E	2	High	2	May 27	Jun 10
F	1	High	2	May 27	Jun 10
G	1	High	2	May 27	Jun 10

From the list of samples, a sample of type A is the first sample to be shifted. This sample is assigned to library construction day 4 in the existing schedule. The earliest day that it can be shifted to is day 1 since the hydroshearing capacity for the week containing day 1 has not been reached. The table below shows the updated schedule after shifting this sample. Once again, the number in the parenthesis beside the sample refers to the order in which shifting has occurred. Samples without a number beside them are samples in the existing schedule.

Week	1		2		3	
Day			1	2	3	4
Date			Jun 8 – Jun 10	Jun 10 – Jun 12	Jun 15 – Jun 17	Jun 17 – Jun19
			E E F G A (1)	D D	B B C A B C	A A

The list is updated and the process is repeated. The next sample to be shifted is a sample of type B.

Type	Number of samples	Priority	Weeks	Work Request Date	Due Date
A	3	Standard	3	May 29	Jun 19
B	3	Standard	3	May 27	Jun 17
C	2	Standard	3	May 28	Jun 18
D	2	Standard	3	May 22	Jun 12
E	2	High	2	May 27	Jun 10
F	1	High	2	May 27	Jun 10
G	1	High	2	May 27	Jun 10

Week	1		2		3	
Day			1	2	3	4
Date			Jun 8 – Jun 10	Jun 10 – Jun 12	Jun 15 – Jun 17	Jun 17 – Jun19
			E E F G A (1) B (2)	D D	B C A B C	A A

After trying to shift all the samples, the final library construction schedule by day is shown in the table below. In some of the iterations, the sample is already at their earliest possible library construction date and no shifting occurs. Examples are samples of types E, F and G. Phase 2 of the heuristic also repeats the whole shifting process on the new schedule until no more shifting is possible.

Week	1		2		3	
Day			1	2	3	4
Date			Jun 8 – Jun 10	Jun 10 – Jun 12	Jun 15 – Jun 17	Jun 17 – Jun19
			E E F G A (1) B (2) A (3) D (4) B (5) C (6) A (7) D (8) B (9) C (10)	A (11)		

3.5 Detailed Description of the Hydroshearing and Library Construction Scheduling Algorithm

This section describes the implemented solution in detail. It begins with a general overview of the whole algorithm. Following that, the two-phased heuristic and the functions creating the library construction schedule by task and hydroshearing schedule are described. This section ends with a list of special cases that require attention. The methods of handling these special cases are also included. The subsections in this section

start by providing a flowchart of the described function. The subsequent paragraphs describe the processes in the flowchart.

3.5.1 Overall Algorithm Flow

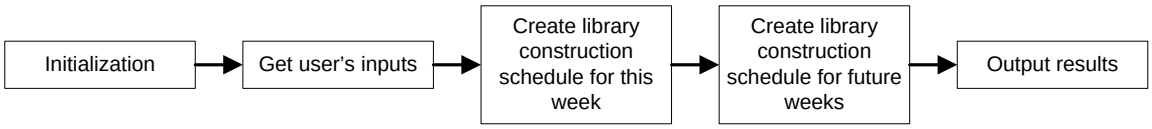

Figure 3.2 Overall algorithm flow.

Initialization:
The sample data are read in from the user input spreadsheet. The sample type is parsed from the first word in the sample's name. This sample type is then translated into a numeric for subsequent use in the program.

Get user's inputs:
The user is prompted to enter the number of hydroshearing slots remaining in the current week, the number of hydroshearing slots for each subsequent week, the number of library construction tasks remaining this week and the number of library construction tasks for each subsequent week. These are the control parameters mentioned in the previous chapter.

Create library construction schedule for this week:
Library construction schedule for this week is only created if today is a Monday, Tuesday or Wednesday. This is because there are tasks to be carried out on these days if today is a Monday or Wednesday or there are tasks to be carried out tomorrow if today is a Tuesday. Furthermore, this schedule only considers samples that have been hydrosheared, that is, samples that are in POST_HYDROSHEARING stage. Following that, a set of library construction dates is created, starting from first Monday or Wednesday from today and up to the latest task date by which the sample with the latest due date must be assigned to. For example, if today is a Monday and the sample with the latest due date is due 3 weeks from today, the first task date would be today's date and the latest task date would be the Wednesday 2 weeks from today. This takes into account the first week of hydroshearing.

After that, the actual library construction task schedule is created by first using the two-phased heuristic which will give a library construction schedule by day and then grouping samples in each day into tasks to give the final library construction schedule by task. Samples in the POST_HYDROSHEARING stage that cannot be scheduled into this week tasks are thrown into the mix of samples at PRE_HYDROSHEARING and HYDROSHEARING stages for scheduling at the next step.

Create library construction schedule for future weeks:
The procedure for scheduling samples at PRE_HYDROSHEARING, HYDROSHEARING stages and samples that could not be scheduled in the previous step is similar as before. A set of dates is created, starting from the next Monday and up to the latest task date by which the sample with the latest due date must be assigned to. The future library construction task schedule is again created by first using the two-phased heuristic which will give a library construction schedule by day and then grouping samples in each day into tasks.

Output results:
Finally, the results are output to file. The results include 1) a summary of all the samples and their library construction date and hydroshearing week if samples are in PRE_HYDROSHEARING stage, 2) the weekly hydroshearing schedule of samples at PRE_HYDROSHEARING stage and 3) the library construction tasks with their assigned samples organized into each day of library construction.

Figure 3.3 summarizes most of the above details.

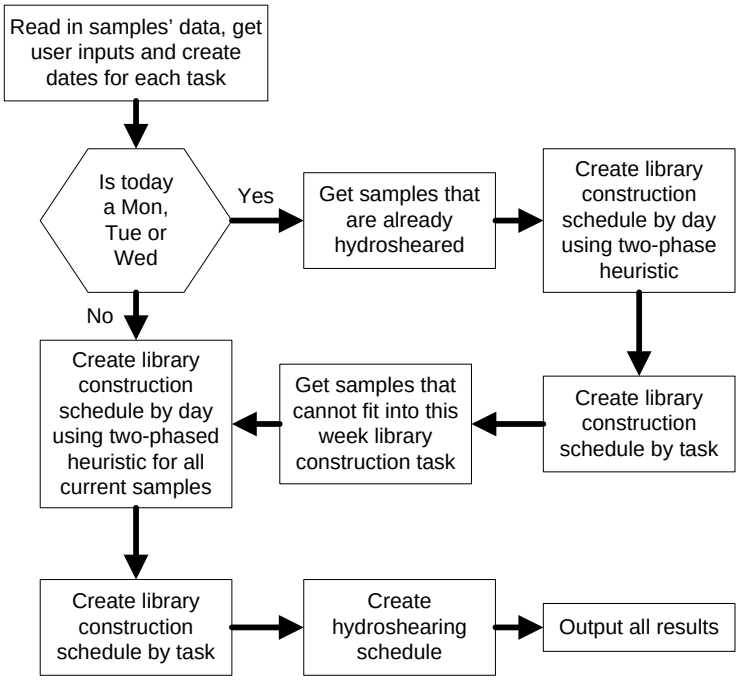

Figure 3.3 Expanded version of overall algorithm flow.

3.5.2 Two-phased heuristic

This section explains in detail the implementation of the two-phased heuristic that is mentioned in the earlier sections. The first part of this subsection will explain phase one of the heuristic and then followed by phase two. The output from phase one is a library construction schedule by day. Phase two tries to minimize the vacancy in the earlier days of this library construction schedule by day.

There are a few inputs to this heuristic. The first is a list of samples that are to be scheduled for a library construction task with their parameters like type, work request date, due dates, batch sizes and stages. The second input is the set of library construction dates that is mentioned in the overall algorithm flow. The last input is the number of library construction tasks in each week of the planning horizon, which is input by the user prior to solving the scheduling problem. The output of this heuristic is the library construction schedule by day for this set of samples. This output will be used to create a library construction schedule by task.

PHASE 1

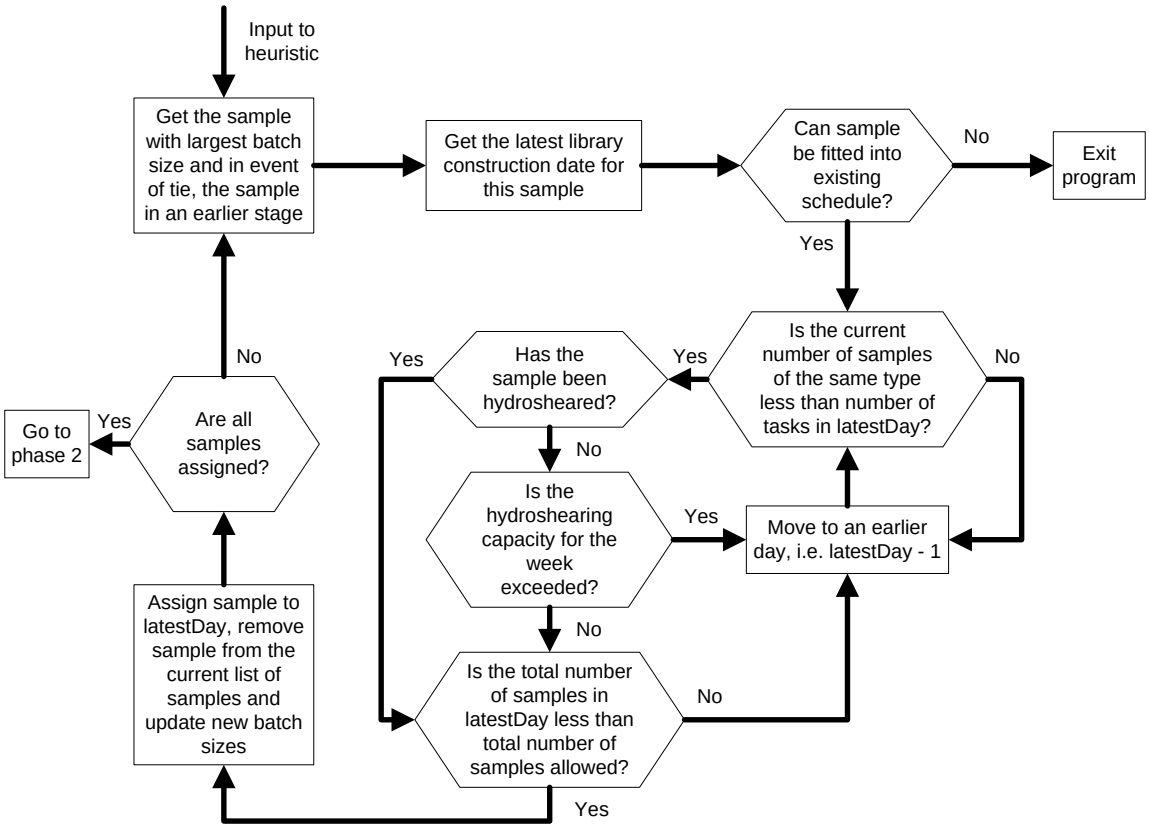

Figure 3.4 Phase 1 of the two-phased heuristic.

Get the sample with largest batch size and in event of tie, the sample in an earlier stage: Given the current list of samples, the sample that has the type with the largest batch size is first chosen for assignment. In the event of a tie, the sample in an earlier stage is chosen. The rationale for using the largest batch size criterion is because the maximum number of weeks for processing a sample is three weeks and there are typically 6 tasks in each week. Hence, it is more likely to reach the maximum number of samples of the same type that can be processed within these three weeks. By placing samples with the largest batch size, it can be quickly verified if a feasible schedule exists for this type of sample. After this criterion, the next criterion is the hydroshearing capacity constraint. Hence, in

the event of a tie in batch size, the sample in the PRE_HYDROSHEARING stage is chosen. By trying to place samples that require hydroshearing earlier on, the program will quickly determine if a feasible schedule is not possible.

Get the latest library construction date for this sample:
Given the due date of the sample, the latest library construction date is the date of the last library construction task that the sample must be assigned to before the sample is past due. Table 3.3 below shows the date of last task given the day that the due date falls on. The latest day starts from this latest library construction date.

Day that due date falls on	Date of last task
Monday	The Wednesday before that due date, i.e. previous week's Wednesday
Tuesday	The Wednesday before that due date, i.e. previous week's Wednesday
Wednesday	The Monday before that due date, i.e. the Monday of this week
Thursday	The Monday before that due date, i.e. the Monday of this week
Friday	The Wednesday before that due date, i.e. the Wednesday of this week
Saturday / Sunday	It is not expected to have due dates on Saturday and Sunday.

Table 3.3 The latest library construction task given a sample's due date. The week in this table refers to the week in which the due date falls on.

Can sample be fitted into existing schedule? & Exit program:
This step checks if the current sample can fit into the existing schedule. If it can be scheduled, the program moves to the next step. If it cannot be scheduled, the program creates an error log and exits. The error log contains the existing library construction schedule by day before the program exits. The criterion that causes the program to stop is also displayed.

There are three criteria that need to be checked. The first is to check if all days from day one to the latest day have reached the total capacity in each day. The total capacity in each day is given by the number of tasks in each day multiplied by the number of samples per task. The number of samples per task is 6 in the library construction scheduling problem. The second criterion is if all tasks in each day starting from day one to the latest day are already filled by samples of the same type as the current sample. The last criterion only applies if the current sample is in the PRE_HYDROSHEARING stage. This last criterion can be broken down into two parts. The first part checks if the total hydroshearing capacities for all weeks from the first week to the week containing the latest day have been reached. The second part checks if the hydroshearing capacities for this type of sample in all weeks from the first week to the week containing the latest day have already been reached.

The next three steps of this function are identical to those mentioned above. However, the program checks each of the three criteria in the current latest day to find a day to fit this sample rather than trying to test the possibility of adding this sample into the existing library construction schedule by day.

Is the current number of samples of the same type less than number of tasks in latestDay? :
The number of samples of the same type as the current sample cannot exceed the number of tasks in the latest day. This is because of the compatibility constraint within each library construction task. That is, no two samples in a task should be of the same type.

Has the sample been hydrosheared? & Is the hydroshearing capacity for the week exceeded:
If the sample is in the HYDROSHEARING or POST_HYDROSHEARING stage, inserting this sample into this latest day does not affect the hydroshearing capacity for this week. The program moves on to check the next criteria. If the sample is in the PRE_HYDROSHEARING stage, there are two criteria to check. The first is to test if the total number of samples in the PRE_HYDROSHEARING stage has already been reached for the current week. The second is to test if the number of samples in the PRE_HYDROSHEARING stage of the same type as the current sample has already been reached in the current week. If both criteria are met, the program moves to check the next criterion.

Is the total number of samples in latestDay less than total number of samples allowed? :
The total number of samples that can undergo library construction in the latest day is given by the number of tasks in that day multiplied by the number of samples in each task which is 6. If the number of samples already assigned to this latest day is less than this total number of samples allowed, the current sample can be added to this latest day.

Assign sample to latestDay, remove sample from the current list of samples and update new batch sizes:
If any of the above criteria cannot be fulfilled, the program moves forward in time by a day and continues searching for a day to fit the current sample. If all the above criteria are met, the current sample is assigned to this library construction day. The remaining total library construction capacity and remaining library construction capacity for this type of sample are updated. The remaining hydroshearing capacity is also updated if the sample is in PRE_HYDROSHEARING stage.

Since the program has already checked that it is possible to fit this current sample into the existing library construction schedule by day, this current sample must be able to fit into any one of the days. In some special cases, the latest day might have gone before the earliest library construction date. That is due to the interaction between this sample and other samples in the existing library construction schedule by day. These special cases are the subject of discussion in a later section.

This sample is then removed from the list of samples. The batch size of this type of sample is also updated after this assignment.

Are all samples assigned? :
The procedure described above is repeated until all the samples have been assigned a library construction day. After assigning all samples, the program moves to the next phase.

PHASE 2

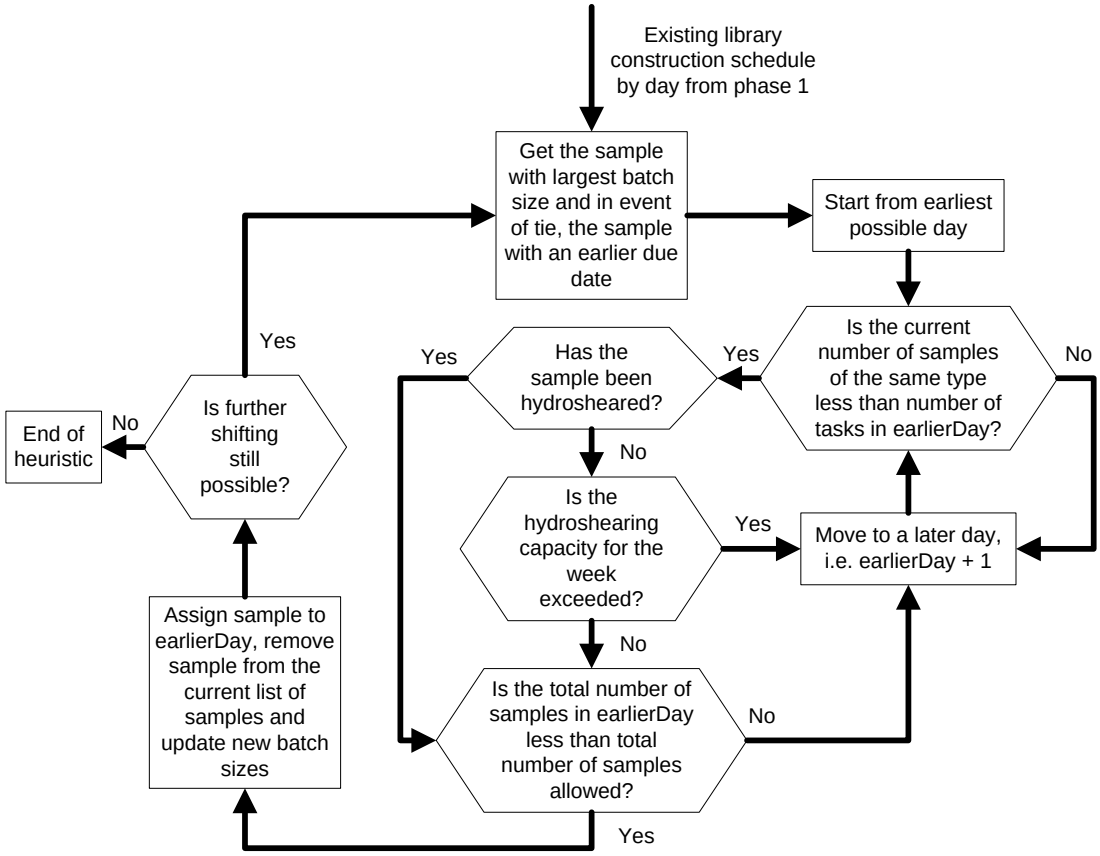

Figure 3.5 Phase 2 of the two-phased heuristic.

The input to phase 2 is the existing library construction schedule by day from phase 1. The implementation of this phase has minimal differences to the previous phase. The few differences are that firstly, the program tries to shift each sample to the earliest possible library construction date and if that fails, the next date and so on. Hence, the program starts from the earliest date and moves forward in time rather than backward in time in phase 1. Secondly, the sample with the earlier due date is chosen in the event of a tie in largest batch size. This is because preference is given to those samples with earlier due date to minimize turnaround time of the samples even though this is not a stated objective of the problem. Lastly, this shifting process is repeated until no shifting has occurred in the last iteration. This is done so as to ensure that no vacancy in an earlier task is left unfilled where possible.

Get the sample with largest batch size and in event of tie, the sample with an earlier due date:
Given the current list of samples, the sample that has the type of the largest batch size is first chosen for assignment. Choosing the largest batch size to move forward will ensure that the vacancy in the next day is always more than the current day. The rationale for selecting the sample of the type with the largest batch size is similar to the explanation given under section on simple heuristics. The only difference is that some samples might already exist in the earlier tasks after phase 1. Hence, rather than choosing 6 types of sample or all types of samples if the types are less than 6 as in the largest batch size heuristic, the number of samples that phase 2 can choose in each task is further limited by the existing samples. Although the scheduling here plans on a day by day basis, by maximizing the number of samples in each task of the day, the number of samples in each day is also maximized. Furthermore, this heuristic does not suffer from the problem associated with the largest batch size heuristic. Due dates are not violated because the existing schedule is constructed to ensure that samples are not scheduled for their library construction any later than the latest library construction date.

In the event of a tie, the sample with an earlier due date is chosen. This is not a stated objective of the optimization problem. However, it is done so as to process the sample with an earlier due date first as it has a tighter time frame and rework might occur.

Start from earliest possible day:
Given the work request date of the sample, the earliest library construction date is the date of the first library construction task that the sample can be assigned. It is the first Monday after the work request date.

The assumption here is that samples arriving in that week can complete hydroshearing by that week subject to hydroshearing capacities constraints.

Is the current number of samples of the same type less than number of tasks in earlierDay? & Has the sample been hydrosheared? & Is the hydroshearing capacity for the week exceeded? & Is the total number of samples in earlierDay less than total number of samples allowed? :
These criteria are identical to those mentioned in phase 1 except that the program is checking those criteria on an earlier day rather than a later day.

If the sample cannot be assigned to the current earlier day, the program moves forward by a day and repeats the checking until the earlier day reaches the originally assigned library construction date for the current sample. When this happens, the sample is removed from the list of samples. This means that the sample is already assigned to the earliest possible day given the existing library construction schedule by day.

Assign sample to earlierDay, remove sample from the current list of samples and update new batch sizes:
After assigning the current sample to the earlier day, the remaining library construction capacity for the earlier day has to be updated. If the current sample is in

PRE_HYDROSHEARING stage, the remaining hydroshearing capacity for the earlier day is also updated. In addition, the remaining library construction and hydroshearing capacities in the previously assigned library construction date have to be restored after removing this sample from the previous assigned date.

The current sample is removed from the list of samples and the batch size of this type of sample is updated.

Is further shifting still possible? :
This procedure is repeated until all the samples in the list have been considered for shifting.

The whole procedure is repeated for the original list of samples until no shifting can occur for original list of samples in the existing library construction schedule by day. This ensures that there is no vacancy in the earlier tasks.

3.5.3 Create Library Construction Schedule by Task Given the Library Construction Schedule by Day

The purpose of this function is to generate the library construction schedule by task based on the output from a library construction schedule by day. This function is required because the two-phased heuristic can only output a day schedule. Hence, it is still necessary to assign samples in each day to tasks.

The input into this function is the library construction schedule by day that is created by the two-phased heuristic. The output from this function is the library construction schedule by task which will be used as the input for the hydroshearing scheduling.

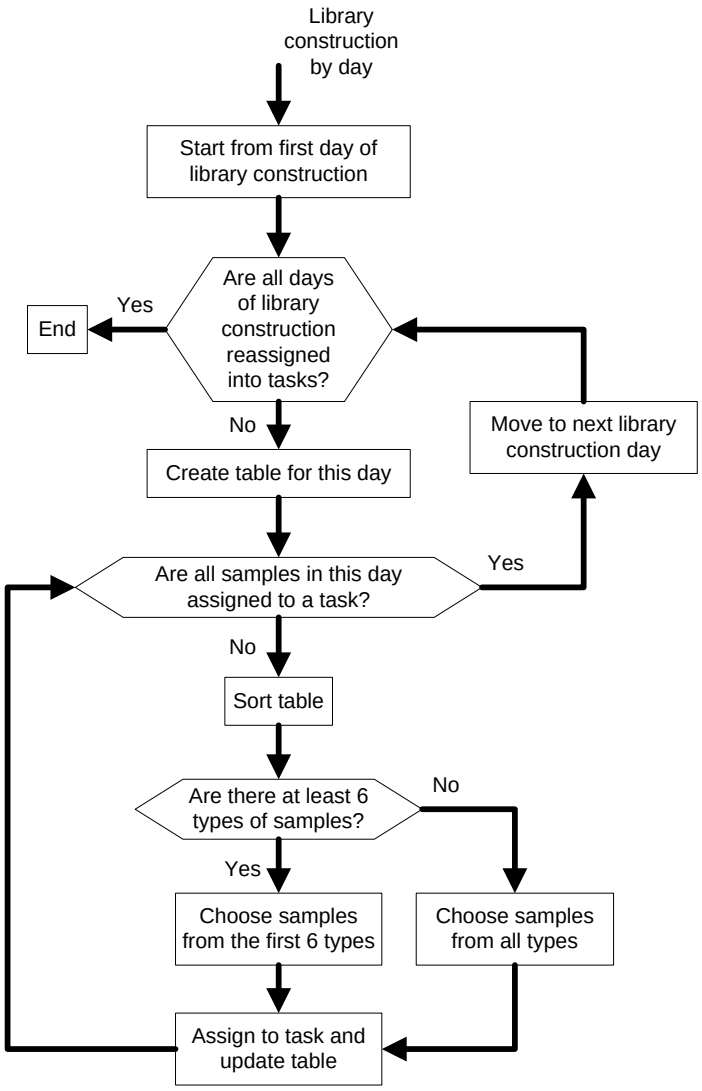

Figure 3.6 Create library construction by task.

Create table for this day:
A table consisting of the different types, the number of samples of each type (batch size) and the earliest due date within each type is generated. This is required for the selection of samples to go into each task.

Sort table:
The generated table is sorted by batch sizes in descending order and in the event of a tie, by due dates in ascending order. The reason for doing so is because the problem of scheduling samples in each library construction day into tasks can be considered as a subproblem that is similar to the original problem without considering hydroshearing. This method of choosing samples to be grouped into tasks will result in less vacancy in the first task as compared to the second task, less vacancy in the second task as compared to the third task and so on.

Since all tasks in this day are carried out on the same day, there is no difference in assigning them to the first task or last task in the day. The only issue is to assign all samples in this day to a minimum number of tasks subject to compatibility constraints in each task.

Choose samples from the first 6 types:
The maximum capacity of each library construction task is 6. Hence, if there are at least 6 types of samples in the table, the first 6 types of samples in the sorted list are chosen to be grouped into a task. After identifying these 6 samples, the actual samples of these types are chosen from the list of samples in this day.

Choose samples from all types:
If there are less than 6 types of samples, only the current number of types can be chosen. The actual samples of these types are then chosen from the list of samples in this day.

Assign to task and update table:
The chosen samples from the previous step are grouped into a new task. The batch sizes and the earliest due date of the chosen types are updated in the table after the assignment.

3.5.4 Create Hydroshearing Schedule Given the Library Construction Schedule by Task

This function goes through the library construction schedule by task starting from the next Monday and picks out samples that are at the PRE_HYDROSHEARING stage for hydroshearing scheduling. The hydroshearing schedule groups the samples that require hydroshearing according to the weeks when the samples will be hydrosheared.

The input to this function is the library construction by task. The output from this function is the hydroshearing schedule for samples that are in the PRE_HYDROSHEARING stage.

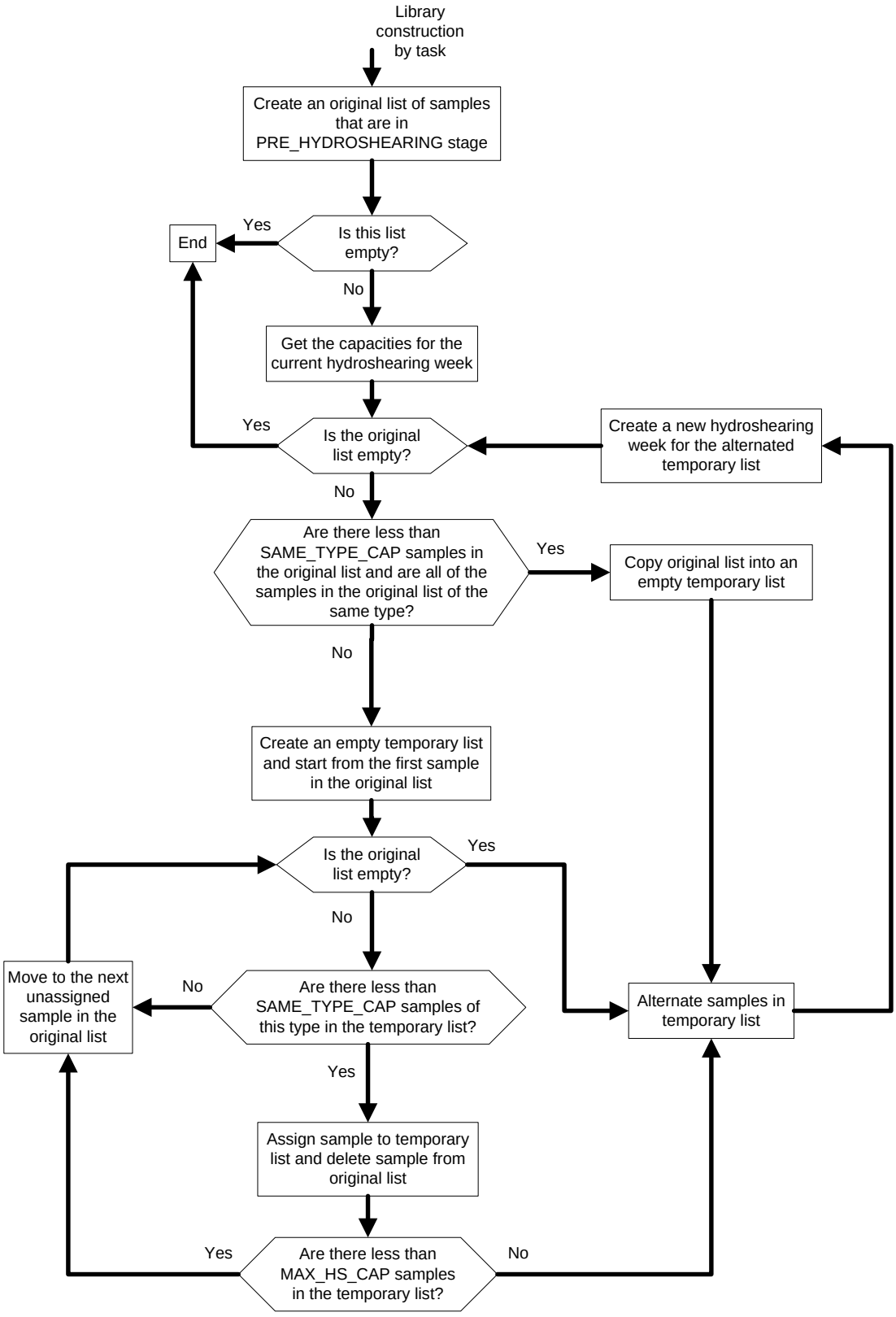

Figure 3.7 Hydroshearing scheduling.

Create an original list of samples that are in PRE_HYDROSHEARING stage:
Starting from the first library construction task, a list of the samples in this task that are in the PRE_HYDROSHEARING stage is created. Samples in the second task that are in the PRE_HYDROSHEARING stage are concatenated to the end of the list. The process is repeated for all tasks in the library construction schedule by task. If this list is empty, there is no need to perform hydroshearing scheduling since there are no more samples at PRE_HYDROSHEARING stage.

Get the capacities for the current hydroshearing week:
There are two kinds of capacity to be determined. The first is the SAME_TYPE_CAP which is the number of samples of the same type that can be accommodated in the current week. The second is the MAX_HS_CAP which the maximum number of samples that can be hydrosheared in the current week. In any week, MAX_HS_CAP is typically twice the SAME_TYPE_CAP.

In the first week, MAX_HS_CAP is given by the number of hydroshearing slots remaining in the current week, as entered previously by the user in an earlier step of the overall algorithm flow. For all other weeks, MAX_HS_CAP is given by the number of hydroshearing slots for each subsequent week.

As an example, if the MAX_HS_CAP for the current hydroshearing week is 20, SAME_TYPE_CAP will be 10. The current hydroshearing week can only hold up to 20 samples and among these 20 samples, at most 10 of them can be of the same type.

The current hydroshearing week starts at week one and moves on to the next week once the hydroshearing constraints are met and all samples in the original list have not been assigned a hydroshearing week.

Are there less than SAME_TYPE_CAP samples in the original list and are all of the samples in the original list of the same type? & Copy original list into an empty temporary list:
Suppose that the original list contains less than SAME_TYPE_CAP samples of the same type. If the program has entered into the next step of the flowchart in figure 3.7, it would have entered into an infinite loop because the exit criterion is never satisfied. Hence, there is a need to check this condition and create a hydroshearing week schedule to contain all these samples.

Create an empty temporary list and start from the first sample in the original list:
A temporary list is needed to store the samples that are assigned to the current hydroshearing week. This temporary list is simply a list of samples extracted in ascending order of the tasks in the library construction by task subject to hydroshearing capacities constraints. This temporary list will show the samples scheduled for hydroshearing in the current hydroshearing week but it is not the actual sequence that hydroshearing will perform because of the possibility of having consecutive samples of the same type. This explains the need for a function to alternate the samples, which will be explained shortly.

The program then proceeds to scheduling each sample in the original list, starting from the first sample in that list.

Are there less than SAME_TYPE_CAP samples of this type in the temporary list? :
The current hydroshearing week should have less than SAME_TYPE_CAP samples of the same type before the current sample can be assigned to this week.

Assign sample to temporary list and delete sample from original list:
After assigning the sample to the temporary list, both the capacities for this type of sample and the total number of samples in the current hydroshearing week are reduced by one. Next, this assigned sample is removed from the original list.

Are there less than MAX_HS_CAP samples in the temporary list? :
The total number of samples in the current hydroshearing week is compared to MAX_HS_CAP. If the maximum capacity is not reached, the program continues to add more samples from the original list into the current hydroshearing week. If the maximum capacity is reached, the program moves on to alternating the samples within the temporary list.

Alternate samples in temporary list:
This step aims to produce a hydroshearing week schedule that does not contain samples of the same type in any two consecutive slots of hydroshearing week schedule. The idea is to take out the identical type samples of the longest batch size and insert the rest of the samples in between those identical type samples. If there are insufficient samples of a different type, empty slots are used. Turning this idea into a sequential program requires more effort. The temporary list might contain such consecutive slots of identical type samples at the top, middle or end of the list. Hence, the general idea of this function is to move the identical type samples to the end of the list before alternating the sample. After moving the identical type samples to the end, there exists a single type of sample that will accumulate at the end.

Starting from the top of the list, the function tries to locate any consecutive samples that are of the same type. If there is such a scenario, a different type of sample is identified from further down the list and inserted in between the consecutive identical type samples. If there are no available samples that are of a different type, the process continues working down the list. Once the end of the list has been reached, all samples at the end of the list would be of the same sample type.

The second step of this function would count the number of identical type samples at the end of the list. The sum of the number of samples that is of a different type and the empty slots required to alternate these identical type samples is given number of identical type samples less one. Hence, the function proceeds back up the list and select those samples of different types. If the function encounters samples of the same type as those identical type samples as it goes back up the list, it would require one more sample of the same type. This continues until the top of the list and results in two lists, one containing samples of the identical type and another containing samples of a different type. The final

hydroshearing week schedule with alternating samples is then created by taking one each from the list and if there are no more samples in the latter list, an empty slot is used.

Create a new hydroshearing week for the alternating temporary list:
The actual hydroshearing week is simply the alternating temporary list. The program moves on to the next hydroshearing week.

The whole procedure is continued until there are no more samples left in the original list.

3.6 Conclusion

This chapter starts with some of the simpler heuristic that could possibly be used for the problem. However, these simpler heuristics may not give a feasible or optimal solution. This motivated the design of a new heuristic. The description of this heuristic is the focus of this chapter. After a brief introduction of the heuristic, the chapter dives into the details of implementation.. The actual implementation is carried out using MATLAB. The program is now in use at Broad Institute. Although the user is pleased with the program, the performance of the heuristic must be investigated.

Although the heuristic is described in detail, no theoretical argument has been presented to support the efficiency and performance of the heuristic. Therefore, a comparison is required with other methods that solve the same problem using historical data. In the next chapter, the integer programming method will be introduced and in the chapter after that, the performance of the heuristic will be compared against that of the integer programming method.

4 Integer Programming Formulations

4.1 Introduction

Integer programming (IP) or mixed integer programming (MIP) formulation is one of the conventional ways to solve scheduling problems, and the mathematical model is usually solved by branch and bound (B&B). However, the efficiency of B&B depends on the tightness of the bounds. The computational effort needed is usually much higher than for a heuristic, but in theory, these methods are designed to find an optimal solution.

Furthermore, as the scale of the problem increases, the computational time required might be such as to make the method infeasible. Nevertheless, it is still a desired model that can be used to evaluate the effectiveness of a heuristic or other techniques. By comparing the solution obtained by heuristic or other approaches to a lower bound, such as generated by Langrangian relaxation for the IP/MIP, researchers [Caricato2007, Ahmadi 1992] are able to judge the efficiency and the effectiveness of a certain approach.

Therefore, our motivation of formulating IP models to solve the hydroshearing and library construction scheduling problem is to compare the performance and computational results of IP formulation and the proposed two-phase heuristic as discussed in Chapter 3.

For comparison purposes, we generate four different test environments which will be introduced in the following section. We then provide the motivation for the different test environments, after which we will present four different IP formulations that correspond to these test environments.

4.2 Test Environments

A test environment consists of a scheduling policy under which the scheduling methods, such as the proposed two-phase heuristic and IP formulation will be applied, and a specific type of scheduling problem to solve.

4.2.1 Static and Dynamic Scheduling Policy

The actual scheduling problem is dynamic in that we have randomly arriving samples and we must periodically re-schedule the samples as time moves forward and we have new input data for the scheduling problem. However for test purposes, we will compare s=our scheduling methods for both a static and dynamic version of the scheduling problem. We describe these two scheduling policies next.

Dynamic Scheduling Policy

In the dynamic scheduling policy, the history is reenacted using the data sets. The heuristic and IP model are applied to the scheduling problem on a rolling horizon basis, where we assume that samples' arrivals cannot be predicted. This means when a

scheduling decision has to be made, the scheduling problem only considers the set of samples on hand, produces a future schedule myopically, and processes the first few jobs scheduled. When it comes to the next decision point, all future work that is previously planned will be re-scheduled. This is the actual scheduling policy we proposed to Broad Institute, which incorporates the heuristic proposed in Chapter 3.

Static Scheduling Policy
In contrast to the dynamic scheduling policy, under the static scheduling policy, we assume that the arrival date of every sample for some finite time period is known, and that we will just develop a single schedule that accounts for all the samples in the data set. Therefore, the fact that samples arrive randomly is neglected.

Furthermore, under the static scheduling policy, we assume that the number of hydroshearing slots is the same in each day of the week. For instance, if the current capacity of hydroshearing is 20 samples per week regardless of sample type, we assume that we can accommodate up to most 4 samples each day regardless of type, among which less than 2 can be of the same type due to hydroshearing compatibility constraints.

Another difference is that under the static scheduling policy, a sample must be scheduled between its arrival date and its due date.

The motivation of applying the above mentioned two scheduling policies is by comparing the results given under the dynamic and static scheduling policies respectively, we are able to test if our dynamic scheduling policy can provide reasonable and good schedules consistently.

4.2.2 Relaxed and Constrained Problems

The second element of a test environment is a specific defined scheduling problem. Our original hydroshearing and library construction scheduling problem can be modeled as a two stage flow shop scheduling problem, and hence can be decoupled into two subproblems: the hydroshearing scheduling problem, and the library construction scheduling problem. In order to isolate the subproblem of library construction scheduling, we can relax the hydroshearing capacity constraints , and consequently create the relaxed problem, while the original problem is referred to as the constrained problem.

Relaxed Problem
In the relaxed problem, the hydroshearing capacity is set to infinite. However, this does not mean that samples do not require hydroshearing. Rather, samples can always be hydrosheared in the week that they arrive, and complete this procedure on the first Friday on or after its arrival date. For example, if it arrives on a Thursday, it will complete hydroshearing on the Friday in the same week, and if it arrives on Friday, it will complete hydroshearing within one day.

The motivation for solving the relaxed problem is that the two-phased heuristic is designed to tackle the library construction scheduling problem. By comparing the results

from the heuristic and IP model, we hope to get insight into the performance of the heuristic.

Constrained Problem
The constrained problem is the original scheduling problem, which includes capacity and compatibility constraints in both hydroshearing and library construction procedures. In this problem, samples can complete hydroshearing on the first Friday on or after the arrival date only if they have been scheduled for hydroshearing before the end of the working week subject to hydroshearing capacity and compatibility constraints. For example, if it arrives on a Thursday, it can complete hydroshearing on the Friday in the same week if there are available hydroshearing slots left on Thursday or Friday.

By comparing the results of the relaxed problem against that of the constrained problem, we can understand the effects of hydroshearing capacity constraints on the solutions. Furthermore, testing under the constrained environment using the historical data sets also enables us to compare the schedule created by the heuristic and the historical schedule that was manually created at Broad Institute.

4.2.3 Summary of Test Environments

The above sections define the two key elements that compose a test environment. The combinations of one chosen from dynamic and static scheduling policy, with one from constrained and relaxed problem compose the four test environments that we will compare the computational results of IP and heuristic. The following figure 4.1 summarizes the four test environments:

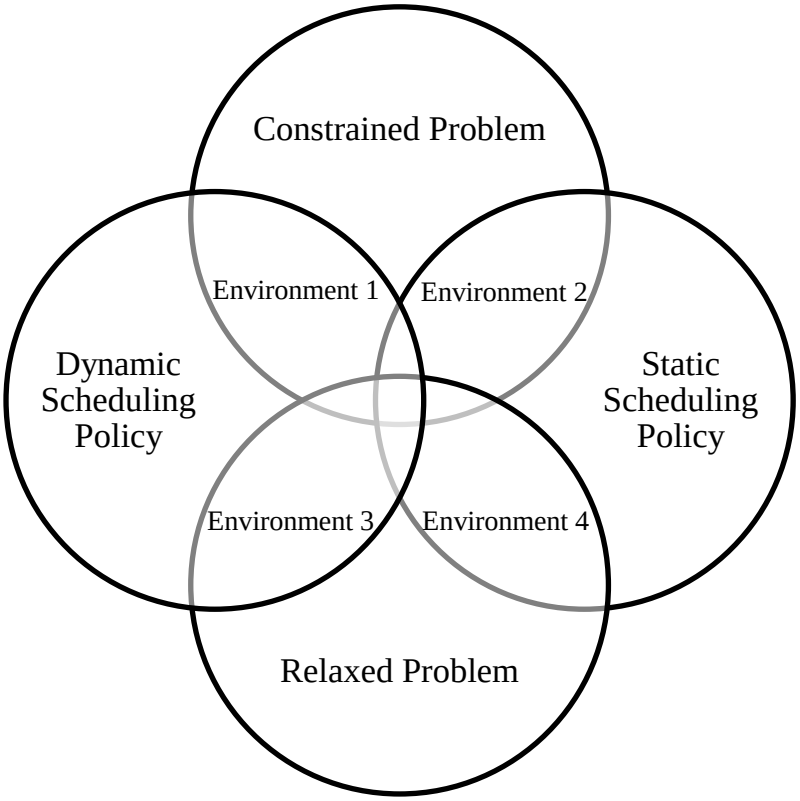

Figure 4.1 Test environments.

As shown in the above figure 4.1, test environment 1 solves the constrained problem under the dynamic scheduling policy, while environment 2 solves the same problem but under the static scheduling policy. Testing in the four different environments helps us to have a better understanding of the nature of the analyzed scheduling problem as well as evaluate the effectiveness of the proposed heuristic. To this end, we formulate four different IP models to perform the corresponding test environment, which is the focus of the following sections in this chapter.

4.3 Notations

This section lists the variables and notations required to understand the IP formulations presented in the rest sections of this chapter.

Parameters

Consider a data set that contains samples from N different types. Let $\{1, \cdots, N\}$ denote the set of types. Let m_i denote the number of samples in type i, for $i = 1, \cdots, N$. In other words, m_i is the batch size of type i. Let M be the largest batch size among all the sample types, or $M = \max\{m_1, m_2, \cdots, m_N\}$. The reason of setting M to the largest batch size is for the sake of simplicity in programming the mathematical model.

Let $A_{i,j}$ denote the arrival date of sample j in type i, for $i = 1, \cdots, N, j = 1, \cdots, M$. In each type i, let $A_{i,j} = 0, \text{for } m_i < j \leq M$. This parameter will only be used in IP models for the test environments which adopt the static scheduling policy.

Let $D_{i,j}$ denote the due date of sample j in type i, for $i = 1, \cdots, N, j = 1, \cdots, M$. In each type i, let $D_{i,j} = 0, \text{for } m_i < j \leq M$. In our IP models, we do not allow delay of any sample. Again, this is for the sake of comparison with heuristic, because the first phase of heuristic guarantees that each sample is processed before its due date.

Let $S_{i,j}$ denote the stage of sample j in type i, for $i = 1, \cdots, N, j = 1, \cdots, M$. $S_{i,j} = 1$, if the sample has just arrived and has not yet been hydrosheared. $S_{i,j} = 0$, if the sample has returned from hydroshearing. This parameter will only be used in the IP models for the test environments that tackle the constrained problem.

Let set $\{1, \cdots, T\}$ denote the set of weekdays (any day of the week other than Saturday or Sunday), and set $\{1, \cdots, W\}$ denote the set of weeks in the scheduling time horizon. For example, day 1, 2, …, 5 is in week 1, while day 6, 7, …, 10 is in week 2. Note that in each IP model, we schedule for a finite time horizon. If the start day of the scheduling time horizon is not a Monday, we will round it to the previous Monday in the same week, but we do not schedule any samples on the days between that Monday and the actual start day. It is the same case when the end day is not a Friday.

Let set {TASK_DAYS} denote the set of library construction task days, in other words, it contains the index of days in $\{1, \cdots, T\}$ that is either a Monday or a Wednesday. As specified in the problem definition, samples can only be scheduled for library construction on these days.

As introduced in Section 3, let SAME_TYPE_CAP be the number of samples of the same type that can be accommodated in a hydroshearing week (from Monday to Friday in the same week), and let MAX_HS_CAP be the maximum number of samples that can be hydrosheared in a hydroshearing week. In any week, MAX_HS_CAP is twice the SAME_TYPE_CAP.

For library construction, let NUM_TASKS be the number of tasks that can be accommodated in library construction day (Monday or Wednesday), and let NUM_SAMPLES_PER_TASK be the maximum number of samples in each library construction task. Furthermore, the compatibility constraint limits that no more than one sample of a type can be allocated in the same library construction task. Therefore, we can easily derive that for each library construction day, the maximum number of samples in the same type is NUM_TASKS.

Decision Variables

We define the following binary variables for hydroshearing and library construction:

For $i = 1, \cdots, N, j = 1, \cdots, M, t = 1, \cdots, T$:

$$x_{i,j,t} = \begin{cases} 1, \text{if jth sample in type i is allocated on day t for hydroshearing} \\ 0, \text{otherwise} \end{cases}$$

$$y_{i,j,t} = \begin{cases} 1, \text{if jth sample in type i is allocated on day t for library construction} \\ 0, \text{otherwise} \end{cases}$$

$$\text{For } i = 1, \cdots, N, j = 1, \cdots, M, w = 1, \cdots, W:$$
$$p_{i,j,w} = \begin{cases} 1, \text{ if jth sample in type i is allocated on week w for hydroshearing} \\ 0, \text{otherwise} \end{cases}$$

$$q_{i,j,w} = \begin{cases} 1, \text{if jth sample in type i is allocated on week w for library construction} \\ 0, \text{otherwise} \end{cases}$$

Furthermore, we have:

$s_{i,t}$	the number of samples in type i that are allocated in day t for library construction
r_t	minimum number of library construction tasks to pack all samples in day t ignoring compatibility constraints, which is the total number of samples in that day divided by NUM_SAMPLES_PER_TASK
π_t	the number of library construction tasks allocated on day t
$C_{i,j}$	the completion time of sample j in type i. It starts from the arrival day until the day the samples is scheduled for library construction under static scheduling policy, while under dynamic scheduling policy, it is the days between the current day and the scheduled library construction day.
C_{max}	the maximum completion time among all samples

4.4 Proposed IP Formulations

In this section we present four different IP formulations solving the scheduling problem in the four test environments respectively.

Although the assumptions and the constraints may vary from model to model, the objectives are the same in the four formulations. In order to have a better understanding of the performance and computational results of IP, the objectives that we consider in IP formulation require finding feasible hydroshearing and library construction schedules such that all of the following cost functions are minimized:

π_t	the number of tasks allocated on day t
$\sum C_{i,j}$	the total completion time of samples
C_{max}	the maximum completion time among all samples

The *completion time* of a sample is defined as the number of weekdays between the work request date and the day when library construction is completed. Among these multiple objectives, we assign different weights to them respectively to show a difference in priorities of these objectives. As discussed in Section 2.3.2, the objective of minimizing the number of tasks in library construction is our first priority, and our second objective is to minimize the total completion time of samples, while minimizing the maximum

completion time is a third priority in our IP models. The weights assigned to the second and the third objectives are denoted by μ and ω respectively.

Furthermore, having such multiple objectives rather than merely minimizing π_t provides us with more comparison criteria for assessing the merits and demerits of both IP and heuristic.

4.4.1 IP Formulation in Environment 1

As defined in Section 4.2, we adopt a dynamic scheduling policy based on rolling time horizon. Therefore, at each decision point, we schedule (or re-schedule) for a finite time horizon, which starts from the current day till the latest due date among all samples.

The first IP model we present is for solving the constrained problem under the dynamic scheduling policy.

IP Model 1

$$\min\ \pi_t + \mu \cdot \sum_{i,j} C_{i,j} + \ \omega \cdot C_{max}$$

Subject to

- Hydroshearing Constraints

$$\sum_{t=1}^{D_{i,j}} x_{i,j,t} = 1\ , \forall\, i, j,\ \text{if}\ S_{i,j} = 1, D_{i,j} > 0 \qquad (1)$$

$$\sum_{t=1}^{T} x_{i,j,t} = 0\ , \forall\, i, j, \text{if}\ S_{i,j} = 0,\ D_{i,j} = 0 \qquad (2)$$

$$\sum_{i=1}^{N} \sum_{j=1}^{M} x_{i,j,t} \leq \text{MAX\_HS\_CAP}, \forall\, t \qquad (3)$$

$$\sum_{j=1}^{M} x_{i,j,t} \leq\ \text{SAME\_TYPE\_CAP}, \forall\, i, t \qquad (4)$$

$$p_{i,j,w} = \sum_{t=5\cdot(w-1)+1}^{5\cdot w} x_{i,j,t}\ , \forall\, i, j, w \qquad (5)$$

Constraints (1) and (2) guarantee that a sample that is in the PRE_HYDROSHEARING stage is allocated for hydroshearing before its due date.

Constraints (3) and (4) indicate the number of samples that hydroshearing can accommodate each day regardless of types, as well as the maximum number of samples in the same type that can be allocated in hydroshearing in a day so that samples from different types can be alternated in that day.

Constraint (5) calculates the week each sample is allocated for hydroshearing.

- Library Construction Constraints

$$\sum_{t=1}^{D_{i,j}} y_{i,j,t} = 1 \ , \forall\, i, j, \text{if } D_{i,j} > 0 \tag{6}$$

$$\sum_{t=1}^{T} y_{i,j,t} = 0 \ , \forall\, i, j, \text{if } D_{i,j} = 0 \tag{7}$$

$$\sum_{i=1}^{N} \sum_{j=1}^{M} y_{i,j,t} \leq \text{NUM\_SAMPLES\_PER\_TASK}, \forall\, t \in \{\text{TASK\_DAYS}\} \tag{8}$$

$$\sum_{j=1}^{M} y_{i,j,t} \leq \text{NUM\_TASK}, \forall\, i, t \in \{\text{TASK\_DAYS}\} \tag{9}$$

$$q_{i,j,w} = \sum_{t=5\cdot(w-1)+1}^{5\cdot w} y_{i,j,t} \ , \forall\, i, j, w \tag{10}$$

Constraints (6) and (7) mean that each sample undergoes library construction before its due date.

Constraints (8) and (9) indicate the capacity of library construction on each task day and the compatibility of types for samples allocated in the same task respectively.

Constraint (10) calculates the week each sample is allocated for library construction.

- Workflow Constraints

$$\sum_{w_1=1}^{w} p_{i,j,w_1} \geq \sum_{w_2=1}^{w+1} q_{i,j,w_2} \ , \forall\, i, j, w = 1, \cdots, W-1 \tag{11}$$

$$p_{i,j,W} = 0, \forall\, i, j \tag{12}$$

$$q_{i,j,1} = 0, \forall\, i, j \tag{13}$$

This set of constraints ensures that the week a sample is hydrosheared is prior to the week of library construction. In other words, each sample must complete hydroshearing before it can be sent to library construction. For instance, if a sample is hydrosheared in week 2, it can only be allocated for library construction after week 2, and of course, before it is due.

- Number of tasks in each day

$$s_t = \sum_{j=1}^{M} y_{i,j,t} \ , \forall\, t \in \{\text{TASK\_DAYS}\} \tag{14}$$

$$r_t \geq \sum_{i=1}^{N} \sum_{j=1}^{M} y_{i,j,t} \,/\text{NUM\_SAMPLES\_PER\_TASK} \ , \forall\, t \in \{\text{TASK\_DAYS}\} \tag{15}$$

$$\pi_t \geq s_t \ , \forall\, t \in \{\text{TASK\_DAYS}\} \tag{16}$$

$$\pi_t \geq r_t \ , \forall\, t \in \{\text{TASK\_DAYS}\} \tag{17}$$

This set of constraints calculates the number of tasks that are allocated on each task day. This is derived from Theorem 1.

- Completion Time

$$C_{i,j} = \sum_{t=1}^{T} t \cdot y_{i,j,t} \ , \forall\, i,j \tag{18}$$

This constraint calculates the completion time of each sample, which is the number of days between the current day and the day when the sample is scheduled for library construction.

- The Maximum of Completion Time

$$C_{max} \geq C_{i,j} \ , \forall\, i,j \tag{19}$$

Constraint (19) calculates the maximum of completion time among all samples.

4.4.2 IP Formulation in Environment 2

In contrast to the dynamic scheduling policy, we assume the exact arrival date of every sample is known in the static environment. Consequently, we can schedule for all the samples in the data set once. In the following IP model, most of the constraints are the same as in IP Model 1. The only difference is that in the environment 2, a sample can only be scheduled after its arrival date.

IP Model 2

$$\min\ \pi_t + \mu \cdot \sum_{i,j} C_{i,j} + \ \omega \cdot C_{max}$$

Subject to

(3) (4) (5) (8) (9) … (17) (19)

$$\sum_{t=A_{i,j}}^{D_{i,j}} x_{i,j,t} = 1 \ , \forall\, i,j, \text{if } S_{i,j} = 1, A_{i,j} > 0, D_{i,j} > 0 \tag{20}$$

$$\sum_{t=1}^{T} x_{i,j,t} = 0 \ , \forall\, i,j, \text{if } S_{i,,j} = 0, A_{i,j} = 0, \ D_{i,j} = 0 \tag{21}$$

$$\sum_{t=A_{i,j}}^{D_{i,j}} y_{i,j,t} = 1 \ , \forall\, i,j, \text{if } A_{i,j} > 0, D_{i,j} > 0 \tag{22}$$

$$\sum_{t=1}^{T} y_{i,j,t} = 0 \ , \forall\, i,j, \text{if } A_{i,j} = 0, D_{i,j} = 0 \tag{23}$$

$$C_{i,j} = \sum_{t=1}^{T} (t - A_{i,j}) \cdot y_{i,j,t} \ , \forall\, i,j \tag{24}$$

Constraints (20) to (23) simply guarantee that each sample is allocated for hydroshearing and library construction respectively after it arrives and before its due date.

Constraint (24) calculates the completion time under the static scheduling policy.

4.4.3 IP Formulation in Environment 3

The IP model of the relaxed problem under the dynamic scheduling policy is similar to IP Model 1, but without any constraints on the hydroshearing and work flow.

IP Model 3

$$\min\ \pi_t + \mu \cdot \sum_{i,j} C_{i,j} + \omega \cdot C_{max}$$

Subject to

(6) (7) … (10) (14) (15) … (19)

4.4.4 IP Formulation in Environment 4

The IP model of the relaxed problem under the static scheduling policy is similar to IP Model 2, but without any constraints on the hydroshearing and work flow.

IP Model 4

$$\min\ \pi_t + \mu \cdot \sum_{i,j} C_{i,j} + \omega \cdot C_{max}$$

Subject to

(8) (9) (10) (14) (15) … (17) (19) (20) … (24)

4.5 Conclusion

In this chapter, we first introduce the test environments that the 4 IP formulations have been implemented for. Following that, the IP formulations of the hydroshearing and library construction of scheduling problem have been presented. In the next chapter, we conduct the experiments to compare the performance of the heuristic against these IP formulations.

5 Performance Tests and Sensitivity Analysis

5.1 Introduction

This chapter covers the comparisons, performance tests and sensitivity analysis of the scheduling problem. Two different solution methods to the problem, namely, heuristic and IP (IP) have been proposed in the previous chapters. The performance of these methods for solving the scheduling problem should be investigated and compared. Using historical data is ideal for this investigation.

Another aim of the tests is to examine the scheduling results in a dynamic and static environment. Our implementation of dynamic scheduling is myopic by planning only for samples that are currently in the inventory and by assuming that future samples' arrivals are unknown. For static scheduling we assume that the arrival dates of all samples are known in advance.

Another objective is to investigate the subproblem of library construction scheduling. In order to do this, two more environments are created, namely, relaxed and constrained. In the relaxed environment, the hydroshearing capacity is unlimited. Hence, the hydroshearing capacity constraints will not affect the second subproblem of library construction scheduling. In the constrained environment, the hydroshearing capacity constraints come into play and affect the library construction scheduling.

Lastly, another aim of the tests is to perform a sensitivity analysis of the schedule to changes in parameters and variables of the problem. Parameters of the problem include the number of library construction tasks per week, the number of samples in each task, and the hydroshearing capacity for the week. The variables of the problem are the characteristics of the samples. These include types, priorities and due dates.

The first comparison criterion is the feasibility of the problem. If the problem is feasible, results will include the total number of tasks required, the runtime of the two methods, the sum of completion time of all the samples and the maximum completion time of the samples. The IP model is coded and solved using ILOG OPL Studio 6.1.1. The heuristic is coded and solved using MATLAB.

This chapter first introduces the data sets on which the tests are carried out. Next, changes to the implementation of heuristic under the relaxed and static environments are provided. After that, each section consists of a test and starts by giving the motivation to conduct the comparison/investigation. A description of the test follows. The results of the tests are presented. Finally, observations and discussions summarize the section.

5.2 Data Sets

We use data obtained from the Broad Institute. There are a total of 326 samples in the data. The time horizon spans from Feb 26, 2008 to Jun 5, 2009, a total of 15 months. Samples in the data have an arrival date and a due date. A *time interval* of a sample is the time frame between the arrival date and the due date. Time intervals of samples in the data overlap in time.

Due to the limited number of variables that the IP can solve on the student version of ILOG OPL Studio, the data set has been truncated into two data sets. These two data sets are named as data set I and II. A third data set, data set III, has been created to have different characteristics from these two data sets. The reason for this third data set will be explained shortly.

In these data sets, the samples cannot possibly be completed within 3 weeks of their arrivals. For example, there are 29 samples of the same type in data set II. Hence, in this chapter, we use more priorities than high/standard as defined in chapter 2. Priorities refer to the number of weeks that samples must complete both hydroshearing and library construction and can vary up to 6 weeks in our tests here.

This section describes the characteristics of these data sets.

5.2.1 Data Set I

Data set I is from the historical data provided by Broad Institute. This data set consists of 121 samples. There are 38 different types of samples. The type of sample with the longest batch size is type 1. There are 17 samples of type 1. In this data set I, samples of the type with larger batch sizes typically arrive on the same day.

The problem is infeasible if samples are of standard priority and have 3 weeks for completing hydroshearing and library construction. The infeasibility is caused by samples of type 2. 15 samples of type 2 arrive on Thursday Sep 25, 2008. The first week of arrival is reserved for hydroshearing. Hence, there will only be 12 tasks left in the remaining two weeks. This is insufficient to process the 15 samples of type 2. Therefore, in this data set, we assume that hydroshearing and library construction must be completed within 4 weeks for all samples. The reason why all samples are given 4 weeks rather than only samples of type 2 is for fair comparison and standardization. The sensitivity of the schedule to the number of weeks allowed for hydroshearing and library construction will be examined in a later section when priorities and due dates of some of the samples are varied.

The time horizon spans from Sep 22, 2008 to Feb 2, 2009. Samples arrive randomly over the time horizon. A decision point is defined as each day when there are new samples' arrivals. Dynamic scheduling occurs at each decision point. The data set I has a total of 16 decision points.

The theoretical minimum number of tasks required for this set of samples is 21 if the due dates are ignored.

5.2.2 Data Set II

Data set II is also created from the historical data provided by Broad Institute. It has a total of 99 samples. There are 11 types of samples. The type of sample with longest batch size is type 2. There are 29 samples of type 2. In this data set, samples of the type with larger batch sizes typically arrive on the same day. Due to the large number of samples of type 2, all samples are given 6 weeks to complete hydroshearing and library construction. The time horizon spans from May 12, 2008 to Jul 31, 2008. There are a total of 17 decision points.

The theoretical minimum number of tasks required for this set of samples is 29 if the due dates are ignored.

5.2.3 Data Set III

Data set III is created randomly to have different characteristics from the previous two data sets. The previous two data sets have samples of type with larger batch sizes arriving on the same day. Each sample of the same type will thus require a task and the number of tasks required is strongly influenced by these samples. Hence, data set III consists of samples that are of different types. There are 96 samples in this data set III and each of them is of a different type. To keep the problem more manageable, six decision points are generated using a uniform distribution between Jan 4, 2010 to Jan 29, 2010. These 96 samples are to arrive on any of these six decision points. All samples must complete hydroshearing and library construction within 3 weeks.

The theoretical minimum number of tasks required for this set of samples is 16 if the due dates are ignored.

5.3 Changes to the Implementation of the Heuristic for the Static and/or Relaxed Environments

The heuristic formulation in chapter 3 describes the dynamic approach to solving the scheduling problem. The static environment is different from the dynamic environment because scheduling is only carried out once and all samples' arrivals are known in advance. This section mentions the differences between the static version of the heuristic and the dynamic version.

5.3.1 Changes to Heuristic Implementation for Static Environment

The first difference is that the sample cannot be pushed to the first day of the time horizon because the sample might not have arrived at that time. Sample can only be shifted to the first task date after the work request date. Similar changes are also required in the hydroshearing scheduling since hydroshearing can only start after the sample has arrived.

Secondly, checking of hydroshearing constraints in the two-phased heuristic uses the daily hydroshearing capacity for static environment as compared to the weekly hydroshearing capacity for the dynamic environment (see section 3.5.2 under Is the hydroshearing capacity for the week exceeded). This is because the samples cannot undergo hydroshearing before they arrive during the week. For example, if a sample arrives on a Thursday, it cannot be scheduled for hydroshearing on the Monday to Wednesday of its week of arrival. Hence, checking of weekly hydroshearing capacity might result in sample being scheduled for hydroshearing early in the week before it arrives.

Lastly, scheduling in the static environment only happens once. As a result, all samples are in the PRE_HYDROSHEARING stage. Hence, it does not make sense to continue to choose a sample in an earlier stage in the event of a tie in batch size for assignment in phase 1 of the two-phased heuristic. Instead, in the event of a tie in batch size, the sample with a shorter time interval is chosen in phase 1.

5.3.2 Changes to Heuristic Implementation for Relaxed Environment

The implementations in chapters 3 and 4 still applies here albeit that the hydroshearing capacity will be set to a large number.

5.4 Comparing Dynamic Scheduling and Static Scheduling

In the original problem, the samples' arrivals are unknown. This motivates the need for a myopic dynamic scheduling policy. The heuristic and IP methods are created to address this dynamic framework. It will be good to understand how the dynamic scheduling policy behaves in action. Hence, a comparison between dynamic and static scheduling is called for.

Another motivation for using static scheduling is to account for the nature of IP because of its processing overhead. In addition, if a solution exists for the IP method in the static environment, the solution will be the optimal schedule for the set of samples. This solution serves as the benchmark for comparison. By comparing dynamic and static scheduling using the IP method, a better conclusion for this comparison test can be reached.

5.4.1 Procedure

For each of the three data sets, the following experiments are conducted using both dynamic and static scheduling policy:
1) Constrained environment using IP method
2) Relaxed environment using IP method
3) Constrained environment using heuristic
4) Relaxed environment using heuristic

In the static policy, the input is the entire list of samples with their types, arrival dates and due dates. The hydroshearing capacity is set according to the environment. The number of tasks per week is 6. The number of samples per task is 6. The output is the hydroshearing and library construction schedule for each sample.

In the dynamic policy, scheduling is first carried out for samples that exist at the first decision point. At the next decision point, samples' stages are updated, samples that have undergone library construction are removed, and samples that have just arrived are added to the inventory. The scheduling at this decision point is based on this updated list of samples. This process is repeated until the last decision point. The number of tasks per week is 6. The number of samples per task is 6. For the constrained environment, the hydroshearing capacity is assumed to be uniformly distributed among the five weekdays. For example, if the hydroshearing capacity for each week is 20 and the decision point falls on a Wednesday, the remaining hydroshearing capacity for the week is 12. The final output in the dynamic policy is the hydroshearing and library construction schedule for each sample.

5.4.2 Results

The tables below show the results from the experiment. For standardization purposes, all results are tabulated into cells, like those shown in the tables below. Each cell contains 4 numbers. The number at the top in the cell is the total number of tasks required. The second number from the top is the sum of the completion time for all the samples. The third number is the maximum completion time for this set of samples. The last number is the runtime of the program. Results in subsequent sections are presented in the same format. Values can be compared horizontally.

Data Set I	Static	Dynamic
Constrained and IP Hydroshearing capacity is 30 per week.	36 tasks 1276 days 19 days 42.0s	51 tasks 1073 days 15 days 33.9s
Relaxed and IP	36 tasks 1131 days 20 days 3.49s	51 tasks 952 days 15 days 3.64s
Constrained and Heuristic Hydroshearing capacity is 30 per week.	51 tasks 1073 days 15 days 2.66s	51 tasks 1061 days 15 days 0.634s
Relaxed and Heuristic	51 tasks 952 days 15 days 2.87s	51 tasks 952 days 15 days 0.297s

Table 5.1 Experimental results for data set I.

Data Set II	Static	Dynamic
Constrained and IP Hydroshearing capacity is 20 per week.	63 tasks 1203 days 29 days 13.2s	72 tasks 1139 days 25 days 41.8s
Relaxed and IP	61 tasks 1178 days 30 days 1.46s	71 tasks 1101 days 25 days 2.9s
Constrained and Heuristic Hydroshearing capacity is 20 per week.	72 tasks 1153 days 25 days 1.66s	72 tasks 1163 days 25 days 0.674s
Relaxed and Heuristic	71 tasks 1101 days 25 days 1.52s	71 tasks 1101 days 25 days 0.313s

Table 5.2 Experimental results for data set II.

Data Set III	Static	Dynamic
Constrained and IP Hydroshearing capacity is 30 per week.	16 tasks 752 days 11 days 9.34s	17 tasks 744 days 15 days 14.1s
Relaxed and IP	16 tasks 548 days 11 days 1.48s	17 tasks 528 days 8 days 1.22s
Constrained and Heuristic Hydroshearing capacity is 30 per week.	17 tasks 744 days 11 days 5.92s	17 tasks 744 days 11 days 0.603s
Relaxed and Heuristic	17 tasks 538 days 8 days 5.83s	17 tasks 538 days 8 days 0.105s

Table 5.3 Experimental results for data set III.

5.4.3 Observations and Discussions

Number of Library Construction Tasks

The IP method in the static environment provides the least number of tasks required as compared to results in the dynamic. This is because in the static environment, arrivals are known in advance. Samples can be delayed to wait for other samples to be grouped into the same tasks, resulting in a smaller number of tasks required.

For the heuristic method, the number of tasks remains the same in both static and dynamic environments. The number of tasks is the same in all other cases because the heuristic is a greedy algorithm. Phase 2 of the heuristic shifts the sample forward to the earliest possible task and does not try to minimize the total number of task by delaying samples. Hence, results in both static and dynamic environments are the same.

Total Completion Time and Maximum Completion Time
For IP method, the total completion time is larger in the static environment as compared to the dynamic environment because samples are delayed to later tasks in order to minimize the total number of tasks. The maximum completion time, which is the longest time taken to complete processing of a sample, is also larger as a result.

For heuristic method, the total completion time for both static and dynamic scheduling is identical in the relaxed environment. The schedules in both cases are also the same. In the constrained environment, the total completion times are comparable in both static and dynamic environments for the heuristic method. Differences are due to the different choices of samples in each task. The maximum completion times are identical and reflect a similarity between heuristic in both static and dynamic environments.

Runtime
The runtime for the IP method in the constrained environment is affected by the number of variables and the number of times the model is to be solved. In the static environment, the number of variables is large since the set of samples consist of all the samples. In the dynamic environment, the number of times the model is to be solved is determined by the number of decision points. Hence, in some data sets, the static runtime is larger while in other data set, the dynamic runtime is larger. The runtime in the relaxed case is comparable because there are less variables and constraints in the IP model.

The runtime for the heuristic method is larger in the static case simply because of the larger number of variables in the static environment. Hence, solving the problem for small number of samples at multiple decision points is faster than solving the problem once for all the samples.

In conclusion, the heuristic in the relaxed framework has similar performance under both dynamic and static scheduling whereas the IP performs better in the static environment if the number of tasks is the main criterion. In comparing the scheduling policies under dynamic and static framework, the static environment can achieve a smaller number of tasks but results in a larger total completion time.

5.5 Comparing Heuristic and IP

As mentioned earlier, applying heuristic to a problem does not always gives an optimal solution. The performance of the heuristic can be compared to a benchmark that is created using the optimal solution from the IP method.

The same set of results from the previous section can also be used for comparison here. The results are reorganized for better presentation.

Data Set I	IP	Heuristic
Constrained and Static Hydroshearing capacity is 30 per week.	36 tasks 1276 days 19 days 42.0s	51 tasks 1073 days 15 days 2.66s
Relaxed and Static	36 tasks 1131 days 20 days 3.49s	51 tasks 952 days 15 days 2.87s
Constrained and Dynamic Hydroshearing capacity is 30 per week.	51 tasks 1073 days 15 days 33.9s	51 tasks 1061 days 15 days 0.634s
Relaxed and Dynamic	51 tasks 952 days 15 days 3.64s	51 tasks 952 days 15 days 0.2970s

Table 5.4 Experimental results for data set I.

Data Set II	IP	Heuristic
Constrained and Static Hydroshearing capacity is 20 per week.	63 tasks 1203 days 29 days 13.2s	72 tasks 1153 days 25 days 1.66s
Relaxed and Static	61 tasks 1178 days 30 days 1.46s	71 tasks 1101 days 25 days 1.52s
Constrained and Dynamic Hydroshearing capacity is 20 per week.	72 tasks 1139 days 25 days 41.8s	72 tasks 1163 days 25 days 0.674s
Relaxed and Dynamic	71 tasks 1101 days 25 days 2.9s	71 tasks 1101 days 25 days 0.313s

Table 5.5 Experimental results for data set II.

Data Set III	IP	Heuristic
Constrained and Static Hydroshearing capacity is 30 per week.	16 tasks 752 days 11 days 9.34s	17 tasks 744 days 11 days 5.92s
Relaxed and Static	16 tasks 548 days 11 days 1.48s	17 tasks 538 days 8 days 5.83s
Constrained and Dynamic Hydroshearing capacity is 30 per week.	17 tasks 744 days 15 days 14.1s	17 tasks 744 days 11 days 0.603s
Relaxed and Dynamic	17 tasks 528 days 8 days 1.22s	17 tasks 538 days 8 days 0.105s

Table 5.6 Experimental results for data set III.

5.5.1 Observations and Discussions

Number of Library Construction Tasks

In the static environment, the IP method uses fewer tasks than the heuristic. This is because the heuristic is a simple and greedy algorithm that blindly shifts samples forward to the earliest possible task. This does not minimize the number of tasks in the static environment.

The IP method is able to reach the theoretical minimum of tasks in data set III in the static environment because the samples overlap each other in time. Samples that have arrived earlier can be delayed until more samples arrived before carrying out the library construction. Due dates are not violated even with the delay.

In the dynamic environment, the heuristic uses the same number of tasks as the IP method.

Total Completion Time and Maximum Completion Time

In the static environment, the total completion time and maximum completion time are larger for the IP method due to the delays.

In the dynamic environment, these times are either identical or comparable. The difference is caused by the different combination of samples in the tasks.

Runtime

The runtime for the heuristic is faster in most cases except in the static and relaxed case. The runtime of the heuristic is affected by the number of samples, hence the comparable

runtime in either static or dynamic environments for the heuristic. However, the IP method performs better in the static and relaxed case because of the reduction in number of variables and constraints.

Performance of the Heuristic

The heuristic approach is to use the two-phased heuristic to tackle the library construction scheduling subproblem while considering hydroshearing constraints and then followed by hydroshearing scheduling based on the output from the two-phased heuristic. To analyze the performance of the heuristic approach, it is observed that results for number of library construction tasks, maximum completion times and runtime are similar under both constrained and relaxed environments. Therefore, it seems that the two-phased heuristic for planning the library construction schedule followed by hydroshearing schedule has created desirable results. In conclusion, the heuristic performs as well as the IP model in the dynamic framework. However, it pales in comparison for the static environment.

5.6 Comparing Schedules in Relaxed and Constrained Environments

This section examines the library construction scheduling subproblem in the absence and presence of hydroshearing constraints. This investigation uses the same set of results as the previous sections.

Data Set I	Constrained	Relaxed
IP and Static	36 tasks 1276 days 19 days 42.0s	36 tasks 1131 days 20 days 3.49s
IP and Dynamic	51 tasks 1073 days 15 days 33.9s	51 tasks 952 days 15 days 3.64s
Heuristic and Static	51 tasks 1073 days 15 days 2.66s	51 tasks 952 days 15 days 2.87s
Heuristic and Dynamic	51 tasks 1061 days 15 days 0.634s	51 tasks 952 days 15 days 0.297s

Table 5.7 Experimental results for data set I. Hydroshearing capacity is 30 per week for the constrained environment.

Data Set II	Constrained	Relaxed
IP and Static	63 tasks 1203 days 29 days 13.2s	61 tasks 1178 days 30 days 1.46s
IP and Dynamic	72 tasks 1139 days 25 days 41.8s	71 tasks 1101 days 25 days 2.9s
Heuristic and Static	72 tasks 1153 days 25 days 1.66s	71 tasks 1101 days 25 days 1.52s
Heuristic and Dynamic	72 tasks 1163 days 25 days 0.674s	71 tasks 1101 days 25 days 0.313s

Table 5.8 Experimental results for data set II. Hydroshearing capacity is 20 per week for the constrained environment.

Data Set III	Constrained	Relaxed
IP and Static	16 tasks 752 days 11 days 9.34s	16 tasks 548 days 11 days 1.48s
IP and Dynamic	17 tasks 744 days 15 days 14.1s	17 tasks 528 days 8 days 1.22s
Heuristic and Static	17 tasks 744 days 11 days 5.92s	17 tasks 538 days 8 days 5.83s
Heuristic and Dynamic	17 tasks 744 days 11 days 0.603s	17 tasks 538 days 8 days 0.105s

Table 5.9 Experimental results for data set III. Hydroshearing capacity is 30 per week for the constrained environment.

5.6.1 Observations and Discussions

Number of Library Construction Tasks

The constrained environment requires at least the same number of tasks as the relaxed environment. In some cases, like in the data set II and static environment using IP method,

the number of tasks is 63 and 61 in the constrained and relaxed cases respectively. This implies that when there are hydroshearing constraints, more tasks might be required in the library construction scheduling subproblem.

The theoretical minimum number of tasks that is required for these data sets is not reached except for data set III in the static environment when using the IP method. The theoretical minimum cannot be attained because the time interval between arrival dates and due dates of all the samples do not overlap. Samples arrive long after some earlier samples are due, resulting in a larger number of tasks required. For data set III in the static environment, the theoretical minimum is reached because the 96 samples in this data set have a time interval between arrival and due date of 3 weeks and all the samples have arrived within a one month period. The time intervals overlap in time. Hence, the samples can be packed exactly into 16 tasks without violating due date constraints.

Total Completion Time and Maximum Completion Time
The total completion time for the constrained environment is greater in all cases since samples take a longer time to wait for hydroshearing and thus, library construction is delayed. The maximum completion time is comparable and differences are due to the combination of samples in a task.

Runtime
The runtime is also larger in most cases because of the increase in the number of constraints and variables.

In conclusion, the existence of hydroshearing constraints might increase the total number of tasks required. This is referred to as the hydroshearing bottleneck. In this experiment, the comparison is conducted between a hydroshearing capacity of 20 or 30 and an infinite hydroshearing capacity. In a later section, the sensitivity of results to a smaller increase or decrease in hydroshearing capacity is investigated.

5.7 Effects of a Change in Number of Tasks in a Week on the Schedules

The next 3 sections investigate the sensitivity of the solution to varying parameters of the problems. These parameters include the number of tasks in a week, the number of samples per task, and the hydroshearing capacity for the week.

Based on the results in the previous sections, both IP and heuristic methods provide similar results under the dynamic framework. Hence, for analysis purposes, only the static environment is pursued henceforth. Furthermore, the heuristic is designed to tackle the dynamic scheduling problem and performs badly in the static environment as compared to the IP method. However, the heuristic will still be used because it often gives a smaller total completion time.

5.7.1 Procedure

Experiments are only carried out in the static environments. For each of the three data sets, the following experiments are conducted:
1) Constrained environment using IP method
2) Relaxed environment using IP method
3) Constrained environment using heuristic
4) Relaxed environment using heuristic

The number of samples per task is 6. The hydroshearing capacity is 30, 20 and 30 for data set I, II and III respectively. The number of tasks per week is made to vary between 4 and 24 for data set I and II and between 4 and 20 for data set III. The lower bound of the number of tasks per week is chosen to be the point when the problem becomes infeasible. The upper bound is chosen when there is minimal variation in results between adjacent tests.

5.7.2 Results

Results are presented in the same format as in previous sections. Each cell contains 4 numbers. The number at the top in the cell is the total number of tasks required. The second number from the top is the sum of the completion time for all the samples in days. The third number is the maximum completion time for this set of samples in days. The last number is the runtime of the program in seconds unless stated otherwise. Results in subsequent sections are presented in the same format. Values are again compared horizontally.

Data Set I	Number of tasks per week								
	4	5	6	7	8	12	16	20	24
Constrained and IP	X	36 1301 20 40.9	36 1276 19 42.0	36 1227 19 40.2	36 1227 19 38.6	36 11.85 19 35.7	36 1171 19 35.1	36 1171 19 36.5	36 1171 19 35.7
Relaxed and IP	X	36 1146 20 3.04	36 1131 20 3.49	36 1090 20 3.73	36 1090 20 3.59	36 1048 19 3.65	36 1046 19 3.59	36 1046 19 3.58	36 1046 19 3.58
Constrained and Heuristic	X	48 1124 18 3.03	52 1073 15 2.66	54 1011 15 2.50	56 990 15 2.26	61 928 14 2.00	61 916 14 2.01	61 910 14 2.14	61 906 14 1.92
Relaxed and Heuristic	X	48 1010 18 2.75	51 952 15 2.87	53 888 15 2.39	54 869 13 2.20	56 791 10 1.90	59 768 10 1.92	59 760 10 2.06	59 752 8 1.85

Table 5.10 Experimental results for variation in number of tasks per week using data set I. X means that the problem is infeasible. Hydroshearing capacity in the constrained environment is 30 per week. The theoretical bound for the minimum number of tasks is 21.

Data Set II	Number of tasks per week								
	4	5	6	7	8	12	16	20	24
Constrained and IP	X	63 1264 29 13.2	63 1203 29 13.2	63 1144 28 13.5	63 1117 28 13.6	63 1059 28 13.6	63 1039 28 14.0	63 1027 28 14.0	63 1027 28 12.8
Relaxed and IP	X	62 1233 30 1.46	61 1178 30 1.46	60 1119 30 1.42	59 1100 30 1.40	55 1073 30 1.44	51 1115 30 1.43	48 1154 30 1.37	48 1142 30 1.43
Constrained and Heuristic	X	68 1238 28 1.79	72 1153 25 1.66	73 1074 23 1.60	75 1035 20 1.48	75 947 17 1.38	75 943 17 1.33	75 927 18 1.38	75 927 18 1.37
Relaxed and Heuristic	X	68 1202 28 1.76	71 1101 25 1.52	72 1007 23 1.54	74 955 20 1.45	76 813 15 1.35	76 745 13 1.32	76 703 13 1.26	76 682 10 1.32

Table 5.11 Experimental results for variation in number of tasks per week using data set II. X means that the problem is infeasible. Hydroshearing capacity in the constrained environment is 20 per week. The theoretical bound for the minimum number of tasks is 29.

Data Set III	Number of tasks per week								
	3	4	5	6	7	8	12	16	20
Constrained and IP	16 896 14 4.13	16 830 14 11.0	16 752 11 8.36	16 752 11 9.34	16 740 11 13.7	16 740 11 9.77	16 728 11 5.61	16 728 11 9.63	16 728 11 5.84
Relaxed and IP	16 896 16 1.68	16 680 11 1.50	16 578 11 1.55	16 548 11 1.48	16 512 11 1.56	16 512 11 1.54	16 488 11 1.61	16 488 11 1.43	16 488 11 1.63
Constrained and Heuristic	X	17 820 11 2.86	17 744 11 2.66	17 744 11 2.36	17 730 11 2.29	17 730 11 1.94	17 718 11 1.97	17 718 11 1.58	17 718 11 1.61
Relaxed and Heuristic	X	16 680 11 0.165	17 568 9 2.61	17 538 8 2.22	17 504 8 2.27	17 504 8 1.88	17 478 6 1.86	17 478 6 1.51	17 478 6 1.54

Table 5.12 Experimental results for variation in number of tasks per week using data set III. X means that the problem is infeasible. Hydroshearing capacity in the constrained environment is 30 per week. The theoretical bound for the minimum number of tasks is 16.

5.7.3 Observations and Discussions

Number of Library Construction Tasks

For IP method, the number of tasks remains constant except for data set II under the relaxed environment. For data set II, samples must complete hydroshearing and library construction within 6 weeks. In the relaxed environment, all samples are hydrosheared on the day that they arrive. There are 8 samples of type 10 arriving on Monday Jun 30, 2008 and 19 samples of type 1 arriving on Thursday Jul 31, 2008. The time interval for samples of type 10 is from Jun 30 to Aug 11. The time interval for samples of type 1 is from Jul 31 to Sep 11. The time intervals of these samples overlap in time. The tasks dates that fall on this overlapping time intervals are Monday Aug 4, 2008 and Wednesday Aug 6, 2008. If there are 3 tasks each on Aug 4 and Aug 6, then 6 samples of type 1 and 10 can be assigned to these tasks. 2 more tasks are required for the other type 10 samples and 13 more tasks are required for other type 1 samples. A total of 21 (6 + 2 + 13) tasks are required. However, if there are 7 tasks on those 2 days, a total of 20 (7 + 1 + 12) tasks are required. Therefore, when there are more tasks in these two task dates, more samples can be packed into these tasks, removing the need for tasks outside the overlapping time intervals. Hence, the number of tasks decreases.

For heuristic method, the number of tasks increases with the number of tasks per week. Samples get shifted forward to the earliest possible tasks and thus occupy more and more tasks. For example, in the relaxed environment for data set I, 48 and 56 tasks are required when the number of tasks per week is 5 and 12 respectively. There are 12 samples of type 1 arriving on the first decision point in data set I. If there are 12 tasks per week, all these

12 samples would have undergone library construction in the following week. Samples that arrive later after these samples cannot be grouped into the same tasks with these 12 samples because these later samples have not been hydrosheared. When there are 5 tasks per week, only 5 samples would have undergone library construction in the following week and 7 samples would have to be delayed to later tasks. Samples that arrive later can be grouped together into tasks with these 7 samples. Hence, a smaller number of tasks is required when there are 5 tasks per week as compared to 12 tasks per week.

Total Completion Time and Maximum Completion Time

For IP method, the total completion time generally decreases with an increase in number of tasks per week. This is because scheduled tasks can be carried forward to an earlier week since there are more tasks in the earlier week. The maximum completion is either decreasing or comparable when number of tasks per week is increased. This is also due to the fact that tasks have been carried earlier in time.

For heuristic method, the total completion time is always decreasing. Samples are pushed forward to the earliest possible task and complete library construction as soon as possible, reducing completion time of the samples. The maximum completion generally decreases also because of the shifting of samples.

Runtime

The runtime remains comparable since the number of variables and constraints remains the same.

IP versus Heuristic

Comparison between IP method and heuristic can be made between alternate rows, i.e. row 1 versus row 3 and row 2 versus row 4. The same conclusions as those in previous sections can be made here. The number of tasks required is smaller for the IP method. The total completion time is smaller for the heuristic approach. Maximum completion time is comparable.

In the results of this experiment, there is an interesting point to note. The heuristic approach shows that the scheduling problem for the data set III is infeasible when the number of tasks per week is 3 whereas the IP method is able to find a solution.

This happens because samples of type with the larger batch sizes are first allocated to a library construction day in phase 1 of the two-phased heuristic. These samples occupy the latest library construction task day where possible. Hence, this clogs up some days in the middle of the planning horizon. When another sample has to scheduled, the two-phased heuristic finds that all days from the sample's arrival date to due date has been already occupied by samples (one of the three criteria that the heuristic checks before trying to assign the current sample, see section 3.5.2 Can sample be fitted into existing schedule?). Thus, the heuristic exits with an error. However, this sample could still fit into the current schedule if one of the other samples in the existing schedule is pushed forward (to before the arrival date of this sample) to make space for this sample. This is why a feasible solution exists for the IP method and not for the heuristic approach.

This problem only occurs in the static implementation of the heuristic. The problem does not occur with the dynamic implementation of the heuristic because the planning horizon starts from today and samples cannot be scheduled for any past hydroshearing slots or library construction tasks before today.

5.8 Effects of a Change in Number of Samples per Task on the Schedules

The aim here is to investigate how the schedule changes when the number of samples per tasks is changed. The theoretical bound on the minimum number of tasks required while ignoring arrival dates and due dates is similar to that given in theorem 1 and is given by

$$\max\left\{|A|,|B|,\ldots,|N|,\left\lceil\frac{|A|+\ldots+|N|}{\eta}\right\rceil\right\}$$

where η is the number of samples per task.

5.8.1 Procedure

Experiments are only carried out in the static environments. For each of the three data sets, the following experiments are conducted:
1) Constrained environment using IP method
2) Relaxed environment using IP method
3) Constrained environment using heuristic
4) Relaxed environment using heuristic

The number of tasks per week is 6. The hydroshearing capacity is 30, 20 and 30 for data set I, II and III respectively. The number of samples per task is made to vary between 2 and 24 for data set I, between 4 and 8 for data set II and between 4 and 16 for data set III.

5.8.2 Results

Data Set I	Number of samples per task										
	2	3	4	5	6	7	8	12	16	20	24
Theoretical minimum number of tasks	62	41	31	25	21	18	17	17	17	17	17
Constrained and IP	X	45 1410 21 39.9	37 1469 20 40.5	36 1353 20 40.8	36 1276 19 42.7	36 1220 19 39.86	36 1195 20 39.3	36 1134 20 38.2	36 1126 20 38.4	36 1126 20 38.1	36 1126 20 38.7
Relaxed and IP	X	45 1410 21 3.81	37 1381 21 5.23	36 1211 20 3.62	36 1131 20 3.60	36 1111 20 3.89	36 1095 20 4.50	36 1077 20 3.60	36 1077 20 3.68	36 1077 20 3.67	36 1077 20 3.67
Constrained and Heuristic	X	X	54 1121 16 2.93	52 1079 15 2.72	52 1073 15 2.66	51 1067 15 2.61	51 1063 15 2.45	51 1063 15 2.51	51 1063 15 2.56	51 1063 15 2.63	51 1063 15 2.71
Relaxed and Heuristic	X	X	53 1071 16 2.88	51 987 15 2.43	51 952 15 2.87	51 942 15 2.20	51 936 15 2.11	51 928 15 2.24	51 928 15 2.22	51 928 15 2.34	51 928 15 2.30

Table 5.13 Experimental results for variation in number of samples per task using data set I. X means that the problem is infeasible when using the method. Hydroshearing capacity in the constrained environment is 30 per week.

Data Set II	Number of samples per task							
	1	2	3	4	5	6	7	8
Theoretical minimum number of tasks	99	50	33	25	23	23	23	23
Constrained and IP	X	63 1340 29 15.7	63 1224 29 13.1	63 1209 29 13.2	63 1208 29 13.4	63 1203 29 13.6	63 1203 29 13.1	63 1203 29 13.2
Relaxed and IP	X	61 1336 30 1.54	61 1208 30 1.51	61 1186 30 1.48	61 1184 30 1.48	61 1178 30 1.50	61 1178 30 1.47	61 1178 20 1.48
Constrained and Heuristic	X	73 1170 25 1.69	72 1153 25 2.38	72 1153 25 1.53	72 1153 25 1.51	72 1153 25 1.66	72 1153 25 1.54	72 1153 25 1.55
Relaxed and Heuristic	X	73 1134 25 1.76	73 1103 25 1.59	71 1101 25 1.51	71 1101 25 1.55	71 1101 25 1.52	71 1101 25 1.51	71 1101 25 1.53

Table 5.14 Experimental results for variation in number of samples per task using data set II. X means that the problem is infeasible. Hydroshearing capacity in the constrained environment is 20 per week.

Data Set III	Number of samples per task							
	3	4	5	6	7	8	12	16
Theoretical minimum number of tasks	32	24	20	16	14	12	8	6
Constrained and IP	X	24 840 14 13.0	~	16 752 11 9.25	14 777 11 16mins	12 808 14 19.4	8 788 14 8.89	6 928 14 10.1
Relaxed and IP	X	24 680 12 1.62	~	16 548 11 2.27	14 565 11 11mins	12 560 11 1.99	8 488 11 1.90	6 608 11 2.01
Constrained and Heuristic	X	25 820 11 0.541	21 766 11 6.31	17 744 11 5.92	16 736 11 0.548	15 730 11 5.89	10 718 11 0.547	9 718 11 5.70
Relaxed and Heuristic	X	24 680 11 0.545	20 586 9 0.539	17 538 8 5.83	16 520 8 5.88	14 504 8 5.44	9 478 6 5.46	8 478 6 5.35

Table 5.15 Experimental results for variation in number of samples per task using data set III. X means that the problem is infeasible. ~ means that there is no solution after 20 mins of runtime. Hydroshearing capacity in the constrained environment is 30 per week.

5.8.3 Observations and Discussions

Number of Library Construction Tasks

The number of tasks required decreases and then stays constant as the number of samples per tasks is increased. The number of tasks decreases in the beginning because each task can only contain a small number of samples and hence more tasks are required. As the number of samples per tasks increases after a certain threshold, the number of tasks does not decrease. This is the minimum number of tasks required for the set of samples when taking due dates into considerations.

Total Completion Time and Maximum Completion Time

The total completion time also shows a similar trend like the total number of tasks. This is because as the number of samples per task increases, more samples can be placed in an earlier task reducing the total completion time. However, there is a limit to the number of samples that can be placed in the earlier tasks because of the other constraints. This results in a constant total completion time after a certain threshold. The maximum completion time is comparable under changes in the number of samples per task.

Runtime

The runtime is similar even when the number of samples per task is changed.

IP versus Heuristic

Once again, the same conclusions can be reached in the comparison between IP and heuristics. The number of tasks is smaller in the IP method. The total completion time is smaller in the heuristic method. The maximum completion time is comparable.

The same problem occurs in the test on data set I when the number of samples per task is 3. The IP method can provide a solution but the heuristic shows that the problem is infeasible. The criterion that causes the heuristic to terminate is that all dates from the arrival date to the due date are already full with samples.

5.9 Effects of a Change in Hydroshearing Capacity on the Schedules

In an earlier section on comparing relaxed and constrained environment, it is found that having hydroshearing constraints can possibly increase the number of tasks required. In this section, more effects of a smaller change in hydroshearing capacity on scheduling results are investigated.

5.9.1 Procedure

Experiments are only carried out in the static environments. For each of the three data sets, the following experiments are conducted:
1) Constrained environment using IP method
2) Relaxed environment using IP method
3) Constrained environment using heuristic
4) Relaxed environment using heuristic

The number of tasks per week is 6. The number of samples per task is 6. The hydroshearing capacity is made to vary between 20 and 100 for data set I and III and between 10 and 50 for data set II. The lower bound is chosen such that the heuristic produces an error message. The upper bound is chosen such that the results at the upper bound are similar to that of the relaxed environment. The hydroshearing capacity is assumed to be uniformly distributed among the 5 weekdays of a week.

5.9.2 Results

	Hydroshearing capacity							
Data Set I	20	25	30	35	40	60	80	100
IP	36 1334 20 34.3	36 1293 19 36.3	36 1276 19 35.3	36 1259 19 35.5	36 1244 19 34.6	36 1196 19 33.6	36 1154 19 34.0	36 1131 19 33.39
Heuristic	X	53 1102 15 2.68	52 1073 15 2.66	51 1048 15 2.69	51 1035 15 2.73	51 997 15 2.34	51 977 15 2.31	51 957 15 2.47

Table 5.16 Experimental results for variation in hydroshearing capacity using data set I. X means that the problem is infeasible when using the heuristic.

	Hydroshearing capacity				
Data Set II	10	20	30	40	50
IP	61 1178 30 12.9	61 1178 30 12.6	61 1178 30 12.6	61 1178 30 12.7	61 1178 30 12.4
Heuristic	X	72 1153 25 1.62	71 1112 25 1.60	71 1104 25 1.63	71 1101 25 1.59

Table 5.17 Experimental results for variation in hydroshearing capacity using data set II. X means that the problem is infeasible when using the heuristic.

Data Set III	Hydroshearing capacity							
	20	25	30	35	40	60	80	100
IP	16	16	16	16	16	16	16	16
	980	884	752	734	548	548	548	548
	14	14	11	11	11	11	11	11
	5.84	62.3	6.87	8.50	3.96	3.96	3.83	3.87
Heuristic		18	17	17	17	17	18	17
	X	818	744	698	666	598	558	538
		14	11	11	11	9	9	8
		0.3969	5.92	0.150	0.136	2.41	2.41	2.40

Table 5.18 Experimental results for variation in hydroshearing capacity using data set III. X means that the problem is infeasible when using the heuristic.

5.9.3 Observations and Discussions

Number of Library Construction Tasks

The number of tasks remains constant for all cases in the IP method. This is because the IP model tries to find the best combination of samples to minimize the number of tasks. All samples can be hydrosheared before their library construction tasks even when the capacity is varied. Hence, the number of tasks remains constant.

For the heuristic approach, the number of tasks remains constant after a certain threshold for data set I and II because there is more than enough hydroshearing slots for the samples and the number of tasks is determined by the hydrosheared samples. However, for data set III, the number of tasks actually increases when the hydroshearing capacity is 80. This is because the samples have not been hydrosheared in batches that is divisible by 6 like in the case of 30 and 60. 21 and 15 samples of different types arrived on Thursday Jan 21, 2010 and Friday Jan 22, 2010 respectively. By Friday Jan 22, 2010, 32 samples have been hydrosheared. The heuristic will schedule all these samples for library construction next week, needing 6 tasks. The last task of these 6 tasks only contains 2 samples. This creates an additional task. This does not happen when the hydroshearing capacity is 40 which is also not divisible by 6 because too many samples arrived on Jan 21, 2010 and Jan 22, 2010, 21 and 15 respectively. This is much more than the hydroshearing capacity of 8 per day. 16 samples will go to library construction on the following week, requiring 3 tasks and the rest of the samples are postponed to another week with the other samples that arrives later on.

Total Completion Time and Maximum Completion Time

The total completion time decreases as the hydroshearing capacity decreases. This is because library construction can be carried out earlier as hydroshearing is completed earlier. This decreases the completion times of samples. The maximum completion time remains comparable though.

Runtime

The runtime in the heuristic method is similar except for data set III. The runtime for IP method is also similar for different values of hydroshearing capacity for data set I and II.

However, in data set III, a longer runtime is required for hydroshearing capacity of 25 and 35. The results in the OPL Studio show that there are more Gomory cuts required to solve the IP model, and thus the longer runtime.

IP versus Heuristic
The same conclusions from previous sections also apply here. The number of tasks is smaller in the IP method. The total completion time is smaller in the heuristic method. The maximum completion time is comparable.

The same problem as in the previous two sections about the feasibility check of the heuristic also happens in all the data set for the smallest hydroshearing capacity shown in the tables above. The error now involves hydroshearing capacity constraints rather than library construction capacity constraints.

5.10 Effects of a Change in Priorities/Due Dates of Samples on the Schedules

The priority of a sample is related to the due dates of the sample. Priority refers to the number of weeks that a sample must complete both hydroshearing and library construction. The due date is this number of weeks plus the arrival date. Hence, in this section, the effects of a change in number of weeks for hydroshearing and library construction on the schedules are analyzed.

5.10.1 Procedure

Experiments are only carried out in the static environments. For each of the three data sets, the following experiments are conducted:
1) Constrained environment using IP method
2) Relaxed environment using IP method
3) Constrained environment using heuristic
4) Relaxed environment using heuristic

The number of tasks per week is 6. The number of samples per task is 6. The hydroshearing capacity is 30, 20 and 30 for data sets I, II and III respectively. Samples in data set I are now assumed to finish both hydroshearing and library construction within 3 weeks. For data set II and III, it is 6 and 2 weeks respectively.

For data set I and II, a random permutation of the indices of the samples is created in MATLAB. The number of weeks for the samples with indices in the first p percent of the permutated indices is decreased by a week. Scheduling is done for this new set of data. The procedure is repeated for p ranging from 10% - 50% and 60% - 100% for data set I and II respectively.

For data set III, samples arrive on 6 different days. The number of weeks is reduced by one for one sample in each day. Scheduling is done on this new data set. In the next test, we reduce the number of weeks by one for another sample in each day. If all samples in

any of the 6 days are already left with 1 week of processing time, the number of weeks for samples in that day will not be reduced. This test is repeated until the problem becomes infeasible. This methodological way of reducing the number of weeks is to identify the day which causes the infeasibility of the problem.

5.10.2 Results

	Percentage of samples with reduced number of weeks				
Data Set I	10%	20%	30%	40%	50%
Constrained and IP	47 1078 15 32.6	47 1078 15 33.0	47 1078 15 33.2	49 1063 15 32.9	X
Relaxed and IP	47 975 15 4.08	47 975 15 3.40	47 975 15 3.38	49 960 15 3.44	X
Constrained and Heuristic	52 1073 15 2.37	51 1073 15 2.29	52 1073 15 2.34	52 1073 15 2.36	X
Relaxed and Heuristic	51 952 15 2.32	51 952 15 2.33	51 952 15 2.26	51 952 15 2.29	X

Table 5.19 Experimental results for variation in number of weeks using data set I. X means that the problem is infeasible. Each sample starts with 3 weeks of processing time. The percentage in the top row refers to the percentage of samples having 2 weeks of processing time.

	Percentage of samples with reduced number of weeks				
Data Set II	60%	70%	80%	90%	100%
Constrained and IP	63 1203 30 12.9	63 1203 30 13.2	64 1293 28 12.7	65 1185 28 12.7	67 1179 25 12.1
Relaxed and IP	61 1178 30 1.31	63 1156 30 1.37	64 1146 28 1.26	65 1138 28 1.27	67 1132 25 1.26
Constrained and Heuristic	72 1153 25 1.62	72 1153 25 1.63	72 1153 25 1.60	72 1153 25 1.60	X
Relaxed and Heuristic	71 1101 25 1.64	71 1101 25 1.57	71 1101 25 1.58	71 1101 25 1.57	X

Table 5.20 Experimental results for variation in number of weeks using data set II. X means that the problem is infeasible when using the heuristic. Each sample starts with 6 weeks of processing time. The percentage in the top row refers to the percentage of samples having 5 weeks of processing time.

	Percentage of samples with reduced number of weeks						
Data Set II	10%	20%	30%	40%	50%	60%	70%
Constrained and IP	68 1169 25 12.1	68 1169 25 12.3	68 1169 25 13.5	68 1169 25 12.3	68 1169 25 12.4	68 1169 25 12.2	X
Relaxed and IP	68 1122 25 1.38	68 1122 25 1.41	68 1122 25 1.38	68 1122 25 1.37	68 1122 25 1.52	68 1122 25 1.31	X

Table 5.21 Experimental results for variation in number of weeks using data set II. X means that the problem is infeasible. Each sample starts with 5 weeks of processing time. The percentage in the top row refers to the percentage of samples having 4 weeks of processing time. The heuristic is unable to give a schedule in all these test cases.

Data Set III	Number of samples in each day with 1 week of processing time			
	1	2	3	4
Constrained and IP	16 752 11 3.24	16 752 11 3.68	17 744 11 3.51	X
Relaxed and IP	16 548 11 1.42	16 548 11 1.40	16 548 11 1.43	16 548 11 1.48
Constrained and Heuristic	17 744 11 2.34	17 744 11 2.46	17 744 11 2.43	X
Relaxed and Heuristic	17 538 8 2.30	17 538 8 2.25	17 538 8 2.26	17 538 8 2.25

Table 5.22 Experimental results for variation in number of weeks using data set III. X means that the problem is infeasible. Each sample starts with 2 weeks of processing time.

Data Set III	Number of samples in each day with 1 week of processing time				
	15	16	17	18	19
Relaxed and IP	16 548 11 1.38	16 548 11 1.35	16 548 11 1.32	17 538 8 1.23	X
Relaxed and Heuristic	17 538 8 2.79	17 538 8 2.21	17 538 8 2.22	17 538 8 2.25	X

Table 5.23 Experimental results for variation in number of weeks using data set III. X means that the problem is infeasible. Each sample starts with 2 weeks of processing time. The constrained case is infeasible for all these tests.

5.10.3 Observations and Discussions

Number of Library Construction Tasks

For the IP method, the number of tasks increases with percentage of samples with reduced number of weeks. The problem becomes infeasible after a certain threshold. For example, in relaxed environment for data set III, the infeasibility is caused by having too many samples arriving on a day. 25 samples have arrived on Jan 13, 2010. Only 18 samples can be scheduled for library construction on the Monday in the following week. Hence, the problem becomes infeasible when 19 samples on Jan 13 are only given 1 week for processing.

For the heuristic method, the number of tasks remains constant except for the constrained case in data set I with 20% of samples having 2 weeks of processing time. This is because the heuristic schedules samples sequentially and inadvertently shifts a sample forward to occupy a hydroshearing slot but does not require a new task.

Total Completion Time and Maximum Completion Time
For IP method, the total completion time decreases with an increase in number of tasks because tasks have been carried out earlier in time, reducing completion times of samples in the tasks. The maximum completion time also decreases with an increase in number of tasks for the same reason.

For heuristic method, the total completion time and maximum completion time remains constant before the problem becomes infeasible.

Runtime
The run time does not vary when percentage of samples with reduced processing time is increased.

IP versus Heuristic
In the constrained environment of data set III with 3 samples having 1 week of processing time, both IP and heuristic have the identical number of tasks, total completion time and maximum completion time.

In data set II, even when the number of weeks is reduced to 4, IP still gives a better number of tasks when the heuristic is already unable to give a schedule with 5 weeks. The heuristic is unable to give a schedule because of the failed criteria (see section 3.5.2 Can sample be fitted into existing schedule?).

5.11 Conclusion

In this chapter, we have examined the performance of the heuristic in generating schedules for hydroshearing and library construction. As compared to the IP method, the heuristic tends to solve the problem faster. Another advantage of using the heuristic is that it usually gives a smaller total completion time and maximum completion time. On the other hand, the number of library construction tasks that is required by the heuristic is almost always larger than that from the IP method.

We also compared dynamic and static scheduling. In dynamic scheduling, the heuristic performs just as well as the IP method. However, the heuristic performs poorly in terms of number of tasks in the static environment.

In addition, we also analyzed the subproblem of library construction under the relaxed environment. We find that the number of tasks does not necessarily reach the theoretical bound even without hydroshearing constraints. This is because the time intervals of samples do not overlap. Furthermore, the number of tasks can increase as a result of hydroshearing constraints.

Sensitivity analysis is also conducted on the three data sets. It is found that the total completion time decreases with an increase with number of tasks per week, number of samples per task and hydroshearing capacity. The number of tasks does not necessarily decrease with an increase in these parameters. The number of tasks can be reduced only when time intervals of samples overlap. For the heuristic approach, the number of tasks required actually increase with an increase in number of tasks per week because it is greedy and push samples to earlier tasks, thus, occupying more tasks. More tasks are required because samples are scheduled for tasks in the earlier weeks when they could have been delayed and grouped into tasks with samples that arrive later. The heuristic performs better in total completion time and maximum completion time but not in number of tasks required. When the due dates of some samples are moved to a week earlier, the problem becomes infeasible. Just before the point of infeasibility, additional tasks might be required but total completion time decreases. The tradeoff between number of tasks required and total completion time is evident in all these experiments. The above conclusions have been based on three data sets and might not be true for all possible data sets.

Some further insights are also gathered after conducting these tests. When the time interval of the samples is larger, there are more overlapping intervals between samples. This can help to reduce the number of tasks required. However, the number of tasks will not go below the theoretical bound. Secondly, when there are samples of large batch size arriving, the bottleneck is on library construction process. If there are 6 tasks in a week and the hydroshearing capacity is 20, only 6 samples of the same type can undergo library construction in each week whereas 10 samples of the same type can be hydrosheared in each week. Lastly, the bottleneck on the hydroshearing process happens when samples arrive late in the week. Samples complete hydroshearing on the Friday of week that it has been sent for hydroshearing regardless of the day that it has been sent. Hence, for every week of delay in hydroshearing, 2 days of library construction (Monday and Wednesday) or 6 tasks are wasted. By arriving late in the week, it is highly possible that samples have to wait for hydroshearing in the following week.

6 Conclusion

6.1 Accomplishments

This research project sets out to solve the hydroshearing and library construction scheduling problem for Broad Institute. Two approaches to tackle the problem have been pursued in this thesis, namely, heuristic and integer programming. In the heuristic approach, we create a two-phased heuristic to schedule samples without violating due dates and to minimize the number of library construction tasks required. In the work here, the heuristic and IP model have been implemented in MATLAB and ILOG OPL Studio respectively.

In addition, we consider a variant of the scheduling problem. The static environment assumes that samples' arrivals are known in advance. Different implementations and methods of the heuristic and IP approaches have been completed in this thesis.

The performance of the heuristic has been compared with that from the IP method in this work. The dynamic framework has been the main focus of this thesis because the arrivals of samples cannot be predicted with a good degree of accuracy. Under the dynamic framework, both approaches works equally well on the three data sets. On the other hand, in the static framework, the heuristic is inflexible (does not give a schedule when IP does), greedy and does not perform well in terms of number of tasks. However, the increase in number of tasks is traded off against the decrease in total completion time and a faster runtime.

The heuristic has been implemented in MATLAB and delivered to Broad Institute. MATLAB is the obvious choice because it is a platform that they currently have and thus do not require a purchase of new software. In addition, the software has been written to take in a Microsoft Excel spreadsheet and output in a similar format, thus circumventing the problem of requiring user's knowledge in optimization programming languages. Finally, the feedback from Broad Institute has been positive.

6.2 Future Work

Broad Institute has mentioned that a lot of their processes involve multiple stages flowshop. The problem that we have solved here is similar to some of those processes. The same algorithm can be applied to those flowshop problems if only the numerical values of the existing problems are changed. However, each new problem should still be evaluated to see if the heuristic can be applied. In addition, Broad Institute has quite a few other problems in the pipeline.

There exists another process after library construction that can be added to the scheduling problem. Libraries output from the library construction are to be placed onto PicoTiterPlate (PTP). A PTP is a technological invention by 454 Life Sciences, a Roche Company. It is a glass plate with one side polished and the other side contains 1.6 million

tiny wells. A PTP can be split into 2, 4, 8 or 16 regions. Libraries are placed on these PTP before they are placed into the sequencing machines. Libraries require a specific region on the PTP. Libraries of different identifiers are not to be placed on the same PTP. An identifier can be thought of to be similar to the type of sample as described earlier in this thesis. Furthermore, libraries from the same funding source (sponsors of the sequencing project) have to be grouped on the same PTP, for accounting and pricing purposes. The enlarged problem is to decide an optimal sequence of samples to undergo hydroshearing, followed by library construction and then PTP grouping for the sequencing technology. This can be a potential problem for future work.

Another possible future work involves theoretical proofs. The intuition of choosing the largest batch size in phase 1 of the heuristic comes from the simpler largest batch size heuristic. It appears that using the largest batch size heuristic can create a theoretical minimum number of tasks. Some theoretical proof or disproof of this intuition might help to explain why the heuristic performs as well as integer programming in our three data sets.

References

[Ahmadi1992] Ahmadi J. H., Ahmadi R.H., Dasu S., Tang C.S. (1992). ”Batching and Scheduling Jobs on Batch and Discrete Processors.” Operations Research Vol. 40, No. 4, pp. 750 - 763.

[Bagchi2006] Bagchi T. P., Gupta J. N. D., Sriskandarajah C. (2006). ”A Review of TSP Based Approaches for Flowshop Scheduling.” European Journal of Operational Research 169: 816 - 854.

[Brucker1997] Brucker P., Gladky A., Hoogeveen H., Kovalyov M. Y., Potts C. Tautenhahn T., van de Velde S. (1997). ”Scheduling a Batching Machine.” Eindhoven University of Technology, Memorandum COSOR 97-04.

[Brucker2001] Brucker P., Drexl A., Möhring R., Neumann t. and Pesch E. (2001). ”Resource-constrained Project Scheduling: Notation, Classification, Models, and Methods.” European Journal of Operational Research 112: 3 - 41.

[Caricato2007] Caricato P., Grieco A., Serino D. (2007). ”TSP-based scheduling in a batch-wise hybrid flow-shop.” Robotics and Computer-Integrated Manufacturing 23: 234 - 241.

[Choi2004] Choi J., Realff M. and Lee J. (2004). ”Dynamic Programming in a Heuristically Confined State Space: A Stochastic Resource-constrained Project Scheduling Application.” Computers and Chemical Engineering 28: 1039-1058.

[Choi2007] Choi J., Realff M. and Lee J. (2007). ”A Q-Learning-based Method Applied to Stochastic Resource Constrained Project Scheduling with New Project Arrivals.” International Journal of Robust and Nonlinear Control 17: 1214-1231.

[Elkhyari2003] Elkhyari A., Guéret C. and Jussien N. (2003). ”Solving Dynamic Resource Constraint Project Scheduling Problems Using New Constraint Programming Tools.” In: Burke E. and De Causmaecker P. (Eds.): PATAT 2002, LNCS 2740, pp. 39-59.

[Illumina2008] Illumina, Inc. Documentation, Part # 1003880, Paired-end Sequencing User Guide For Cluster Station and Genome Analyzer, http://www.rockefeller.edu/genomics/pdf/Preparing-Samples-for-Paired-End-Sequencing.pdf, 2008

[Marimuthu2005] Marimuthu S., Ponnambalam S. G. (2005). ”Heuristic Search Algorithms for Lot Streaming in a Two-Machine Flowshop.” International Journal of Manufacturing Technology 27: 174 - 180.

[Megow2006] Megow N., Uetz M. and Vredeveld T. (2006). ”Models and Algorithms for Stochastic Online Scheduling.” Mathematics of Operations Research, Vol. 31, No. 3, pp. 513-525.

[Mosheiov2004] Mosheiov G., Oron D., Ritov Y. (2004). ”Flow-Shop Batch Scheduling with Identical Processing-Time Jobs.” Published online 21 June 2004 in Wiley InterScience (www.interscience.wiley.com).

[Ng2007] Ng C. T., Kovalyov M. Y. (2007). ”Batching and Scheduling in a Multi-Machine Flow Shop.” Journal of Scheduling Vol. 10, No. 6, pp. 353 - 364.

[Pinedo2002] Pinedo M. (2002). ”Scheduling: Theory, Algorithms, and Systems (Second Edition).” Upper Saddle River, USA: Prentice-Hall; 2002.

[Potts2000] Potts C., Kovalyov M. Y. (2000). ”Scheduling with Batching: A Review.” European Journal of Operational Research 120: 228 - 249.

[Ruml2005] Ruml W., Do M. and Fromherz M. (2005). ”On-line Planning and Scheduling for High-speed Manufacturing.” Proceeding from International Conference on Automated Planning and Scheduling Workshop on Integrating Planning into Scheduling, pp. 2-11.

[Smith1956] Smith W. E. (1956). ”Various Optimizers for Single-Stage Production.” Naval Research Logistics Quarterly 3: 59 - 66.

[Stafford2002] Stafford Jr. E. F., Tseng F.T. (2002). ”Two Models for a Family of Flowshop Sequencing Problems.” European Journal of Operational Research 142(2): 282 - 293.

[Topaloglu2006] Topaloglu H. and Powell W. (2006). ”Dynamic Programming Approximations for Stochastic, Time-staged Integer Multicommodity Flow Problems.” Informs Journal on Computing, Vol. 18, No. 1, pp. 31-42.

[Wang2001] Wang J. T., Chern M. S. (2001). ”A Two-Machine Multi-Family Flowshop Scheduling Problem with Two Batch Processors.” Journal of the Chinese Institute of Industrial Engineers, Vol. 18, No. 3, pp. 77 - 85.

[Webster1995] Webster S., Baker K. R. (1995). ”Scheduling Groups of Jobs on a Single Machine.” Operations Research 43: 692 - 703.

Appendix A: Data Set I

Date Work Request Issued	Number of Weeks to Complete Library Construction	SampleID#	Organism	Stage
09/22/08	4	26695.1	1	1
09/22/08	4	26685.1	1	1
09/22/08	4	26704.1	1	1
09/22/08	4	26689.2	1	1
09/22/08	4	26707.2	1	1
09/22/08	4	26701.2	1	1
09/22/08	4	26702.2	1	1
09/22/08	4	26699.2	1	1
09/22/08	4	26694.2	1	1
09/22/08	4	26697.2	1	1
09/22/08	4	26700.2	1	1
09/22/08	4	26705.2	1	1
09/25/08	4	27491.1	2	1
09/25/08	4	27492.1	3	1
09/25/08	4	27493.1	2	1
09/25/08	4	27494.1	2	1
09/25/08	4	27495.1	2	1
09/25/08	4	27496.1	2	1
09/25/08	4	26157.3	3	1
09/25/08	4	23452.1	4	1
10/02/08	4	14028.8	5	1
10/02/08	4	25768.3	1	1
10/02/08	4	25767.3	1	1
10/02/08	4	26696.3	1	1
09/25/08	4	27497.1	2	1
09/25/08	4	27498.1	2	1
09/25/08	4	27499.1	2	1
09/25/08	4	27501.1	2	1
09/25/08	4	27503.1	2	1
09/25/08	4	27504.1	2	1
09/25/08	4	27505.1	2	1
09/25/08	4	27506.1	2	1
09/25/08	4	27500.2	2	1
09/25/08	4	27502.2	2	1
10/03/08	4	16654.2	6	1
10/03/08	4	16655.2	7	1
10/03/08	4	16657.2	8	1
10/03/08	4	16666.2	9	1

10/03/08	4	16672.2	10	1
10/03/08	4	16674.2	11	1
10/03/08	4	16679.2	12	1
10/03/08	4	16683.2	13	1
10/03/08	4	16691.2	14	1
10/03/08	4	16692.2	14	1
10/03/08	4	16693.2	15	1
10/03/08	4	17325.1	16	1
10/03/08	4	17326.2	17	1
10/03/08	4	17330.2	9	1
10/03/08	4	17331.2	18	1
10/03/08	4	17334.2	19	1
10/03/08	4	17342.2	12	1
10/03/08	4	17349.2	20	1
10/03/08	4	17350.2	21	1
10/03/08	4	22680.2	22	1
10/03/08	4	22688.2	23	1
10/07/08	4	21965.2	7	1
10/07/08	4	21974.2	7	1
10/07/08	4	21975.2	7	1
10/07/08	4	21969.2	7	1
10/07/08	4	21970.2	7	1
10/07/08	4	21971.2	7	1
10/07/08	4	21972.2	7	1
10/07/08	4	21973.2	7	1
10/07/08	4	22822.2	24	1
10/07/08	4	24472.2	25	1
10/07/08	4	24469.2	15	1
10/07/08	4	22487.2	26	1
10/07/08	4	22491.2	26	1
10/07/08	4	22492.2	26	1
10/07/08	4	22570.2	26	1
10/07/08	4	22496.2	26	1
10/07/08	4	22569.2	26	1
10/07/08	4	24470.2	27	1
10/07/08	4	22689.2	28	1
10/07/08	4	22691.2	28	1
10/07/08	4	24471.2	28	1
10/07/08	4	22696.2	14	1
10/21/08	4	22495.2	26	1
10/08/08	4	28541.1	2	1
10/21/08	4	27271.2	1	1
10/21/08	4	27262.2	1	1
10/21/08	4	22493.1	26	1
10/21/08	4	22494.1	26	1

10/21/08	4	22488.2	26	1
10/21/08	4	22489.2	26	1
10/21/08	4	22490.2	26	1
10/28/08	4	28764.1	28	1
10/28/08	4	29035.1	29	1
11/13/08	4	14631.6	26	1
11/13/08	4	13447.6	26	1
11/13/08	4	11854.8	30	1
11/26/08	4	20697.2	31	1
12/16/08	4	16661.2	32	1
12/16/08	4	17318.2	32	1
12/16/08	4	17319.2	32	1
12/16/08	4	17320.2	32	1
12/16/08	4	17323.2	32	1
12/16/08	4	17343.2	33	1
12/16/08	4	27711.2	33	1
12/16/08	4	22679.2	32	1
12/30/08	4	30501.1	34	1
12/30/08	4	30566.1	35	1
12/30/08	4	30561.1	36	1
01/16/09	4	30778.2	5	1
12/30/08	4	30567.1	36	1
12/30/08	4	30562.1	37	1
12/30/08	4	30568.1	36	1
12/30/08	4	30563.1	37	1
12/30/08	4	30569.1	36	1
12/30/08	4	30564.1	37	1
12/30/08	4	30570.1	36	1
12/30/08	4	30565.1	37	1
02/02/09	4	30663.3	38	1
02/02/09	4	30664.3	38	1
02/02/09	4	30665.3	38	1
02/02/09	4	30666.3	38	1
02/02/09	4	30667.3	38	1
02/02/09	4	30668.3	38	1
02/02/09	4	30662.1	38	1
01/22/09	4	30501.2	36	1
01/27/09	4	29170.3	5	1

Appendix B: Data Set II

Date Work Request Issued	Number of Weeks to Complete Library Construction	SampleID#	Organism	Stage
05/12/08	6	12513.5	4	1
05/12/08	6	12521.4	4	1
05/12/08	6	14327.5	4	1
05/12/08	6	14328.4	4	1
05/12/08	6	13690.3	4	1
05/12/08	6	13770.3	4	1
05/14/08	6	20103.1	5	1
05/15/08	6	20107.2	5	1
05/15/08	6	20110.2	5	1
05/15/08	6	21376.2	5	1
05/15/08	6	21377.2	5	1
05/15/08	6	21378.2	5	1
05/15/08	6	21379.2	5	1
05/15/08	6	21405.2	5	1
05/15/08	6	20108.2	9	1
05/15/08	6	21406.2	9	1
05/19/08	6	11029.5	2	1
05/19/08	6	14338.5	3	1
05/21/08	6	22683.1	6	1
05/21/08	6	22687.1	7	1
05/22/08	6	18905.2	5	1
05/27/08	6	14329.2	4	1
05/27/08	6	14326.2	4	1
05/27/08	6	13693.2	4	1
05/27/08	6	13689.2	4	1
05/27/08	6	13692.2	4	1
05/27/08	6	13688.2	4	1
05/27/08	6	13687.2	4	1
05/27/08	6	13768.2	4	1
05/27/08	6	12522.3	4	1
05/27/08	6	12527.2	4	1
05/27/08	6	12524.3	4	1
05/27/08	6	12525.2	4	1
05/27/08	6	12515.2	4	1
05/27/08	6	12516.2	4	1
05/27/08	6	12520.2	4	1
05/27/08	6	12514.2	4	1
05/27/08	6	12518.3	4	1
05/27/08	6	12519.3	4	1

05/27/08	6	12528.2	4	1
05/30/08	6	17333.1	1	1
05/30/08	6	17335.1	6	1
05/30/08	6	17344.1	6	1
05/30/08	6	17345.1	6	1
05/30/08	6	17346.1	6	1
05/30/08	6	17340.1	6	1
05/30/08	6	16656.1	8	1
05/30/08	6	17317.1	8	1
05/30/08	6	17321.1	8	1
06/06/08	6	12526.5	4	1
06/10/08	6	20104.3	9	1
06/12/08	6	25225.2	11	1
06/30/08	6	24604.2	10	1
06/30/08	6	24605.2	10	1
06/30/08	6	24606.2	10	1
06/30/08	6	24607.2	10	1
06/30/08	6	24608.2	10	1
06/30/08	6	24609.2	10	1
06/30/08	6	24614.1	10	1
06/30/08	6	24615.1	10	1
06/30/08	6	24616.1	10	1
06/30/08	6	24610.2	10	1
06/30/08	6	24611.2	10	1
06/30/08	6	24612.1	10	1
06/30/08	6	24613.1	10	1
06/30/08	6	24617.1	10	1
06/30/08	6	24618.1	10	1
06/30/08	6	24619.2	10	1
06/30/08	6	24620.1	10	1
06/30/08	6	24621.2	10	1
06/30/08	6	24622.1	10	1
06/30/08	6	24624.2	10	1
06/30/08	6	24625.1	10	1
06/30/08	6	24626.2	10	1
07/01/08	6	22765.2	10	1
07/15/08	6	20104.4	9	1
07/16/08	6	16681.3	6	1
07/21/08	6	25508.2	4	1
07/21/08	6	25509.2	4	1
07/21/08	6	25510.2	4	1
07/31/08	6	25774.2	1	1
07/31/08	6	25781.2	1	1
07/31/08	6	25763.2	1	1
07/31/08	6	25773.2	1	1

07/31/08	6	25771.2	1	1
07/31/08	6	25776.2	1	1
07/31/08	6	25784.2	1	1
07/31/08	6	25778.2	1	1
07/31/08	6	25779.2	1	1
07/31/08	6	25761.2	1	1
07/31/08	6	25766.2	1	1
07/31/08	6	25770.2	1	1
07/31/08	6	25777.2	1	1
07/31/08	6	25772.2	1	1
07/31/08	6	25767.2	1	1
07/31/08	6	25783.2	1	1
07/31/08	6	25769.2	1	1
07/31/08	6	25775.2	1	1
07/31/08	6	25765.2	1	1

Appendix C: Data Set III

Date Work Request Issued	Number of Weeks to Complete Library Construction	SampleID#	Organism	Stage
7-Jan-10	1	10001.00	1	1
7-Jan-10	1	10002.00	2	1
7-Jan-10	1	10003.00	3	1
7-Jan-10	1	10004.00	4	1
7-Jan-10	1	10005.00	5	1
7-Jan-10	1	10006.00	6	1
7-Jan-10	1	10007.00	7	1
7-Jan-10	1	10008.00	8	1
7-Jan-10	1	10009.00	9	1
7-Jan-10	1	10010.00	10	1
7-Jan-10	1	10011.00	11	1
7-Jan-10	1	10012.00	12	1
13-Jan-10	1	10013.00	13	1
13-Jan-10	1	10014.00	14	1
13-Jan-10	1	10015.00	15	1
13-Jan-10	1	10016.00	16	1
13-Jan-10	1	10017.00	17	1
13-Jan-10	1	10018.00	18	1
13-Jan-10	1	10019.00	19	1
13-Jan-10	1	10020.00	20	1
13-Jan-10	1	10021.00	21	1
13-Jan-10	1	10022.00	22	1
13-Jan-10	1	10023.00	23	1
13-Jan-10	1	10024.00	24	1
13-Jan-10	1	10025.00	25	1
13-Jan-10	1	10026.00	26	1
13-Jan-10	1	10027.00	27	1
13-Jan-10	1	10028.00	28	1
13-Jan-10	1	10029.00	29	1
13-Jan-10	1	10030.00	30	1
13-Jan-10	1	10031.00	31	1
13-Jan-10	1	10032.00	32	1
13-Jan-10	1	10033.00	33	1
13-Jan-10	1	10034.00	34	1
13-Jan-10	1	10035.00	35	1
13-Jan-10	1	10036.00	36	1
13-Jan-10	1	10037.00	37	1
21-Jan-10	1	10038.00	38	1
21-Jan-10	1	10039.00	39	1

21-Jan-10	1	10040.00	40	1
21-Jan-10	1	10041.00	41	1
21-Jan-10	1	10042.00	42	1
21-Jan-10	1	10043.00	43	1
21-Jan-10	1	10044.00	44	1
21-Jan-10	1	10045.00	45	1
21-Jan-10	1	10046.00	46	1
21-Jan-10	1	10047.00	47	1
21-Jan-10	1	10048.00	48	1
21-Jan-10	1	10049.00	49	1
21-Jan-10	1	10050.00	50	1
21-Jan-10	1	10051.00	51	1
21-Jan-10	1	10052.00	52	1
21-Jan-10	1	10053.00	53	1
21-Jan-10	1	10054.00	54	1
21-Jan-10	1	10055.00	55	1
21-Jan-10	1	10056.00	56	1
21-Jan-10	1	10057.00	57	1
21-Jan-10	1	10058.00	58	1
22-Jan-10	1	10059.00	59	1
22-Jan-10	1	10060.00	60	1
22-Jan-10	1	10061.00	61	1
22-Jan-10	1	10062.00	62	1
22-Jan-10	1	10063.00	63	1
22-Jan-10	1	10064.00	64	1
22-Jan-10	1	10065.00	65	1
22-Jan-10	1	10066.00	66	1
22-Jan-10	1	10067.00	67	1
22-Jan-10	1	10068.00	68	1
22-Jan-10	1	10069.00	69	1
22-Jan-10	1	10070.00	70	1
22-Jan-10	1	10071.00	71	1
22-Jan-10	1	10072.00	72	1
22-Jan-10	1	10073.00	73	1
28-Jan-10	1	10074.00	74	1
28-Jan-10	1	10075.00	75	1
28-Jan-10	1	10076.00	76	1
28-Jan-10	1	10077.00	77	1
28-Jan-10	1	10078.00	78	1
28-Jan-10	1	10079.00	79	1
28-Jan-10	1	10080.00	80	1
28-Jan-10	1	10081.00	81	1
28-Jan-10	1	10082.00	82	1
28-Jan-10	1	10083.00	83	1
28-Jan-10	1	10084.00	84	1

29-Jan-10	1	10085.00	85	1
29-Jan-10	1	10086.00	86	1
29-Jan-10	1	10087.00	87	1
29-Jan-10	1	10088.00	88	1
29-Jan-10	1	10089.00	89	1
29-Jan-10	1	10090.00	90	1
29-Jan-10	1	10091.00	91	1
29-Jan-10	1	10092.00	92	1
29-Jan-10	1	10093.00	93	1
29-Jan-10	1	10094.00	94	1
29-Jan-10	1	10095.00	95	1
29-Jan-10	1	10096.00	96	1