Cost-efficient response time optimization of analytical measurement processes of a clinical laboratory

Michael Asplund

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 26.4.2018

Thesis supervisor:

Prof. Tomi Laurila

Thesis advisor:

Ph.D. Antti Leinonen

Aalto University
School of Electrical Engineering

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE MASTER'S THESIS

Author: Michael Asplund

Title: Cost-efficient response time optimization of analytical measurement processes of a clinical laboratory

Date: 26.4.2018 Language: English Number of pages: 6+42

Department of Electrical Engineering and Automation

Professorship:

Supervisor: Prof. Tomi Laurila

Advisor: Ph.D. Antti Leinonen

As a result of increased demand, difficulties had been noticed at the department of therapeutic drug monitoring, abused drug and doping analysis (HuLaDo) of United Medix Laboratories Ltd (Medix) in keeping the response times promised to customers. A few different measures had been proposed in order to decrease these response times, but the relative effectiveness of the measures were still unclear.

This thesis aims to model the sample analysis process at HuLaDo as an Integer Linear Programming problem. By simulating the model using the MATLAB software by MathWorks information can be acquired to help in decision making. The main goals are to find information about how these different proposed measures would affect both the response times and the overall cost-effectiveness of the process.

The results seem positive, although more qualitative than quantitative as a result of few simulations. The results show that the problem formulation is suitable to model the sample analysis process.

Keywords: Integer Linear Programming (ILP), linear programming, response time, scheduling

AALTO-UNIVERSITETET
HÖGSKOLAN FÖR ELEKTROTEKNIK

SAMMANDRAG AV DIPLOMARBETET

Författare: Michael Asplund

Titel: Kostnadseffektiv optimering av analysers responstid vid ett kliniskt laboratorium

Datum: 26.4.2018 Språk: Engelska Sidantal: 6+42

Institutionen för elektroteknik och automation

Professur: Translational Engineering

Övervakare: Prof. Tomi Laurila

Handledare: FD Antti Leinonen

Som en följd av ökad efterfrågan, hade svårigheter noterats vid avdelningen för terapeutisk läkemedelsövervakning, drogmissbruk- och dopinganalytik (HuLaDo) vid Förenade Medix Laboratorier Ab (Medix) med att hålla de responstider som lovats åt kunder. Några olika åtgärder hade föreslagits för att minska dessa responstider men åtgärdernas relativa effekter var fortfarande oklara.

Denna avhandling ämnar modellera analytikprocessen vid HuLaDo som ett heltalsprogrammeringsproblem. Genom att simulera modellen med hjälp av MATLAB-mjukvaran från MathWorks fås information som kan hjälpa till med beslutsfattande. Huvudmålen är att hitta information om hur dessa olika föreslagna åtgärder skulle påverka både responstiderna och processens kostnadseffektivitet.

Resultaten verkar positiva, trots mer kvalitativa än kvantitativa som ett resultat av få simuleringar. Resultaten påvisar att problemformuleringen lämpar sig för att modellera analytikprocessen.

Nyckelord: Heltalsprogrammering, linjär programmering, responstid, schemaläggning

AALTO-YLIOPISTO
SÄHKÖTEKNIIKAN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Michael Asplund

Työn nimi: Kliinisen palvelulaboratorion analyysien vastausaikojen kustannustehokas optimointi

Päivämäärä: 26.4.2018 Kieli: Englanti Sivumäärä: 6+42

Sähkötekniikan ja automaation laitos

Professuuri: Translational Engineering

Työn valvoja: Prof. Tomi Laurila

Työn ohjaaja: FT Antti Leinonen

Kasvavan kysynnän seurauksena, oli havaittu ongelmia Yhtyneet Medix Laboratoriot Oy:n (Medix) huumeiden väärinkäytösten, lääkeaineiden seurannan ja dopinganalytiikan (HuLaDo) osastolla pitää asiakkaille lupaamia vastausaikoja. Erilaisia toimenpiteitä oli ehdotettu näiden vasteaikojen vähentämiseksi mutta toimenpiteiden suhteellinen tehokkuus oli vielä epäselvä.

Tämän opinnäytetyön tarkoituksena on mallintaa HuLaDo:n näyteanalyysien prosessia kokonaislukuoptimoinnin avulla. Simuloimalla mallia MathWorks:in MATLAB-ohjelmiston avulla halutaan kerätä tietoa joka auttaisi päätöksenteossa. Opinnäytetyön päätavoitteina on löytää tietoa siitä, miten nämä erilaiset ehdotetut toimenpiteet vaikuttaisivat ensinäkin vasteaikaan, ja toiseksi koko prosessin kustannustehokkuuteen.

Tulokset näyttävät myönteisiltä, vaikkakin enemmän laadullisemmilta kuin kvantitatiivisilta vähäisistä simulaatiokerroista johtuen. Tulokset osoittavat, että käytetty ongelman määritys soveltuu näyteanalyysien prosessin mallintamiseen.

Avainsanat: Lineaarinen kokonaislukuoptimointi, lineaarinen ohjelmointi, vastausaika, aikataulutus

Contents

1 Introduction

This thesis describes, and aims to optimize, a sample analysis process at the department of therapeutic drug monitoring, abused drug and doping analysis (HuLaDo) of United Medix Laboratories Ltd (Medix). Medix is a Finnish service company providing an extensive selection of laboratory services. They offer comprehensive central laboratory services to the fields of clinical chemistry, clinical microbiology, pathology and genetics as well as to therapeutic drug testing, drugs of abuse testing and drug testing in sports. They also support customers in local laboratories, covering point-of-care testing and specimen collection services at customer locations.

Medix serves the healthcare throughout Finland, such as private health clinics, public sector medical laboratories and primary care as well as drug testing in sports both nationally and internationally.

1.1 Background

As a result of increased demand over the years, difficulties had been noticed at HuLaDo in keeping the response times promised to customers. A few different measures had been proposed of how to overcome this problem and a few of them had already been implemented. However, uncertainties remained over how as of yet unimplemented measures would affect the overall process. The primary goal was naturally to decrease the response times but another large question was how eventual changes would affect the cost-effectiveness of the process.

A few of these measures were:

- Increasing work shifts during evenings
- Extending work shifts to weekends
- Investments in additional equipment
- Redesigning certain analysis
- Outsourcing certain analysis

As already mentioned some of these measures had already been implemented. During seasons of large sport events (resulting in an increase in doping related analysis) the work shifts had been temporarily extended to Saturdays. Some analysis had been redesigned for smoother and faster workflow, some were still in the process of being redesigned, while others had been deemed more appropriate to outsource. There were also plans under way to invest in an automated pipetting machine to further increase the response times of certain analysis methods.

This thesis attempts to model the sample analysis process at HuLaDo as an Integer Linear Programming (ILP) problem. An ILP problem is a type of linear programming problem where the variables are restricted to integers to better model discrete values that arise naturally in various applications. However, finding the optimum of a linear programming problem is far more straight forward compared to an ILP problem. How the problem is mathematically formulated and what methods are used to solve it affects the size of the model, how fast a solution is found and how near that solution is to the optimum. Different formulations will be discussed and compared in this thesis while the actual solving, or simulation, will be done with the commercial MATLAB software by MathWorks.

1.2 Thesis objectives

Different cases will be compared to each other by changing the parameters of the formulated ILP problem, thus hopefully giving some insight into the changes they would have on the overall process. The primary goal of this thesis will be to minimize and estimate the response times of each individual case, with the secondary goal of estimating how well these maintain the overall cost-effectiveness of the process.

The particular measures considered are the ones regarding working hours, that is increasing work shifts during evenings and extending work shifts to weekends, and ones regarding redesign of specifically chosen analysis. This will result in a total of four cases to compare, one of which will be the current day situation.

The end result of this thesis is meant to be used as an indicative tool to better understand the process and to more easily make informed decisions based on it. On the other hand, this thesis does not attempt to make any recommendations or suggestions one way or the other and only provides the results as is.

2 Theoretical background

One way to look at the optimization problem at HuLaDo is as a scheduling problem. Scheduling problems in turn have been modelled and solved extensively in the past through Mixed-Integer Linear Programming (MILP), or simply Integer Linear Programming (ILP). This chapter describes the sample analysis process at HuLaDo in more detail and presents earlier research on more or less theoretically similar cases.

2.1 The sample analysis process

In this thesis we focus on the assays and automated measurement of samples. By an assay we refer to any predefined steps which are carried out for a batch of samples in preparation for a final automated analysis. The assays consist of more or less manual work such as labeling, pipetting and mixing. During the final automated analysis, or measurement, the preprocessed samples are examined through fully automated methods such as mass spectrometry or chromatography. After analysis it is determined whether the results of the measurement are valid and acceptable or whether further tests are required. In case further tests are required a sample is reprocessed through these assay and measurement stages.

These two stages are by no means the only parts of the whole process of handling a sample which affect the response times. Instead, the total response time of a sample can be seen as the whole chain of events starting from the moment the sample is taken from a patient to the moment the results are delivered back to the customer. Prior to being processed through an assay samples need to be transported to the laboratory where they are handed to the correct department. Only after all the needed analysis are completed and the results are interpreted, which may require a sample to go through several assays and measurements, the results can be delivered to the customer. These parts are illustrated in Figure 1. In addition to the assay and measurement parts, the interpretation also has a large effect on the overall response times. The interpretation part of the chain is however not considered in the problem formulation and simulations of this thesis.

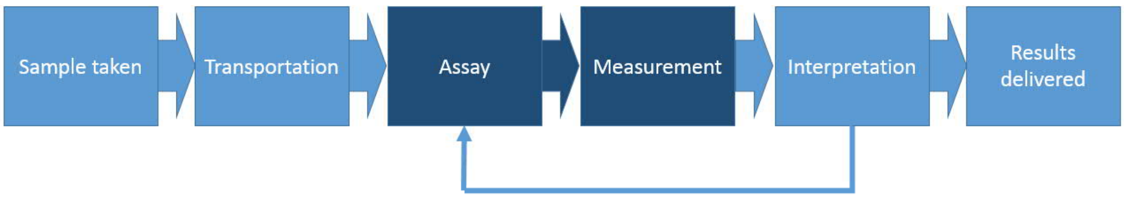

Figure 1: The chain of events samples go through. The reprocessing of some samples is indicated by the smaller arrow.

Some results are more easily interpreted than others and for some analysis an expert might need to make the decision whether further testing is required and to present the results to the customer in a more understandable form. Even so, any delays resulting from a late interpretation of the results are not taken into account in this thesis.

2.2 Mixed-Integer Linear Programming

Linear programming is only one method out of many to model and solve any particular problem, and knowing when to choose which method can seem overwhelming. In the end, there is rarely any single objectively correct method, and the problem of finding a suitable method ultimately boils down to choosing one which provides sufficient results given reasonable time and resources.

Linear programming is a method for solving optimization problems where the objective is to either maximize or minimize a value called a cost function. The cost function is subject to certain constraints and both the cost function and these constraints are linearly dependent on the variables of the problem. Going forward in this thesis it is assumed that the reader is already familiar with linear programming.

Linear programming comes with some advantages. Linear programming has been applied for decades in a wide range of applications, and because of that it is likely to find well-documented research and ready-to-use tools for any given problem. The linear formulation of the problem makes it simple and easy to understand. The simple and general form also makes it less dependent on a particular optimization problem and can be easily altered to fit possible changes to any real-world application it would aim to model.

Naturally, linear programming also has its disadvantages. The linear formulation of the cost function and constraints could be hard to fit to an extremely non-linear problem and could make the approximation of the method too large. As will be shortly mentioned in chapter 3.1.2, finding a proper balance for the cost function in optimization problems where the minimization or maximization of multiple factors simultaneously is desirable (so-called multi-objective optimization), might prove difficult and is in many cases even a matter of subjective opinion.

In many applications the variables of a model may present something that in reality is indivisible, or that one simply would prefer to keep restricted to whole numbers for the sake of simplicity. Such variables could express the amount of samples analyzed simultaneously or the hour at which they are analyzed, using the optimization problem of this thesis as an example. In such cases the linear problem formulation takes on additional constraints, restricting the variables to integers. A variable could also be further restricted to having two states to introduce a true or false situation into the formulation. These variables are called binary variables.

When linear programming models are restricted to purely integer and/or binary variables they are called Integer Linear Programming (ILP) models. If some variables are still real-valued a model is in turn called a Mixed-Integer Linear Programming (MILP) model. The solution spaces of these models are subsets of the solution spaces their corresponding relaxed problems, which is a term used to refer to the more general problems without the constraints restricting variables to discrete values. This is illustrated in Figure 2, where the outlines of the solution space consisting of integer variables is slightly smaller than the solution space without such constraints. The scales of the axes in the illustration are arbitrary. The optimums of the ILP- or MILP-problems can even differ drastically compared to the optimums of their relaxed problems, depending on the shape of the solution space of the relaxed problem.

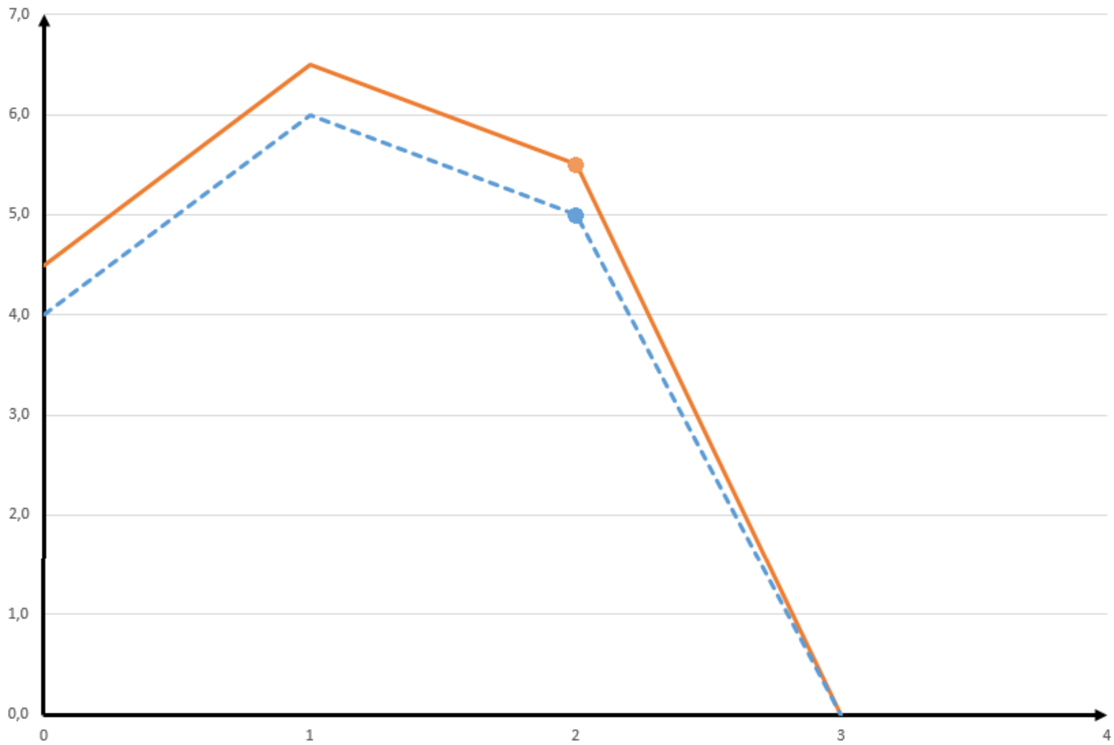

Figure 2: Simplified illustration of the restriction of a two-variable linear programming problem to an ILP.

Many methods exist to find solutions to an ILP or MILP model often including some form of heuristics, cut generation and branch-and-bound algorithms. These are all methods readily implemented into the MATLAB software by MathWorks used in this thesis. They will not be discussed much further and the discussion will instead be focused on the problem formulations. For a more detailed explanation of these methods MathWorks refer to publications by Andersen and Andersen (1995), Cornuéjols (2008), Danna et al. (2005), Berthold (2006), Mészáros and Suhl (2003), Nemhauser and Wolsey (1999), Savelsbergh (1994), and Wolsey (1998) [32, 33, 34, 35, 36, 37, 38, 39].

2.3 Scheduling problems

The goal of this thesis is to minimize the response times of the sample analysis process at HuLaDo. The response times are directly related to the arrival times of samples and the times at which they are analyzed. In other words, given the time of arrival of a number of samples, the goal is to schedule working hours and runtimes of automated analysis equipment in a way which minimizes response times. Thus, the optimization problem at HuLaDo can be seen as a scheduling problem.

A scheduling problem in general is a problem where certain events are to be scheduled in time so that a value is minimized (or maximized) while a list of constraints are to be met. This value, expressed through some mathematical expression, could be some single intuitive real-world measure such as working hours or a resource, or it could be some, possibly less intuitive, combination of such measures. As will be seen in this chapter, it is common to formulate scheduling problems as ILP or MILP models.

2.3.1 Vacation scheduling and days-off optimization

In scheduling problems involving a workforce the most natural problem might be one where shifts and working hours are scheduled. However, the somewhat converse problem of scheduling vacations and days off is one which also has gotten some attention in publications. Vacation planning might be simple enough in most cases, so that it is done manually instead of formulating a computationally solved model. However, there are cases where the planning is more complex, e.g. where long training periods are required, thus limiting the availability of temporary personnel, and where such modelling and computational solving could be advisable.

The transit industry has seen some research on the subject of vacation planning. Koutsopoulos and Wilson (1987) were early to introduce vacation planning into a workforce planning problem [2]. Much later, Dewess (2010) developed a model for a local public transport company in Germany where the full focus was on vacation scheduling [3]. Similarly, Kinnunen (2016) developed a vacation planning model for the Finnish railway operator VR Group [4]. Dewess' model was solved by a heuristic two-stage algorithm where the first stage finds an initial solution which the second stage then aims to improve upon. Kinnunen, on the other hand, solved his model by constraint programming, stating that such an approach was more efficient than solving it as an MILP model, as the formulation then required fewer constraints. This decision was in part motivated by the results by He and Qu (2012), who investigated the effectiveness of a hybrid constraint programming based column generation model in solving nurse rostering problems [5]. He and Qu stated that constraint programming is advantageous in highly constrained rostering problems, or problems where constraints are complex, where pure linear programming techniques have been applied in the past.

Azmat and Widmer (2004) considered a scheduling model taking vacation planning into account by introducing a three-step method [6]. The first two steps were designed to minimize the required workforce based on demand and to minimize weekly overtime hours. The final step introduced a heuristic method to allocate working days among employees in a way that balances the workload. Azmat et al. (2004) further improved on these ideas while proposing four nonlinear mixed integer programming formulations to replace the third step [7]. The four formulations were derived from two different approaches to minimize two different scenarios. In the first scenario each employee chooses his/her holiday weeks from a set of possible weeks and in the second the models get to assign these holiday weeks. The two different approaches mentioned differ in that the first approach aims to minimize the difference in workload among each pair of workers while the second only compares the employee with the largest workload to the employee with the smallest workload.

In both of these later publications, test-problems were generated to test the efficiency of the proposed methods. Even though the results were mostly theoretical, Swiss labor laws were taken into consideration in the problem constraints, introducing some aspects of a real-world application.

Azmat et al. noticed little overall variation among the results, drawing the conclusion that individual employee wishes have a very small impact on overtime hours. One interesting result was that simulations with a smaller workforce generally lead to a more evenly balanced workload. With a larger workforce the second approach of the second scenario even failed to find a single feasible solution.

2.3.2 Annualized working hours

For increased flexibility to fluctuations in demand, possibly because of seasonal markets, annualized working hours is one way to model a workforce scheduling problem. By annualizing working hours each employee is given a set amount of working hours for a whole year, which can then be irregularly divided over the course of a year according to demand. While indirectly working as a vacation and days-off solution as well, this method has a clear disadvantage of potentially decreasing employee satisfaction through largely irregular and fluctuating working weeks.

Beaumont (1997), Corominas et al. (2007) and Hertz et al. (2010) have all attempted to solve problems using this method through MILP formulations [8, 9, 10]. Beaumont optimized a weighted sum of three criteria; preference of long work stretches and long breaks for employees, the number of days worked in all months should be approximately equal, and the number of staff on duty each day should be proportional to demand. In contrast, the model by Corominas et al. used only a single criteria of maximizing profits. Hertz et al. formulated up to four criteria but refrained from using a weighted multi-objective model. Instead, they simulated each criterion individually so that each different simulation then could be used as its own in decision making.

The simulation times varied largely between these cases. The simulations of the model by Corominas et al. ran extremely fast, ranging from 5 seconds to 11 minutes, whereas the simulations of the model by Hertz et al. were cut off at three hours.

The simulations by Beaumont initially produced solutions that were clearly suboptimal and several iterations were required to constrain the problem in a way that would result in better solutions. Additionally, some constraints were dropped as they were considered comparatively unimportant by the client of the project. Simulations were run for a total of 1000 seconds although in most cases a solution was found during the first 100 seconds and never improved upon. The simulations by Beaumont demonstrates quite clearly how dependent the solutions of an MILP or ILP problem can be on small changes to the constraints.

In each of these three publications the simulated time span was roughly one year.

2.3.3 Project scheduling problems

Project scheduling problems are scheduling problems attempting to model the execution of one or multiple projects. These projects are divided into activities where some activities need to be completed before others can begin. In this way the activities have a certain predefined order or precedence, although some may be more dependent on previous work than others and some may be executed simultaneously. Whereas project scheduling problems have a clearer activity precedence, the events scheduled in a more general scheduling problems might not.

Artigues et al. (2016) attempted to classify and compare different formulations of ILP and MILP resource-constrained project scheduling problems [1]. The comparison was mainly done according the size of the problems (number of variables and constraints) and according to the strength of their linear programming relaxation. Artigues et al. described a general classification into three categories: time-indexed formulations, sequencing/natural-date formulations and positional-date/assignment formulations.

Time-indexed formulations use binary variables to describe discrete start and end times of activities. For each discrete point in time during the simulated time period a binary variable indicates the state (started or ended) of a single activity during that time. Sequencing/natural-date and positional-date/assignment formulations on the other hand use real-valued variables for start and end times. Instead, sequencing/natural-date and positional-date/assignment formulations use binary variables to model the precedence of activities. Artigues et al. stated that time-indexed formulations, in general, have a better linear programming relaxation at the expense of a larger size. They also noted that time-indexed formulations have been widely studied as they are easily extended to various constraints and objectives [1].

As a result of their simple form, strong linear programming relaxation (although not a guarantee of performance) and how easily extendable they are to further objectives (response times in the case of this thesis) the discussion is restricted to time-indexed formulations.

Time-indexed formulations can be further classified into three types:

- Formulations based on pulse variables
- Formulations based on step variables
- Formulations based on on-off variables

In formulations based on pulse variables, binary variables for each discrete point in time and each activity indicate whether that particular activity was started at that particular time. As the number of variables increase linearly based on the total number of activities as the time span of the model is increased, it is common for preprocessing techniques to further restrict the execution times of individual activities to a subset of the total time span of the model. Step variables resemble a rising edge of an electrical signal and indicate whether an activity was started on or before that point in time. Similarly, on-off variables resemble a rising and falling edge combination where the binary variables indicate if the corresponding activities are in the process of being executed at a particular point in time. These three types can all be obtained from one another through non-singular transformations.

Pritsker et al. (1969) and Christofides et al. (1987) described what could be seen as the more standard time-indexed formulations based on pulse variables [11, 12]. Pritsker et al. proposed the initial general formulation to which Christofides et al. proposed a modification including a so-called disaggregated precedence constraint resulting in a stronger linear programming relaxation.

The time-indexed formulation based on step variables was initially proposed by Pritsker and Watters (1968) and again later by Klein (2000) [13, 14]. A similar disaggregated formulation as the one presented by Christofides et al. for the pulse variables type can be achieved for the step variables type as well. Both the aggregated and disaggregated forms of the two have been compared by Cavalcante et al. (2001) among others, noting the stronger linear programming relaxation of the disaggregated forms [15]. Results, including the stronger relaxation by the disaggregated forms and linear transformation relationships between the pulse and step variable types, described in earlier publications were collected and presented by Möhring et al. (2001) [16].

Time-indexed formulations based on on-off variables can also be presented in both aggregated and disaggregated variants, which are once more obtainable from their respective pulse variable or step variable formulations through non-singular transformations. Kaplan (1998) proposed one formulation based on which Klein (2000) presented another [14, 18]. Demeulemeester and Herroelen (2002) later described these again as well as presented their own variant [17]. These all had the restriction that activities of 0 duration were not allowed. Artigues et al. (2013) later presented a still stronger formulation without this restriction on activity duration while also reiterating earlier results for pulse and step variable formulations [19].

Artigues et al. (2016) mentioned that the practical performance of a formulation and its linear programming relaxation strength are not necessarily related and that aggregated forms might outperform their disaggregated counterparts [1]. For experimental performance comparisons the reader is referred to research by Bianco and Caramia (2013) as well as Koné et al. (2011) [20, 21].

2.3.4 Staff scheduling and rostering

Another form of scheduling problem often discussed in literature is that of staff scheduling, or rostering. In general, one could say that vacation planning in its most simple form aims to find any acceptable solution for existing personnel, while staff scheduling often aims to estimate and minimize the number of personnel required in addition to other aspects such as overtime. In this way staff scheduling can be seen as a larger problem of which vacation planning is only one part. As already noted in chapter [sub:Vacation-scheduling], hybrid solutions exist of staff scheduling problems taking vacations into account alongside other goals.

There are multiple variations of staff scheduling problems and their mathematical formulations, often incorporating standards or legislations of any particular real-world application they aim to model. Early surveys describing personnel rostering problems of specific industries have been done by Arabeyre et al. (1969) for airline crew scheduling; by Wren (1981) for bus crew scheduling; and by Sitompul and Randhawa (1990) for nurse scheduling, to name a few [22, 23, 24]. The slightly more recent application of call centers has been considered by Grossman et al. (1999) [25].

Ernst et al. (2004a) reviewed different research that had been done so far across multiple applications while attempting to provide a classification dividing a general staff scheduling problem into subproblems, or modules. The general idea was that the modules would suggest a step by step procedure for solving a larger staff scheduling problem, although any particular problem might not require all modules and several modules might in practice be combined and solved simultaneously. They proposed a total of six modules ranging from determining the total number of employees required to assigning shifts and tasks to individual employees. They also mentioned that other classifications had been proposed earlier, such as the similar five subproblem framework suggested by Tien and Kamiyama (1982) [26]. [27]

Ernst et al. (2004a) also reviewed the different methods used to model and solve these problems. They stated that constraint programming is a powerful tool for finding feasible solutions when the problem is highly constrained or when any feasible solution will suffice. However, it is less likely to find optimal or near optimal solutions in problems with large sets of feasible solutions. Mathematical programming approaches have been common and formulations have often been different variations of the one proposed by Dantzig (1954) [30]. Still, mathematical programming formulations come with a few difficulties such as how they are more limiting in the way constraints and objective should be expressed or how implementing a good integer programming method could prove relatively difficult and time consuming. Some other models have been used, such as fuzzy set theory by Teodorovic and Lucic (1998) in solving an airline crew scheduling problem [31]. [27]

In another paper Ernst et al.(2004b) provide a broad bibliography of staff scheduling applications, including about 700 different publications [28]. A similar, more recent review, was done by Van Den Bergh et al. (2013) [29].

Recently Klemettilä (2016) considered the theoretical case of optimizing the personnel of a fictitious restaurant [40]. The work shares some similarities to this thesis as both are master's theses attempting to formulate and solve an ILP scheduling problem. The model when simulated produced four different configurations of personnel with nine-week-long work schedules. The problem was formulated in two different stages. First, a model for estimating the work shifts based on demand and regulations was formulated. After that the formulation was extended to one assigning employees to shifts while minimizing the size of the personnel. The simulation is divided into four subsequent parts, where the second and the third parts uses the solution of the first one as an initial solution. These simulations result in the different personnel configurations:

– Scheduling of monthly paid full time employees

– Additionally including hourly paid employees, often working part-time

– Including hired labor instead of hourly paid employees

- The inclusion of all three types of employees

In addition to categorizing employees by these types they are also categorized by profession; waiter/waitress, cook or restaurant manager.

Klemettilä used actual Finnish regulations and standards of the industry in formulating the constraints of the problem, such as how many consecutive work days are allowed or how close to each other work shifts can be planned. The model also takes into account how employees earn their days off although not fully implementing vacations. A full vacation planning might on the other hand have been hard to fit into a nine-week schedule without giving a false indication of performance during periods of fewer or no vacations.

3 Model formulation and simulation

This chapter describes the problem formulation of the model and how the simulations of the thesis were implemented. The mathematical formulation of the problem is described on a general level in chapter 3.1. More specific assumptions and approximations relating to the particular case at HuLaDo are described in chapter 3.2, along with the implementation of simulations.

The implementation includes how the MATLAB software by Mathworks was used to describe the problem and to run the simulations.

3.1 Integer Linear Programming model definition

The Integer Linear Programming (ILP) model is defined so that we have multiple procedures, all consisting of both an assay part and a measurement part. One procedure represents any single type of analysis in a laboratory environment, while all the procedures together represent the laboratory process as a whole. The assay part corresponds to any work phases, such as pipetting or centrifugation, leading up to a final automated analysis. The measurement part corresponds this final automated analysis, such as chromatography or mass spectrometry. The reason for this division is, that all work phases corresponding to the assay part should be carried out during working hours, while the measurement part can be scheduled to run at any time.

A procedure may be repeated multiple times during the time frame of the model. These instances will be referred to as repetitions of a procedure. In practice, these repetitions are used to represent sample batches and in further chapters where batches are mentioned it is these repetitions that the discussion is referring to.

The amount of repetitions that are needed vary depending on how many samples arrive in total during the modelled time span and how many samples can be processed simultaneously, for any specific procedure.

3.1.1 Cost function and constraints

The abbreviations used for variables and parameters in this chapter are explicitly explained where it has not been considered self-evident from the context. A complete overview of abbreviations can be found in Appendix C.

The problem formulation uses three different types of variables:

- x_{prt}^{assay}, a binary decision variable specifying whether the assay of procedure p and repetition r started at the beginning of hour t
- $x_{prt}^{measurement}$, a binary decision variable specifying whether the measurement of procedure p and repetition r started at the beginning of hour t
- q_{prd}, an integer variable specifying how many of the samples for procedure p, that arrived on day d are processed during repetition r

The cost function in equation 1 attempts to minimize the response times of samples by subtracting the mean arrival day of samples from the day the measurement (i.e. the last part of a procedure) was carried out.

Normally, the mean response time of the samples of a particular repetition r would be calculated by taking the sum of all terms

$$\frac{\text{(samples that arrived on day X)} \times \text{(days since X)}}{\text{(total amount of samples in batch)}},$$

with X representing all the different days samples processed in the batch have arrived on.

Each of these terms can be divided into two terms like so:

$$\frac{\text{(samples arr. day X)} \times \text{(day of measurement)}}{\text{(amount of samples in batch)}} - \frac{\text{(samples arr. day X)} \times \text{(day X)}}{\text{(amount of samples in batch)}}.$$

Say that we would further like to weight the importance of each batch by the number of samples in it. This would get rid of the denominator. All the terms of the second kind would then be linear in terms of our variables, however the terms of the first kind would not. The terms of the first kind add together to an expression of the total number of samples in the batch multiplied by a numerical value for the day of measurement.

To keep the cost function linear, the mean is calculated as an approximation, where the total number of samples from the sum of all terms of the first kind is replaced by an expected amount of samples processed per repetition.

The use of $(d+1)$ instead of d in the cost function is a result of even the day of arrival counting as one day of the response time. For instance, if the sample would arrive on the same day as it is processed, this should be calculated as a response time of one day, instead of zero days.

$$\text{Min.} \sum_{p \in P} \sum_{r \in R_p} \sum_{d \in D} \left[(d+1) s_p^{expected} \sum_{\tau = ds_d}^{de_d} x_{pr\tau}^{measurement} - d q_{prd} \right] \tag{1}$$

The way the cost function is defined introduces a larger error the further actual sample amounts differ from expected ones. However, constraint 2 attempts to minimize this error by requiring all arrived samples to be processed. This is done by equating the sum of all arrived samples s_{pd} for procedure p on day d to the sum of all integer variables q_{prd}.

If the approximation of the response times were to still pose a too large error for a certain type of procedure, one could apply an additional constraint, limiting the way actual sample amounts are allowed to differ from expected sample amounts.

$$\sum_{r \in R_p} \sum_{d \in D} q_{prd} = \sum_{d \in D} s_{pd}, \quad \forall p \in P \tag{2}$$

Constraints 3 and 4 ensure that the measurement parts are neither started prior to the assay parts finishing nor later than a specified maximum waiting time t^{delay}.

$$\sum_{t\in H} tx_{prt}^{measurement} - \sum_{t\in H} tx_{prt}^{assay} \geq t_p^{assay}, \qquad \forall p\in P,\ \forall r\in R_p \tag{3}$$

$$\sum_{t\in H} tx_{prt}^{measurement} - \sum_{t\in H} tx_{prt}^{assay} \leq t_p^{assay} + t^{delay}, \qquad \forall p\in P,\ \forall r\in R_p \tag{4}$$

Constraint 5 ensures that no samples are processed before they even have arrived. This is done by calculating the sum of all binary variables up until the day of any arrived sample included in that repetition, which should always be equal to 0. This is ensured by multiplying the sum with a sufficiently large coefficient m_{large} before comparison with an integer variable in the expression.

$$\sum_{\tau=d} q_{pr\tau} > m_{large} \sum_{\tau=0}^{ds_d-1} x_{pr\tau}^{assay}, \qquad \forall d\in D\backslash\{0\},\ \forall p\in P,\ \forall r\in R_p \tag{5}$$

Constraint 6 ensures that no more samples can be processed from any one day than what has arrived on that day.

$$q_{prd} \leq s_{pd}, \qquad \forall p\in P,\ \forall r\in R_p,\ \forall d\in D \tag{6}$$

Constraint 7 introduces resources into the model so that there exists a maximum of resources to be used simultaneously for every hour. Parameters ρ_{pk}^{assay} and $\rho_{pk}^{measurement}$ are binary and indicate whether execution of the assays or measurements of procedure p requires resource k. The parameter ρ_{kt}^{tot} is the total amount of that resource available at hour t.

$$\begin{aligned}\sum_{\tau=t-t_p^{assay}+1}^{t} \rho_{pk}^{assay} x_{pr\tau}^{assay} + \sum_{\tau=t-t_p^{measurement}+1}^{t} \rho_{pk}^{measurement} x_{pr\tau}^{measurement} \leq \rho_{kt}^{tot}, \\ \forall t\in H,\ \forall p\in P,\ \forall r\in R_p,\ \forall k\in\mathcal{R}\end{aligned} \tag{7}$$

Work shifts are divided so that there are normal working hours and "extended working hours" (not to be confused with the simulated case of extended working hours discussed in chapter 3.2). The workforce, which is one form of resource restricting the model, may be smaller during these extended working hours while no employees are present at all outside of working hours. Another resource restricting the model would be the available analysis equipment for the measurement parts of procedures.

Constraints 8 and 9 specify that all assays and measurements of every repetition of every procedure must be processed once and only once.

$$\sum_{t\in H} x_{prt}^{assay} = 1, \qquad \forall p\in P,\ \forall r\in R_p \tag{8}$$

$$\sum_{t\in H} x_{prt}^{measurement} = 1, \qquad \forall p\in P,\ \forall r\in R_p \tag{9}$$

Finally, constraints 10, 11 and 12 restrict decision variables to binary and sample amount variables to non-negative integers.

$$x^{assay}_{prt} \in \{0,1\}, \qquad \forall t \in H,\ \forall p \in P,\ \forall r \in R_p \tag{10}$$
$$x^{measurement}_{prt} \in \{0,1\}, \qquad \forall t \in H,\ \forall p \in P,\ \forall r \in R_p \tag{11}$$
$$q_{prd} \in \mathbb{N}, \qquad \forall d \in D,\ \forall p \in P,\ \forall r \in R_p \tag{12}$$

3.1.2 Calculation of cost-effectiveness

The cost function in equation 1 only aims to minimize response times but takes no stance on how cost-efficient a solution is. Instead, the impact a case has on the cost-effectiveness is simply calculated later. It is up to the person(s) making any decisions to take both the effects on response times as well as the effects on cost-effectiveness into consideration. Together these two estimates give a far better picture than they do individually.

The cost-effectiveness can be estimated by dividing the total cost of one case to that of another. In this work all procedures are carried out over the same amount of time with close to equal amount of samples and thus there is no need for any initial normalization of the total costs. If one were to, say, compare the total cost of one procedure during a period of one month to that of another procedure during a period of two months, it stands to reason that the later value would have to be halved before any comparison between the two (i.e. normalizing the total cost to a cost per month estimate instead).

In this thesis, the costs considered are the salaries of employees and so only the cost of assays need to be taken into account. The total cost c_{tot} of all assays is calculated in the following way:

$$c_{tot} = \sum_{p \in P} \sum_{r \in R_p} \sum_{d \in D} \sum_{\tau = ds_d}^{de_d} c^{procedure}_{p\tau} x^{assay}_{pr\tau} \tag{13}$$

where

$$c^{procedure}_{pt} = \sum_{\tau = t}^{t + t^{assay}_p - 1} c_\tau \tag{14}$$

The value c_t is an hourly cost of any assay at hour t, representing to the hourly wages of employees. This makes $c^{procedure}_{pt}$ an estimate of how much it would cost to carry out procedure p from hour t onwards until finished.

Even though our model does not take cost-effectiveness into account directly, it could be modified to do so by choosing how the importance of fast response times relates to the cost of procedures, e.g.

$$\text{Min. } r + \alpha c_{tot} \tag{15}$$

In equation 15, r would be the cost function from equation 1 and c_{tot} would be the total cost from equation 13. They are scaled to one another through a linear relation using the scalar α. This means that if we would have $\alpha = 2$, for instance, then a change in the total cost of a solution is regarded as twice the times more meaningful for the minimization problem than an equally large change in our calculation of r. Scaling multiple criteria of a optimization problem in this way is often referred to multi-objective optimization or multi-objective programming. However, finding a suitable value for α can be difficult, with a bad choice resulting in suboptimal solutions.

3.2 Simulation

In this chapter the ILP-model from chapter 3.1 will be used together with additional assumptions and approximations from chapter 3.2.2 to simulate both the manual assays and the more automated measurements of the sample analysis process at HuLaDo. The model will be simulated for different cases in order to compare them to the current day situation. The main goal is to find objective and measureable indications of how these cases could affect sample response times. Later in chapter 4 the results of these simulations are discussed in terms of how they should be interpreted and how reliable they are.

In this chapter, as well as in chapter 4, procedures are referred to by their short names. These names are short identifiers, such as OLA or 02H, by which the procedures are referred to in the digital systems at Medix. The reason they are used in this thesis is because they keep the text brief and excuses the reader from having to understanding what is analyzed in any particular procedure.

3.2.1 Simulation goals

The main goal of this thesis is to find ways to decrease sample response times in a cost-effective way. Different cases were considered in ordered to achieve this goal. Furthermore, the goal of these simulations is then to provide an objective and numerical answer to how these cases would affect both the response times and the cost-effectiveness of the process in comparison to the current day situation. The results of this thesis is to be used as an indicative tool in future decision making. However, it should be pointed out that the goal of the thesis is not to make any direct suggestions to what decisions should be made regarding the processes at HuLaDo. These simulations only focus on one aspect of the process as a whole, which in turn is only one part of the company's operations and goals. As this is the case, it should be stressed that the simulations only give insight into one piece of a larger picture and should only be used as one indicative tool among others in future decision making.

Four different cases are simulated:

- The current situation
- Older, recently modified procedures
- Newly planned procedures
- Extended working hours

These four cases are simulated in order to answer two principal questions:

- How would a number of newly planned procedures affect response times and the costs of the process?
- How would the extending of working hours affect response times and the costs of the process?

In order to answer the first question, the current situation case is compared to both a case when older, only recently modified procedures were still used, as well as to a case where newly planned, not yet applied procedure changes would be used. The comparison between the current situation and the newly planned changes is to give an indication of how these changes would affect the process, while the comparison between older changes and the current situation is to give an indication of how reliable the results regarding the newer changes would be. Since the actual effects that the older changes have had on the process are known, the simulations comparing older changes to the current situation can therefor hopefully help put the simulated results comparing the current situation to possible future changes into perspective.

In the case of older, recently modified procedures, older versions of the OLA and KLO procedures were simulated. In the case of newly planned procedures, four new procedures were simulated:

- A new procedure termed D03 in this thesis would replace procedures 01A, 02A and HES
- The existing procedure AMF is modified, making procedure 01H obsolete
- A new procedure termed H03B in this thesis would replace procedures DXP and 02H and PREGAB
- A new procedure termed ANTB in this thesis would replace procedures ANT, MOM and SME

There were a few additional procedures replaced by these newer ones. However, these are not mentioned here as they were omitted from the simulations altogether based on factors further discussed in chapter 3.2.2.

In order to answer the second question, the current situation is compared to a case where the working hours are extended to later evenings during normal working days, as well as including some working hours during Saturdays. This is not to say that an individual employee should work longer weeks but rather that the work shifts are diversified to vary more between employees. This could hopefully lead to an increased readiness and so create a more flexible flow of samples through the process. This case is not to be confused with the term “extended working hours” used in our model to describe working hours during which fewer employees are present.

Current working hours are 8-16 on Mondays through Fridays, with “extended working hours” being 16-19 on Tuesdays, Wednesdays and Thursdays during which only two employees were usually present. No work is done during the weekend. In the case where we extended the working hours, normal working hours would still be

8-16. However, the “extended working hours” would be 16-20 on Mondays through Fridays with the usual two employees present as well as 8-14 on Saturdays with three employees present.

3.2.2 Assumptions and approximations

The ILP-model used, explained in detail in chapter 3.1, imposes some restrictions on what kind of process it can represent. Naturally, the actual process is more complicated than what the model would suggest. The way the response times is calculated in the cost function in order to fit a linear model could be mentioned as one of the perhaps stricter approximation. In this chapter we will take a look at some further assumptions and approximations that have been necessary in order to fit the actual process into the mold of the model.

These assumptions and approximations, which will be explained in more detail afterwards, can be summarized as:

- The workforce is not considered to be a bottleneck of the process
- Sample batches are considered to be more or less evenly spaced out
- Only more recently arrived samples are processed in sample batches (with earlier samples processed by earlier batches)
- Sample reruns are modeled as separate procedures
- Parameters for new procedures or modifications of procedures are based on current procedures
- Some procedures have been omitted from the simulations

The available workforce is considered unlimited in the model. To be more specific, apart from the workforce restrictions of equation 7, at no point would there be a situation where no employee could be found to work during a shift. The reason for this is that during discussions with the staff at HuLaDo it became apparent that the analysis equipment (such as chromatography or mass spectrometry equipment) were to some extent the bottleneck of the process. In other words, the amount of employees available at any given moment is much less likely to be a limiting factor than the available analysis equipment of the measurement parts of procedures.

To save on computer memory and to reduce the simulation time, further restrictions have been made to the possible run times of sample batches. As new samples are expected to arrive at least somewhat uniformly throughout the whole simulated period, it would make no sense for the sample batches to be, for instance, all hastily executed during the last few days. As the possible optimality of such solutions seem unlikely, further restrictions have been made to the model which requires sample batches to be more evenly spaced out. Thus, for each batch, an “execution window” is calculated during which the assay as well as the measurement of that specific batch must be carried out.

Let us denote the total amount of simulated days by d^{tot}. Let us also denote the amount of sample batches carried out for each procedure p by r_p^{tot} and the chronological number of a batch r by i_{pr}(so that the first occurring batch of a procedure would have $i_{pr} = 1$, the second one would have $i_{pr} = 2$, and so on). The first day of the execution window e_{pr}^{start} for a batch could then be expressed as $e_{pr}^{start} = max(1,\ (i_{pr} - 1)d^{tot}/r_p^{tot}$ rounded down, while the last day of the execution window e_{pr}^{end} could be expressed as $e_{pr}^{end} = min(d^{tot},\ (i_{pr} + 1)d^{tot}/r_p^{tot}$ rounded up. This gives each batch an execution window of roughly $2d^{tot}/r_p^{tot}$ days, spare for those which are processed near the beginning or end of the simulated interval.

Which samples can be processed during a batch is similarly restricted. This assumption is made based on the educated guess that it would make sense for a batch to only process more recently arrived samples. Samples that arrive much earlier should be handled by earlier batches. The first day of this interval would be calculated as $max(1,\ (i_{pr} - 2)d^{tot}/r_p^{tot}$. The last day would be the same as the last day of the execution window of the corresponding batch, as no sample can be processed before they arrive (restricted by equation 5).

Some samples, or even whole batches, need to be reanalyzed through the process from time to time. The reasons for this range from an error in execution to a sample being flagged for further testing in order to get a more precise reading of a concentration level. These samples are bulked together as a couple of “rerun”-procedures. These procedures are completely separate from the other ones as far as the model is concerned, having their own sets of parameters.

Parameters such as sample amounts and processing times for newly planned, not yet applied procedures or modifications of procedures are estimated based on current procedures. Sample arrivals are estimated based on sample arrivals for the procedures they are substituting. Amounts of samples typically processed during sample batches and execution times are based on procedures with similar steps.

Some procedures have been omitted from the simulations. Some may have been removed as a result of the sample throughput per month being too small to have any significant impact on the overall result. Others (usually procedures related to doping control analysis) might be seasonal and only done concurrently with large sport events. Yet others may require such special methods and equipment during analysis that they could not be seen as having any significant impact on the other procedures and vice versa.

3.2.3 Data aquisition

Different parameters of the model, such as the processing time of an assay or expected amount of samples arriving per week, have been estimated based partly on definitive data from logs and reports as well as partly on estimates made by laboratory technicians and supervisors. In this chapter we will take a look at how the data acquisition was done for the simulations in order to give a better understanding of how accurate and reliable certain parameters were.

The total amount of samples arriving during any given interval has been calculated based on HuLaDo logs and reports. The way this has been estimated varies slightly between procedures.

For some procedures, data on how many samples had arrived during which days is available and easily gathered through graphical interfaces in the form of different logs and reports from Medix databases. The weekly mean and variance of a normal distribution have been calculated for some of these procedures. The means and variances have been calculated based on historical data spanning one year and the number of newly arrived samples for each day of each simulation are estimated based on them. Prior to each simulation, the number of arriving samples for each simulated week is drawn from the distribution of each procedure. These weekly amounts are then distributed among weekdays according to the distributions between weekdays noticed in the historical data.

For most procedures, however, historical data of sample arrivals on a day-to-day basis was not gathered through these interfaces. Instead, laboratory supervisors provided an estimate of expected yearly number of samples. The weekly means have been calculated from these by dividing the yearly estimate by 52. Instead of estimating a variance, the amount of samples arriving have been evenly distributed among weekdays in the simulations.

The data that could not be easily gathered through digital logs and reports were either acquired through direct interviews with employees or through forms filled out by laboratory technicians. Information that was gathered through direct interviews regarded the overall process and the state of the available equipment (e.g. how many machines with certain capabilities were in use).

Most of the data was gathered through forms. One thing asked on the forms was an estimate for a typical amount of samples processed during one sample batch of a procedure. The amount of sample batches to simulate for each procedure could then be calculated based on these estimates and the calculated amount of arriving new samples (estimated from logs and reports). Other things asked on the forms were

- Estimates for assay processing times
- Estimates for runtimes of automated analysis (corresponding to the measurement part of procedures in the model)
- What work phases the assays consisted of
- What equipment was used for the automated analysis of pre-treated samples

The forms can be read in Appendix A (in Finnish).

Parameters such as sample amounts and runtimes for newly planned, not yet applied procedures and procedure changes are estimated based on current sample arrivals for the procedures they are substituting and runtimes of procedures with similar steps.

3.2.4 Simulation execution

After formulation of the problem, as was done in chapter 3.1, and the inclusion of additional constraints, as discussed in chapter 3.2.2, the model could be simulated.

For each individual simulation, the amount of new samples arriving on each day of the time span were changed to introduce some probability into the simulations. This was to ensure that the final results would better represent a typical situation, where there exists variation in the week-to-week samples for each procedure. Multiple simulations were done for each case. This workflow, starting from the problem formulation and ending at the comparing of results, is illustrated in Figure 3.

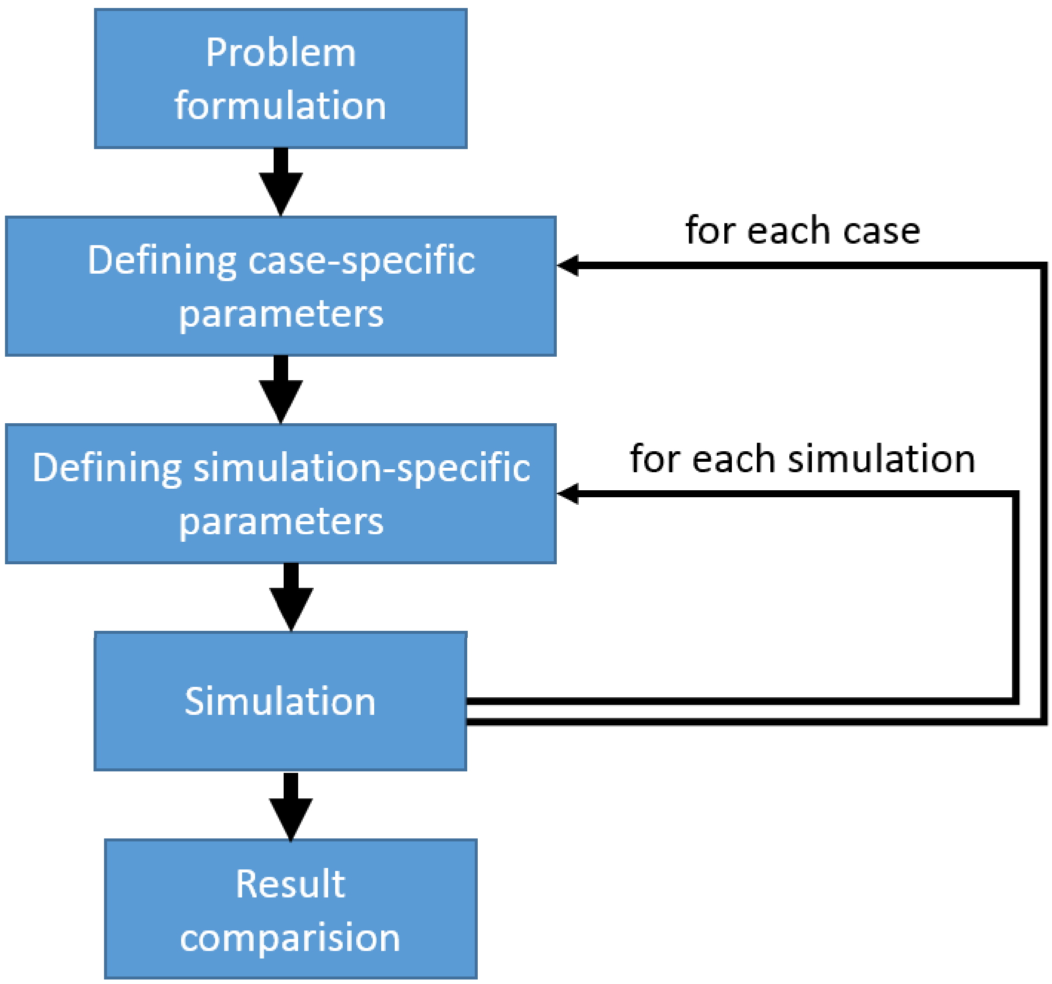

Figure 3: The workflow of model simulation of multiple cases, where the parameters of each individual simulation varies.

The simulations were done using the computer software MATLAB by the company MathWorks. In particular, the main simulation was solved by the intlinprog function. The function is a mixed-integer linear programming algorithm first introduced as part of the software package in version R2014a, released during the first half of 2014[41, 42]. It has been improved on ever since, with the arguably most relevant changes for this thesis being additions to the cut generation and heuristics options. The version R2017b was used for simulations in this thesis.

The implementation consisted of three separate code files written in MathWorks own m-file format. One of these files import CSV-formatted data containing different parameters needed for the simulations, one file does the simulations, and one file compares the results. A more detailed description of the implementation can be found in a manual written for and delivered to Medix along with the simulation code. The manual is included as Appendix B of this thesis.

After finding a solution to the relaxed linear programming problem, finding viable integer solutions proved computationally challenging and time consuming. It was however noticed during initial experimental simulations taking up to 12 hours that newer more optimal solutions were rarely found after the first hour of simulation. Ultimately, the runtime of each individual simulation was restricted to one and a half hours, with each case simulated five times. For any statistical significance, a much higher amount of simulations would have been preferable, however, a total of five simulations per case at least enabled the removal of outliers and gave a general ball park estimate of the expected changes.

4 Results

In this chapter we present and discuss the results of executed simulations both in terms of changes in response times and changes in cost-effectiveness. In sections 4.1 and 4.2 we take a look at the estimated impacts of incorporating new procedure changes and the impacts of extending working hours, respectively.

It should be pointed out that these results are only means of five simulations and therefore one should avoid drawing any direct quantitative conclusions and instead consider these as qualitative indications. As the values might vary drastically from simulation to simulation, running more simulations in order to get more reliable mean values is strongly advised. However, these results still give ballpark estimates of how the response times of different procedures, or the process as a whole, are affected by the changes made. In addition, the results give directional indications. That is, if a change could be considered as having an increasing or decreasing effect on some factor.

The results regarding the process as a whole could be seen as more reliable than results regarding individual procedures. This is because one could suspect variations to be evened out by a larger number of sample batches and higher overall volumes to downplay potentially highly volatile swings of low-volume procedures.

As in chapter 3.2, we refer to procedures by their short names.

4.1 Estimated effect of procedure changes

Simulations of a case where earlier versions of the OLA and KLO procedures were still in use were compared to simulations of the current process. The estimated impact on response times is presented in Table 1.

After an absolute change in the response times of 0.48 per cent one might conclude that the modifications made to procedures OLA and KLO ultimately had little effect on the cumulative response times of the whole process. Specific response times of OLA and KLO, especially KLO, even appear to have been negatively affected by the modifications at first glance.

However, it should be pointed out that the modifications that were done to these two procedures never made any changes to the expected sample throughput per batch. Instead, the modifications were related to what methods and equipment were used as well as the how long it took to process a batch of each procedure. Even though the total sample amounts per week has increased for OLA and KLO since introducing these changes, both cases were simulated with current amounts. With an equal amount of samples arriving each week and an equal amount of samples per sample batch, the amount of batches simulated becomes the same and thus there should be next to no simulated change to response times. In reality, this is most likely not the case as suggested by the estimated effect the changes have had on working hours as is discussed shortly.

In other words, these procedure specific differences in response times should only be interpreted as indications of how volatile response times of each procedure can be between simulations. This seems to be especially true for the procedure OXP

Procedure	Mean response time (before)	Mean response time (after)	Change (%)
01A	5.64	5.68	0.83
01H	5.93	5.67	-4.35
02A	9.40	8.93	-5.01
02H	4.48	4.48	-0.10
04M	5.68	5.71	0.63
AMF	4.58	4.51	-1.40
ANT	7.20	7.09	-1.53
BUP	4.50	4.51	0.20
BZD	4.24	4.23	-0.18
CAN	4.60	4.50	-2.17
D01	5.71	5.66	-0.86
D02_HES	4.15	4.25	2.36
DXP	9.00	9.00	0.00
H02	4.15	4.14	-0.28
H03	6.20	5.67	-8.56
H04	4.49	4.50	0.06
JD01	2.79	2.61	-6.48
JD02	2.69	2.51	-6.47
KLB	5.83	3.84	-34.05
KLO	3.69	4.42	19.66
KOR	9.33	8.96	-3.95
L01	3.40	3.29	-3.12
MET	5.69	5.56	-2.27
MNR	4.37	4.37	0.04
MOM	9.53	9.45	-0.80
OLA	9.17	9.32	1.59
OXP	2.57	4.91	90.91
PREGAB	5.93	5.93	0.00
SEU	4.76	4.65	-2.47
SIT	7.74	9.13	17.92
SME	7.77	9.09	17.03
VIT	5.78	5.52	-4.61
All procedures	4.86	4.88	0.48

Table 1: Simulated response time changes comparing old OLA and KLO procedures to current situation

Procedure	Mean response time (before)	Mean response time (after)	Change (%)
01A	5.68	0.00	0.00
01H	5.67	0.00	0.00
02A	8.93	0.00	0.00
02H	4.48	0.00	0.00
04M	5.71	5.68	-0.56
AMF	4.51	4.52	0.22
ANT	7.09	0.00	0.00
ANTB	0.00	5.68	0.00
BUP	4.51	4.70	4.33
BZD	4.23	4.05	-4.31
CAN	4.50	4.50	0.00
D01	5.66	5.66	-0.07
D02	0.00	5.69	0.00
D02_HES	4.25	0.00	0.00
D03	0.00	5.91	0.00
DXP	9.00	0.00	0.00
H02	4.14	4.25	2.70
H03	5.67	5.81	2.46
H03B	0.00	3.15	0.00
H04	4.50	4.49	-0.09
JD01	2.61	2.87	9.94
JD02	2.51	2.68	6.57
KLB	3.84	4.42	14.95
KLO	4.42	3.80	-14.04
KOR	8.96	9.45	5.47
L01	3.29	3.19	-3.10
MET	5.56	0.00	0.00
MNR	4.37	4.19	-4.04
MOM	9.45	0.00	0.00
OLA	9.32	9.06	-2.83
OXP	4.91	1.40	-71.48
PREGAB	5.93	0.00	0.00
SEU	4.65	4.59	-1.27
SIT	9.13	9.27	1.55
SME	9.09	0.00	0.00
VIT	5.52	5.97	8.19
All procedures	4.88	4.50	-7.81

Table 2: Simulated response time changes comparing the current situation to a case where newer procedure changes have been applied to the process.

Change in total work hours (%)	-4.04
Change in total sample batches (%)	-0.37
Change in salaries (%)	-4.03

Table 3: Changes to cost-effectiveness observed in simulations comparing old OLA and KLO procedures to the current situation.

Change in total work hours (%)	-12.77
Change in total sample batches (%)	-14.58
Change in salaries (%)	-12.71

Table 4: Changes to cost-effectiveness observed in simulations comparing the current situation to a case where newer procedure changes have been applied to the process.

with an eyesore of a 90 per cent increase in response times. OXP is a procedure with such a low amount of samples processed per year that it was on the verge of being omitted from the simulations altogether, which might be one explanation for its high volatility. The benefits of modifications to OLA and KLO will be more clear when considering the effect that they have had on work hours.

Looking at the simulated changes of newly planned procedures, or modifications to procedures, in Table 2 we see that the decrease in overall response times are noticeable. These newly panned modifications differ from the previous OLA and KLO modifications in that they also change the sample throughput of each batch. Estimated as a decrease of 7.81 per cent in response times, the result is a decrease of almost half a day to the overall process. The means of the simulated response times for newer procedures and modifications are all less than a week while some current ones still have simulated means above that (e.g. 9.09 days for SME or 9.45 days for MOM). Again, we see that the response times of OXP vary heavily with a supposed decrease of 71.48 per cent.

In Tables 3 and 4 we see the effects to cost-effectiveness of older modifications and newly planned modifications respectively. As mentioned earlier, the modifications to the OLA and KLO procedures should have affected processing times, even though no significant differences were noticed for the simulated response times. This seems to be in line with the estimated 4.04 decrease in estimated total work hours.

The simulation of newly planned procedures and modifications of procedures resulted in an additional 12.77 per cent decrease in total work hours. This seems promising although one should keep in mind that this tells little about the effects of individual procedure changes and is only an indication of their combined effect. In other words, there may be synergies at play between the newly planned modifications which are not evident from these results.

4.2 Estimated effect of extended working hours

Simulations of the current process were compared to simulations of a case where working hours were extended. Table 5 summarizes the estimated effects these extended working hours would have on response times.

Procedure	Mean response time (before)	Mean response time (after)	Change (%)
01A	5.68	6.28	10.48
01H	5.67	5.65	-0.35
02A	8.93	9.17	2.74
02H	4.48	4.56	1.91
04M	5.71	5.28	-7.62
AMF	4.51	4.84	7.26
ANT	7.09	7.50	5.86
BUP	4.51	3.90	-13.44
BZD	4.23	3.98	-6.07
CAN	4.50	4.33	-3.70
D01	5.66	6.44	13.86
D02_HES	4.25	3.08	-27.49
DXP	9.00	9.37	4.07
H02	4.14	3.84	-7.15
H03	5.67	6.00	5.83
H04	4.50	4.45	-1.10
JD01	2.61	2.53	-2.97
JD02	2.51	2.45	-2.42
KLB	3.84	6.92	80.02
KLO	4.42	2.87	-35.18
KOR	8.96	8.37	-6.62
L01	3.29	3.07	-6.78
MET	5.56	5.38	-3.26
MNR	4.37	3.38	-22.73
MOM	9.45	10.17	7.61
OLA	9.32	9.26	-0.62
OXP	4.91	1.71	-65.08
PREGAB	5.93	6.20	4.49
SEU	4.65	5.08	9.25
SIT	9.13	7.86	-13.87
SME	9.09	8.42	-7.42
VIT	5.52	5.10	-7.56
Kaikki työt	4.88	4.55	-6.86

Table 5: Simulated response time changes comparing the current situation to a case where working hours are extended.

Change in total work hours (%)	0.18
Change in total sample batches (%)	0.00
Change in salaries (%)	0.80

Table 6: Changes to cost-effectiveness observed in simulations comparing the current situation to a case where working hours are extended.

As was the case for simulations regarding procedure modifications, estimated response times of individual procedures vary heavily and might be deceitful. Especially so for the procedure KLB, where the changes to working hours supposedly increase response times by roughly 80 per cent. This is most likely the result of a suboptimal outlier in the simulations. Again, the effects on the overall process are more reliable and estimate a decrease in response times of 6.86 per cent.

Table 6 indicates that these changes would have no effect on total work hours but a slight increase in salaries (0.80 per cent). This is a result of later shifts and work done during Saturdays, which have a higher hourly wage.

4.3 Reliability of results

The estimation of margins of error for simulations of an ILP- or MILP-formulation is multifaceted. There can be multiple sources of error, from problem formulation to actual simulation, and each of these needs to be investigated individually. Firstly, the problem formulation attempts to model a real-world process. This model is only an approximation and the larger the approximation the larger the error. Secondly, each individual simulation might or might not find a near-optimal solution. How far the found solution lies from the optimum is another source of error. Thirdly, if multiple simulations are done, each with slightly different parameters, as in this thesis, the combined estimate of these simulations introduce yet one type of error. The better the simulations together represent the typical case, the smaller the error is.

As already mentioned, only five simulations were done per case so how well these together can estimate a typical situation is debatable. Response time changes for individual procedures varied heavily and this was most likely the cause of high variation between simulations. The overall results of the whole process were more reliable but even then the results are more qualitative than quantitative. More simulations would eventually have smoothed out these variations.

When an algorithm such as a branch and bound algorithm is used, there are upper and lower bounds between which the optimum is expected to be. The gap between these two are adjusted based on new feasible points found. An initial absolute lower bound (in the case where the goal is to find a minimum of the cost function) would naturally be the solution of the relaxed linear programming problem without any integer and binary constraints on the variables. The MATLAB-software outputs this gap at the end of a simulation as a percentage of the upper bound (called the relative gap), calculated in the following way:

$$relative\ gap = 100 * (U - L)/(|U| + 1), \tag{16}$$

where U and L are the upper and lower bounds, respectively. In the total 20 simulations done this gap varied from as low as 0.34 % up to as much as 17 %. The median of these were at 2.9 %. These gaps were noticeably higher for the case of extended work hours which might be the result of a less constrained problem, leading to more possibilities.

Lastly, the error introduced by the problem formulation itself should be investigated in some way. The most reliable way to estimate this would probably be to compare the results of problem simulations with actual data of the process changes once available. However, this would first require the changes to be implemented and in many cases it even takes a considerable amount of time to gather this data. In some cases, any directly comparable data may never be gathered at all. Still, it could be argued that, in this thesis, the largest approximation of the model comes from expressing the response times linearly in the cost function. This linearization expected actually simulated batch sizes to be close enough to a parameter representing the typical batch size of such procedures.

In Figure 4 we can see that this was not nearly always the case. It should be pointed out that the possible points in the figure are ratios of discrete values (and thus discrete themselves) and the frequency of occurrence of each point is not visible. The total amount of batches represented is 2016. A slight symmetry is also noticeable in the figure, as a lower than typical amount of samples in one batch most likely leads to a larger amount of samples in another batch, as all samples need to be accounted for in the end.

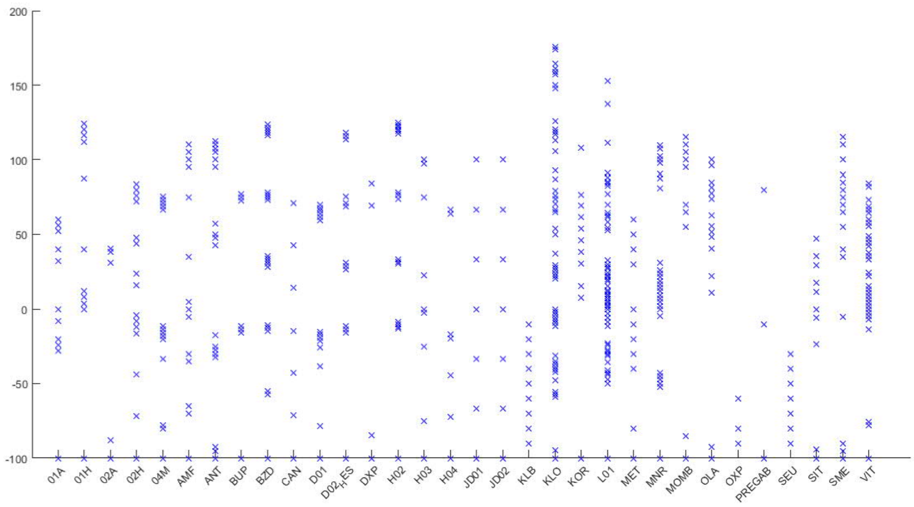

Figure 4: The actual batch sizes simulated compared to a typical batch size estimate. The actual batch sizes are expressed as a percentage of the typical batch size.

5 Discussion

This chapter summarizes and compares the results to previously published research. The initial goals of the thesis are once again stated while attempting to answer how they were met.

5.1 Model suitability

The work of this thesis went through several iterations before it was deemed suitable for its use case and the final simulated solutions were reached. Initially, this meant changes to the problem formulation to better model the analysis process, after which the implementation of the MATLAB m-code went through variations to more precisely and optimally model the problem. So did Beaumont (1997) need to modify the original formulation according to further information from the customer and to avoid clearly suboptimal solutions, as already mentioned in chapter 2.3.2 [8]. This gives further credit to the statement by Ernst et al. (2004a) that implementing a good integer programming method can prove difficult and time consuming [27].

Still, the solutions reached by the simulations ultimately showed realistic results, showing that the model, although approximative in its linearity, can be used to simulate the analysis process at HuLaDo. Further, it shows that such formulations can be of use when estimating and minimizing response times of similar processes more generally.

Additionally, apart from the way in which response times were modelled in this thesis, the extremely broad bibliography by Ernst et al. (2004b) and the equally broad reviews by both Ernst et al. (2004a) and Van Den Bergh et al. (2013) indicate through extensive lists of different applications just how generally applicable linear programming formulations are to scheduling problems [28, 27, 29]. What started off as minimizing traffic delays at toll booths with the initial formulation by Dantzig (1954) has now spread over to areas that are very different in practice, such as the different rostering applications considered in chapter 2.3.4 [30].

5.2 Applicability of the results

The results of the simulations were positive overall, albeit vague because of only few simulations. Still, the results give indicative information that seems to be aligned with the positive experiences they have had at HuLaDo for the earlier modifications to the OLA and KLO modifications. The variations between simulations are most likely largely due to the probabilistic nature of how the parameters are set up (as discussed in chapter 3.2.3). Naturally, part of the variation can also be attributed to how well the simulations manage to find feasible solutions that lie closer to the optimal.

The runtimes of the individual simulations were restricted to one and a half hour even though many simulations rarely found any better solutions after the first half hour. This is comparable to what Beaumont (1997) described to have experienced, although this probably is true for a lot of cases [8]. Azmat et al. (2004) had noticed little variation among their results which might be a result of the long runtime used (at least over 15 hours) [7].

Where Klemettilä (2016) used specific regulations and standards of the industry as additional constraints, this work implemented the hourly wages from employee contracts into the calculation of cost-effectiveness [40]. This can be seen as a direct result of the fact that the automated equipment turned out to be the bottle neck in this thesis. As a result, the individual employees and their shifts were never considered and it was simply assumed that the workforce was constant and sufficiently large. For the same reason vacations, days off and differences among employees (e.g. in what analysis they are trained to process) conveniently never had to be considered. However, if Medix ever decided to estimate a minimal required workforce a slightly different model would have to be formulated.

Still, some similarities remained between the simulations of Klemettilä and those in this thesis. Klemettilä simulated four different configurations of employees and work shifts where this work simulated four different cases. In this way both provided multiple point of views to help in decision making.

The indicative nature of the simulations give insight into how the response times are affected by different changes to the analysis process at HuLaDo. The additional estimations of work hours and costs further indicate how the cost-effectiveness is affected. These were the two main goals of the thesis. Secondly, by changing the parameters of the provided implementation it is relatively simple to simulate and estimate the effects of possible future changes to procedures or work hours.

6 Conclusions

In this thesis, a general ILP-model was formulated to minimize the response times of analytical measurements of a clinical laboratory. The model was used in the simulations of four different cases in order to estimate the effects they would have at the HuLaDo-department at Medix. These cases, when compared to each other, were to give some insight into how certain procedure changes and changes to work hours would affect both the response times and the cost-effectiveness of the process. The simulations were done with the MATLAB-software by MathWorks.

The problem formulation used proved to be suitable to model the analysis process at HuLaDo. The results gave indicative answers to how both the response times and the cost-effectiveness of the process is affected by the specified changes to procedures and work hours.

When comparing simulations with earlier versions of the OLA and KLO procedures to the current day situation the response times of the process did not change much. However, a clear decline of about 4 % in overall work hours was noticed. This decline indicates that some of the workforce could be released as a result of these changes and reassigned to indirectly affect the response times. As the number of batches simulated in both cases were equal, such indirect effect on response times is not shown.

When comparing newly planned modifications of certain procedures to the current day situation a clear decline in both response times (roughly 8 %) and work hours (roughly 13 %) was noticed. As the simulations were done with either all or none of the modifications included, the effect of individual changes remains unknown.

The comparison of the current process to one where working hours were extended to evenings and weekends saw only a slight increase in overall salaries (roughly 1 %). On the other hand, the simulations estimated a reduce to overall response times of about 7 %.

For more precise results, especially regarding the changes to individual procedures, further simulations would have been required. In future situations where one or few procedures could be successfully isolated, as a result of having little or no effect on the rest of the process, the size of the model and the run times of such more specific simulations could be drastically reduced.

The approximation of response times to a linear model was a large one. It would be interesting to see how the problem formulation and simulations of this thesis would compare to a non-linear model. Additionally, a time-indexed formulation was used like the ones described in chapter 2.3.3, and it could be worth considering one of the other two types (a sequencing/natural-date formulation or a positional-date/assignment formulation) in the hopes of reaching a similar simulation performance for a fraction of the model size.

The problem formulation developed in this thesis held an expression for the response times a cost function, while treating the automated measurement equipment as a limited resource. Another way this problem could be considered is treating the response times promised to customers as a constraint and treating the equipment instead as the variable. In this way you could estimate the minimal equipment required to meet customer demands. This other formulation could even be seen as closer to a staff scheduling problem; instead of minimizing the workforce one would minimize the needed equipment which both are schedulable resources.

References

[1] Artigues, C., Koné O., Lopez P. and Mongeau M. (2016). *Mixed-Integer Linear Programming Formulations.* In: Handbook on Project Management and Scheduling, vol. 1, Springer, pp.17-41.

[2] Koutsopoulos, H.N. and Wilson, N.H.M. (1987). *Operator workforce planning in the transit industry.* Transportation Research Part A: General, 21(2), pp. 127-138.

[3] Dewess, S. (2010). *Socially acceptable annual holiday planning for the crew of a local public transport company in Germany.* Public Transport, 2(1), pp. 25-49.

[4] Kinnunen, T., Punkka, A. and Bartolini, E. (2016). *Cost-efficient vacation planning with variable workforce demand and manpower.* Master's thesis, School of Science, Aalto University, Espoo.

[5] He, F. and Qu, R. (2012). *A constraint programming based column generation approach to nurse rostering problems.* Computers & Operations Research, 39(12), pp. 3331-3343.

[6] Azmat, C.S. and Widmer, M. (2004). *A case study of single shift planning and scheduling under annualized hours: A simple three-step approach.* European Journal of Operational Research, 153(1), pp. 148-175.

[7] Azmat, C.S., Hürlimann, T. and Widmer, M. (2004). *Mixed Integer Programming to Schedule a Single-Shift Workforce under Annualized Hours.* Annals of Operations Research, 128(1-4), pp. 199-215.

[8] Beaumont, N. (1997). *Using Mixed Integer Programming to Design Employee rosters.* The Journal of the Operational Research Society, 48(6), pp. 585–590.

[9] Corominas, A., Lusa, A. and Pastor, R. (2007). *Using a MILP model to establish a framework for an annualised hours agreement.* European Journal Of Operational Research, 177(3), pp. 1495-1506.

[10] Hertz, A., Lahrichi, N. and Widmer, M. (2010). *A flexible MILP model for multiple-shift workforce planning under annualized hours.* European Journal Of Operational Research, 200(3), pp. 860-873.

[11] Pritsker, A., Watters, L. and Wolfe, P. (1969). *Multi-project scheduling with limited resources: a zero-one programming approach.* Management Science, 16(1), pp. 93-108.

[12] Christofides, N., Alvarez-Valdés, R. and Tamarit, J. (1987). *Project scheduling with resource constraints: a branch and bound approach.* European Journal of Operational Research, 29(1), pp. 262-273.

[13] Pritsker, A. and Watters, L. (1968). *A zero-one programming approach to scheduling with limited resources.* The RAND Corporation, RM-5561-PR.

[14] Klein, R. (2000). *Scheduling of resource-constrained projects.* Kluwer, Boston u.a.

[15] Cavalcante C.C.B., de Souza C.C., Savelsbergh M.W.P., Wang Y. and Wolsey L.A. (2001). *Scheduling projects with labour constraints.* Discrete Applied Mathematics, 112(1), pp. 27-52.

[16] Möhring, R.H., Schulz, A.S., Stork, F. and Uetz, M. (2001) *On project scheduling with irregular starting time costs.* Operations Research Letters, 28(1), pp. 149-154.

[17] Demeulemeester, E.L. and Herroelen, W. (2002). *Project scheduling: a research handbook.* Kluwer, Boston u.a.

[18] Kaplan, L.A. (1998). *Resource-constrained project scheduling with preemption of jobs.* Unpublished Ph.D. dissertation, University of Michigan, USA.

[19] Artigues, C. (2013). *A note on time-indexed formulations for the resource-constrained project scheduling problem.* Technical Report 13206, LAAS, CNRS, Toulouse.

[20] Bianco, L. and Caramia, M. (2013). *A new formulation for the project scheduling problem under limited resources.* Flexible Services and Manufacturing Journal, 25(1-2), pp. 6-24.

[21] Koné, O., Artigues, C., Lopez, P. and Mongeau, M. (2011). *Event-based MILP models for resource-constrained project scheduling problems.* Computers & Operations Research, 38(1), pp. 3-13.

[22] Arabeyre, J.P., Fearnley, J., Steiger, F.C. and Teather, W. (1969). *The airline crew scheduling problem: A survey.* Transportation Science, 3(2), pp. 140-163.

[23] Wren, A. (1981). *A general review of the use of computers in scheduling buses and their crews.* In: Wren, A. (Eds.), Computer Scheduling of Public Transport, Urban Passenger Vehicle and Crew Scheduling, North-Holland, Amsterdam, pp. 3–16.

[24] Sitompul, D. and Radhawa, S. (1990). *Nurse scheduling: A state-of-the-art review.* Journal of the Society for Health Systems, 2(1), pp. 62-72.

[25] Grossman, T., Samuelson, D., Oh, S. and Rohleder, T. (1999). *Call centers.* Technical Report, Haskayne School of Business, University of Calgary.

[26] Tien, J., Kamiyama, A. (1982). *On manpower scheduling algorithms.* SIAM Review, 24(3), pp. 275-287.

[27] Ernst, A.T., Jiang, H., Krishnamoorthy, M. and Sier, D. (2004). *Staff scheduling and rostering: a review of applications, methods and models.* European Journal of Operational Research, 153(1), pp. 3-27.

[28] Ernst, A.T., Jiang, H., Krishnamoorthy, M., Owens, B., and Sier, D. (2004). *An annotated bibliography of personnel scheduling and rostering.* Annals of Operations Research, 127(1-4), pp. 21-144.

[29] Van den Bergh, J., Belieen, J., De Bruecker, P., Demeulemeester, E. and De Boeck, L. (2013). *Personnel scheduling: A literature review.* European Journal of Operational Research, 226(3), pp. 367-385.

[30] Dantzig, G. (1954). *A comment on Edie's traffic delay at toll booths.* Operations Research, 2(3), pp. 339-341.

[31] Teodorović, D. and Lucić, P. (1998). *A fuzzy set theory approach to the aircrew rostering.* Fuzzy Sets and Systems, 95(3), pp. 261-271.

[32] Andersen, E. D. and Andersen, K. D. (1995). *Presolving in linear programming.* Mathematical Programming, 71(2), pp. 221-245.

[33] Cornuéjols, G. (2008). *Valid inequalities for mixed integer linear programs* Mathematical Programming B, 112(1), pp. 3-44.

[34] Danna, E., Rothberg, E. and Le Pape, C. (2005). *Exploring relaxation induced neighborhoods to improve MIP solutions.* Mathematical Programming, 102(1), pp. 71-90.

[35] Berthold, T. (2006). *Primal Heuristics for Mixed Integer Programs.* Technische Universität Berlin.

[36] Mészáros C. and Suhl, U.H. (2003). *Advanced preprocessing techniques for linear and quadratic programming.* OR Spectrum, 25(4), pp. 575-595.

[37] Nemhauser, G.L. and Wolsey, L.A. (1999). *Integer and Combinatorial Optimization.* Wiley-Interscience, New York.

[38] Savelsbergh, M.W.P. (1994). *Preprocessing and Probing Techniques for Mixed Integer Programming Problems.* ORSA Journal on Computing, 6(4), pp. 445-454.

[39] Wolsey, L.A. (1998). *Integer Programming.* Wiley-Interscience, New York.

[40] Klemettilä, S., Haahtela, Y. and Salo, A. (2016) *Henkilöstön strateginen mitoitus kokonaislukuoptimoinnilla.* Master's thesis, School of Science, Aalto University, Espoo.

[41] Mathworks.com, (2018). *Documentation: intlinprog.* [online] Available at: https://se.mathworks.com/help/optim/ug/intlinprog.html [Accessed 6 Feb. 2018].

[42] Mathworks.com, (2018). *MathWorks Product Release Schedule.* [online] Available at: https://se.mathworks.com/products/new_products/release_model.html [Accessed 6 Feb. 2018].

A Forms for data collection

Työkohtainen kyselylomake

Työn tunnus tai nimi ______________
Sarjan tyypillinen koko (potilasnäytteet) ______________
Esikäsittelyyn kuluva aika* ______________
josta sarjan valmisteluun kuluva aika** ______________
Yksittäisen näytteen ajo kestää*** ______________

Työvaiheet jotka kuuluvat analyysiin
☐ hydrolyysi ☐ proteiinisaostus tai dilute & shoot
☐ 1. nesteuutto ☐ 2. nesteuutto
☐ kiinteäfaasiuutto
☐ derivatisointi ☐ ajoliuosten valmistaminen LC tai LC/MS
muita; mitkä

____

____

____

Käytetty analysointitekniikka
☐ LC ☐ LC/MS
☐ GC ☐ GC/MS
☐ muu; mikä______________________________

Voidaanko joitain työvaiheita tehdä jo edellisenä päivänä ja jos, niin mitä vaiheita?

____

____

____

Voiko mielestäsi työtä tai sen vaiheita automatisoida? Mitä vaiheita? Mitä se edellyttäisi?

____

____

____

Muita huomautuksia voi kirjoittaa paperin toiselle puolelle. (esim. jos samanaikaisesti tyypillisesti suoritetaan muita töitä – mitä töitä?)
Kiitos vastauksistasi!

* varsinaiset **työtunnit** työlistan tulostamisesta analysointiajon käynnistämiseen
** näytteiden haku, tarroittaminen ja pipetointi = toimenpiteet ennen varsinaisen esikäsittelyn alkamista
*** LC, LC/MS, GC, GC/MS tai muu vastaava analysointimenetelmä

Työn seurantalomake

Työn tunnus tai nimi ____________
Sarjan koko (potilasnäytteet) ____________
Työ alkoi ____________

Työvaihe	Alkoi	Loppui	Huomautuksia
☐ sarjan valmistelu			
☐ hydrolyysi			
☐ proteiinisaostus tai dilute & shoot			
☐ 1. nesteuutto			
☐ 2. nesteuutto			
☐ kiinteäfaasiuutto			
☐ derivatisointi			
☐ ajoliuosten valmistaminen			

Työ loppui ____________

Muita huomautuksia voi kirjoittaa paperin toiselle puolelle.
Kiitos vastauksistasi!

B MATLAB script manual

This manual comes attached to the master's thesis "Cost-efficient response time optimization of analytical measurement processes of a clinical laboratory - Case United Medix Laboratories Ltd" written by Michael Asplund in 2018.

The manual briefly explains how to use the MATLAB scripts written for the Integer Linear Programming simulations done in the thesis. It covers three different script-files written in the MATLAB m-code and it is recommended to use a version R2017b or newer of the MATLAB software by MathWorks to run them. As the script-files import model parameters from an external csv-file, the format of that file is also covered.

B.1 main_model_01.m

The file `main_model_01.m` like the name suggests is the main script-file which runs the whole simulation. This file starts by loading an external csv-file containing procedure parameters specific to a certain case one would like to model. The file parses the content of the csv-file and then enters a simulation loop. By modifying the number of loops it does one can choose how many different simulations of a particular case should be run in one go.

Inside the loop there are a number of additional parameters one can freely modify for the simulation. These are:

- The number of days simulated
- The normal working hours
- The extended working hours
- The resources

The changing the number of days one can affect the duration of the simulated time span.

The normal working hours have two parameters for each day; the hour at which they begin and the hour at which they end. During these hours the execution of assays is possible.

The extended working hours have three parameters for each day; the hour at which they begin, the hour at which they end and the amount of employees that these working hours should be further restricted to in addition to the overall personnel restriction (read further).

The resources restrict how many assays and measurements can be executed simultaneously. The T-parameter indicates the overall employee restriction. The rest of the parameters indicate different equipment restrictions. For instance, LCMS stands for Liquid Chromatography/Mass Spectrometry and specifies how many sample batches requiring such equipment can be executed simultaneously. By choosing a sufficiently high employee restriction one can make sure that the equipment resources are the bottleneck of the process or vice versa.

At the very end the file it calls the script-file `model_linear.m`, which does the actual simulation. When execution returns back the results are saved into a file in the folder which open in MATLAB.

B.2 model_linear.m

The file `model_linear.m` gets the parsed and additionally specified parameters from `main_model_01.m`, does some preprocessing and ends by running the optimization simulation.

First, the sample arrivals for each procedure and each day are randomized based on parameters from the csv-file. Based on these arrivals the preprocessing estimates the number of sample batches needed for each procedure in order to process all samples.

Secondly, the matrices and arrays needed as parameters for the simulation are generated. These are the constraints and the cost function of the Integer Linear Programming problem. Various optimization is done along the way to help maintain these possibly massive amounts of data as compact as possible.

Thirdly, it simulates the model for the time specified (one and a half hours used in the thesis).

Lastly it outputs the results on the screen before returning execution to the calling script-file.

B.3 result_comparison.m

After each looped simulation of the script-file `main_model_01.m`, the data is saved into an external file. The script-file `result_comparison.m` makes it possible to compare different groups of simulations (different cases) to each other. In the beginning of the file one specifies the names of the previously simulated and generated data-files one wishes to compare and the script then does a comparison based on procedure response times, working hours and salary costs.

There is also a small commented script at the end for comparing how the sample sizes of each simulated sample batch differs from the expected amounts imported and parsed from the csv-file. This script helps notice possible suboptimal outliers resulting from the approximation the model does in order to model response times linearly.

B.4 The csv-files

A csv-file imported by the main script-file `main_model_01.m` should contain all the procedures one wishes to simulate and parameters related to them. A correctly formatted csv-file contains the following columns:

- assay
- mean week

- variance week
- slope week
- mon-sun
- phase resources
- phase execution times
- expected amount

The assay-column is where the procedure short name goes.

The mean week and variance week values are meant to express the weekly variance in sample arrivals as the parameters of a normal distribution. By setting the variance to 0 one can make sure that the weekly sample arrivals of that particular procedure do not vary.

The slope week value is there in case on wishes to model a linear rise in the mean value per week. This can be used to model a procedure for which demand has been steadily rising.

The values mon-sun specify how the weekly sample arrivals are to be distributed along the week. The values work like percentages and are weighted in comparison to each other. This can be used if one knows that samples arrive more during certain weekdays. To divide all samples equally among the weekdays, set all values to be equal in a row.

The phase resources column expresses what resources the procedure uses and it has the format T-X where X is substituted by the automated measurement equipment, for instance LC for Liquid Chromatography.

The phase execution times specifies the execution times of the procedure parts. First comes the value indicating the execution time of the assay in hours, after which a dash follows and lastly comes the execution time of the measurement.

Expected amount specifies how many samples are expected to be processed simultaneously during a normal batch of that procedure. These values are used in estimating the response times.

C ILP-model abbreviations

C.1 Sets

P	The set of all procedures
H	The set of all hours
D	The set of all days
R_p	The set of all repetitions of procedure $p \in P$
$\mathcal{R}$	The set of all resources

C.2 Variables

x_{prt}^{assay}	Binary decision variable specifying whether the assay of procedure $p \in P$ and repetition $r \in R_p$ started at the beginning of hour $t \in H$
$x_{prt}^{measurement}$	Binary decision variable specifying whether the measurement of procedure $p \in P$ and repetition $r \in R_p$ started at the beginning of hour $t \in H$
q_{prd}	Integer variable specifying how many of the samples for procedure $p \in P$, that arrived on day $d \in D$ are processed during repetition $r \in R_p$

C.3 Parameters

ds_d	specifies the first hour of H during day $d \in D$
de_d	specifies the last hour of H during day $d \in D$
$s_p^{expected}$	the amount of samples expected to be processed per repetition of procedure $p \in P$
t_p^{assay}	execution time of the assay part of procedure $p \in P$ in hours
$t_p^{measurement}$	execution time of the measurement part of procedure $p \in P$ in hours
t^{delay}	a maximum waiting time after an assay part has finished until the measurement part must have been started
m_{large}	a sufficently large value to make sure that no samples are processed before they even arrive (see constraint 5)
s_{pd}	amount of samples that has arrived on day $d \in D$ for procedure $p \in P$
ρ_{pk}^{assay}	binary parameter specifying whether the assay part of procedure $p \in P$ utilizes resource $k \in \mathcal{R}$
$\rho_{pk}^{measurement}$	binary parameter specifying whether the measurement part of procedure $p \in P$ utilizes resource $k \in \mathcal{R}$
ρ_{kt}^{tot}	maximum amount of resource $k \in \mathcal{R}$ that can be in use simultaneously at hour $t \in H$