Czech Technical University in Prague

Faculty of Electrical Engineering

Department of Computer Science

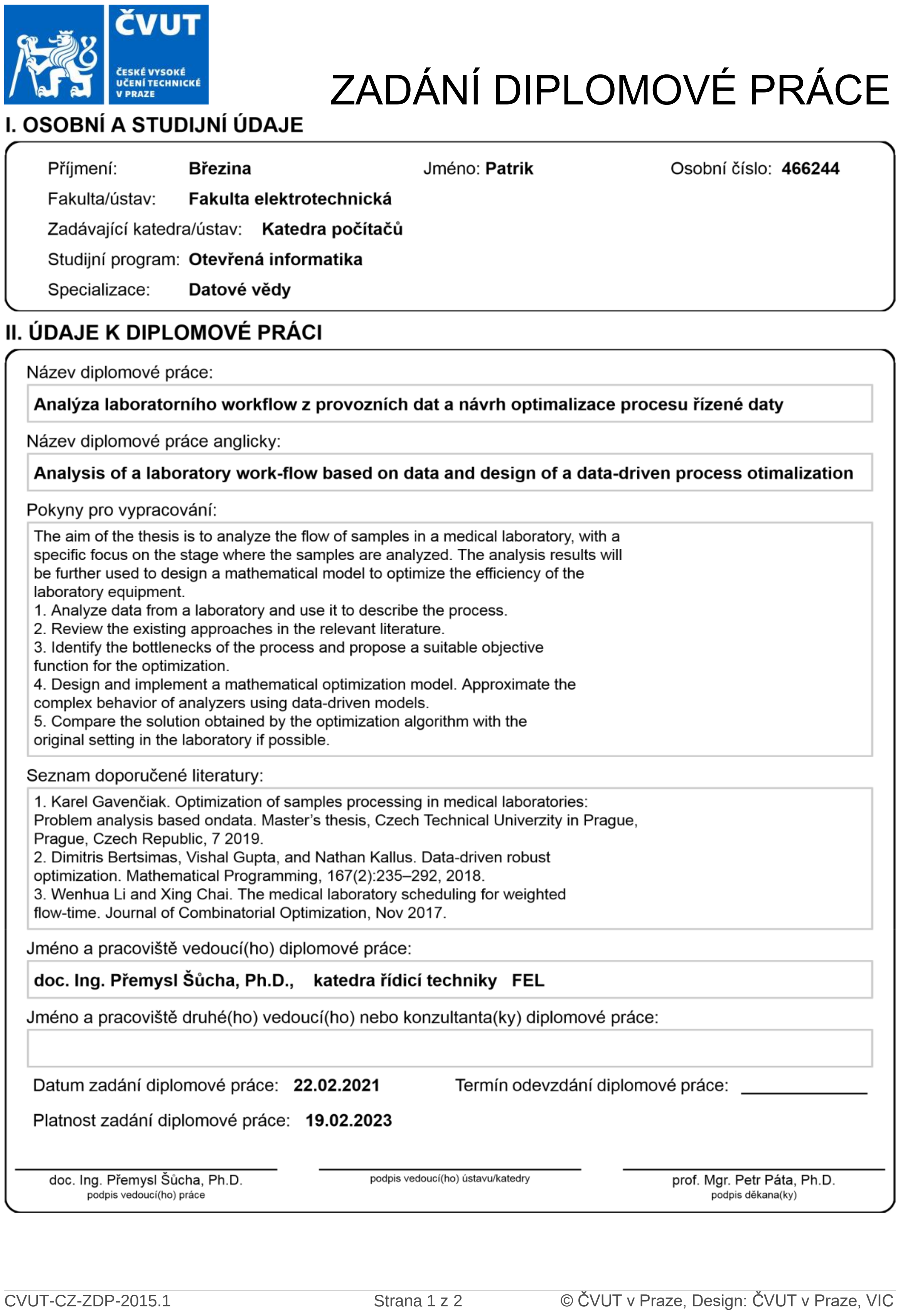
Master’s Thesis

Data Analysis of Laboratory Workflow and Data-driven Laboratory Process Optimization

Patrik Březina

Supervisor: doc. Ing. Přemysl Šůcha, Ph.D.

Data Analysis of Laboratory Workflow and Data-driven Laboratory Process Optimization



ČVUT
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Březina** Jméno: **Patrik** Osobní číslo: **466244**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra počítačů**

Studijní program: **Otevřená informatika**

Specializace: **Datové vědy**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Analýza laboratorního workflow z provozních dat a návrh optimalizace procesu řízené daty

Název diplomové práce anglicky:

Analysis of a laboratory work-flow based on data and design of a data-driven process otimalization

Pokyny pro vypracování:

The aim of the thesis is to analyze the flow of samples in a medical laboratory, with a specific focus on the stage where the samples are analyzed. The analysis results will be further used to design a mathematical model to optimize the efficiency of the laboratory equipment.

1. Analyze data from a laboratory and use it to describe the process.
2. Review the existing approaches in the relevant literature.
3. Identify the bottlenecks of the process and propose a suitable objective function for the optimization.
4. Design and implement a mathematical optimization model. Approximate the complex behavior of analyzers using data-driven models.
5. Compare the solution obtained by the optimization algorithm with the original setting in the laboratory if possible.

Seznam doporučené literatury:

1. Karel Gavenčiak. Optimization of samples processing in medical laboratories: Problem analysis based ondata. Master’s thesis, Czech Technical Univerzity in Prague, Prague, Czech Republic, 7 2019.
2. Dimitris Bertsimas, Vishal Gupta, and Nathan Kallus. Data-driven robust optimization. Mathematical Programming, 167(2):235–292, 2018.
3. Wenhua Li and Xing Chai. The medical laboratory scheduling for weighted flow-time. Journal of Combinatorial Optimization, Nov 2017.

Jméno a pracoviště vedoucí(ho) diplomové práce:

doc. Ing. Přemysl Šůcha, Ph.D., katedra řídicí techniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **22.02.2021** Termín odevzdání diplomové práce: ____________

Platnost zadání diplomové práce: **19.02.2023**

doc. Ing. Přemysl Šůcha, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

CVUT-CZ-ZDP-2015.1 Strana 1 z 2 © ČVUT v Praze, Design: ČVUT v Praze, VIC

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

.	
Datum převzetí zadání	Podpis studenta

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guidelines for adhering to ethical principles when elaborating an academic final thesis.

In Prague on May 17.5.2021 ___________________________

Acknowledgments

I would like to express my sincere gratitude to my supervisors Doc. Ing. Přemysl Šůcha, Ph.D. and Ing. Antonín Novák, for offering me to participate in this research and providing guidance and friendly working environment during its completion.

I would also like to thank the technicians of the Prevedig Laboratory, namely Petr Matějka and Boris Popsimov, for their assistance and explanations of the laboratory system. Without them, our research would likely not be as successful as it was.

Lastly, I would like to thank my family for all the provided support during my studies.

Abstrakt

Zdravotní laboratoře hrají kritickou roli v kvalitě zdravotní péče. Provádí analýzu organického materiálu odebraného z těla pacientů k určení jejich zdravotního stavu. Včasná léčba může zachytit nemoc již během její inkubace a zabránit tak většímu dopadu. Toto může být rozhodující faktor mezi životem a smrtí pacienta. Proto je prvořadé optimalizovat konfiguraci těchto systému, aby byly výsledky analýzy vydány co nejdříve. Bohužel, téma laboratorní optimalizace není v současné době dostatečně prozkoumáno. Naším cílem je vyplnit tuto mezeru ve výzkumu. Spolupracovali jsme se soukromou laboratoří Prevedig Medical, která nám poskytla provozní data a vhled do jejich automatizace. Analyzovali jsme jejich provozní data a navrhli optimalizační algoritmy zlepšující jejich automatizaci. V optimalizaci jsme se soustředili na průchodnost biochemických analyzátorů, kde jsme dosáhli 33.10% zlepšení. Navíc jsme navrhli přístup optimalizující nastavení laboratoře vzhledem ke specifickým požadavků laboratoře s cílem minimalizace průměrného TAT vzorků.

Klíčová Slova:

Laboratorní Automatizace

Průchodnost Automatizace

Turn-around Time (TAT)

Datově Řízená Optimalizace

Celočíselné Lineární Programování

Evoluční Algorithmus

Data Analysis of Laboratory Workflow and Data-driven Laboratory Process Optimization

Abstract

Medical laboratories play a vital role in healthcare quality. They perform analysis of organic material sampled from a patient’s body to asses the patient’s health status. Timely medical intervention can intercept the incubation of an illness before it causes harm. This can be a deciding factor between the life and death of the patient. Therefore, it is paramount to optimize the configuration of these systems to deliver analysis results as soon as possible. Unfortunately, the area of laboratory optimization is currently not thoroughly examined. We aim to fill in this research gap. We collaborated with laboratory Prevedig Medical who provided us operational data and insight into their automatization. We analyzed the operational data and devised optimization approaches to improve the automatization system. In the optimization, we focused on the throughput of the biochemical analyzers, increasing their throughput by 33.10%. Furthermore, we designed another approach that optimizes laboratory configuration, minimizing the average sample TAT while respecting the laboratory requirements.

Keywords:
Laboratory Automatization
Automatization Throughput
Turn-around Time (TAT)
Data-Driven Optimization
Integer Linear Programming
Evolutionary Algorithm

Contents

List of Figures

List of Tables

1 Introduction

Laboratories are one of the essential participants in the healthcare sector. They provide services in the form of laboratory tests, analyzing various types of samples to assess the patient's condition and determine the appropriate treatment. A laboratory test is a procedure where a sample from a patient's body, such as blood, urine or plasma, is examined with respect to some criterion like the presence of an antigen or the amount of a specific substance in the sample. In laboratories, expensive equipment is operated, and even small equipment usage inefficiencies may significantly impact the total laboratory operational cost. However, in a hospital environment where any delay in the appropriate treatment of a patient can endanger their life, the most critical is the amount of time needed to evaluate the tests while maintaining high accuracy. Therefore, for most laboratories and physicians, the essential quality indicator of laboratory services is turn-around time (TAT).

In general, turn-around time is the time interval between 2 arbitrary milestones in a process. In the case of laboratories, the two milestones are usually reception of a sample by the laboratory and results reporting of all the tests requested by a physician. For most laboratories, the goal is to minimize their turn-around time. One possibility to achieve this is to expand the laboratory system and adding additional components to its automatization. However, adding new equipment would result in higher installation and maintenance costs of the system, which might not always be possible. Another option is to use equipment that is currently available in the laboratory efficiently. This work focuses on the second problem regarding efficiency.

We worked together with the private laboratory Prevedig Medical [14] and their team of technicians. They provided us the operational data of their laboratory and insight into the functionality of the installed automatization system DxA5000. The core difference between a hospital and a private laboratory is that in a private laboratory, the system does not work with high emergency samples where untimely analysis of the sample could endanger a patient's life. Therefore, the TAT criterion is not of primary concern for these laboratories, although they still maintain sample TAT guarantees as a quality measure of their services. Instead, effective utilization of equipment or high system throughput is more important. With Prevedig's assistance, we designed two optimization approaches that find an optimal assignment of methods to laboratory analyzers given the laboratory operational data. The proposed algorithms exploit the information hidden in the data, such as the correlation between methods or distribution of requested tests in time.

The first approach focuses on the assignment of biochemical tests to the biochemical analyzers in Prevedig's automatization, balancing the workload among the considered components and maximizing the total biochemistry throughput. However, based on a single parameter's value, the algorithm's focus can be changed to the TAT of individual samples while maintaining as high throughput as possible, making the algorithm applicable in private and hospital laboratories.

The second approach is designed to optimize average sample TAT in a hospital environment. It utilizes operational data of a hospital laboratory provided by Faculty Hospital Královské Vinohrady applied to Prevedig's automatization. The algorithm considers the TAT of individual samples w.r.t. various processes occurring at the biochemical and immunological analyzers. For hospital laboratories, it is common to impose TAT requirements on samples based on their priority. The optimization model finds a system configuration that minimizes the average sample TAT while satisfying the laboratory requirements.

1.1 Related work

The research regarding the optimization of laboratory processes is currently not thoroughly explored. However, in their review of management techniques designed to improve hospital laboratories' performance [4], Leaven stresses the importance of such research to achieve high-quality healthcare and possibly reduce the total healthcare cost.

Many factors influence these processes, and only a small fraction of these factors have been considered in existing work. We separated the existing literature into several sections. The first Section 1.1.1 describes the quality indicators used to measure the efficiency of laboratory workflow and relations between these indicators. Section 1.1.2 contains research based on a statistical analysis of laboratory operational data. Literature in the following Section 1.1.3 focuses on the optimization of laboratory automatization. Section 1.1.4 introduces approaches optimizing laboratory workflow considering the available laboratory personnel. The final Section 1.1.5 describes methods improving the laboratory efficiency by optimizing factors that influence the laboratory but are outside the laboratory itself, such as routes of couriers who transport samples from physicians to the laboratory.

1.1.1 Laboratory Quality Indicators

To help researchers determine the quality of laboratory services, Tsai et al. created an extensive review [5] of quality indicators (QIs) in laboratory production processes. The authors confirm that turn-around time (TAT) is the most common laboratory performance criterion as it is of primary interest to physicians and clinicians who utilize laboratory services. However, they also suggest using other measurements used in production processes such as work in process, resource utilization, system throughput, or material handling cost. The reasoning behind this suggestion is that a laboratory is essentially a make-to-order production process where products are produced only if a customer places an order for them. In the case of laboratories, customers are physicians or clinicians who provide samples, and products are results of the ordered tests conducted on these samples. When selecting meaningful quality indicators for research, it is crucial to keep in mind that different QIs might correlate with each other, leading to inevitable trade-offs. The authors of the review acknowledged this possibility and constructed a diagram illustrated in Figure 1.1, visualizing possible correlations between the quality indicators considered in their work. Turn-around time (TAT) stands in the center of the image as the essential criterion.

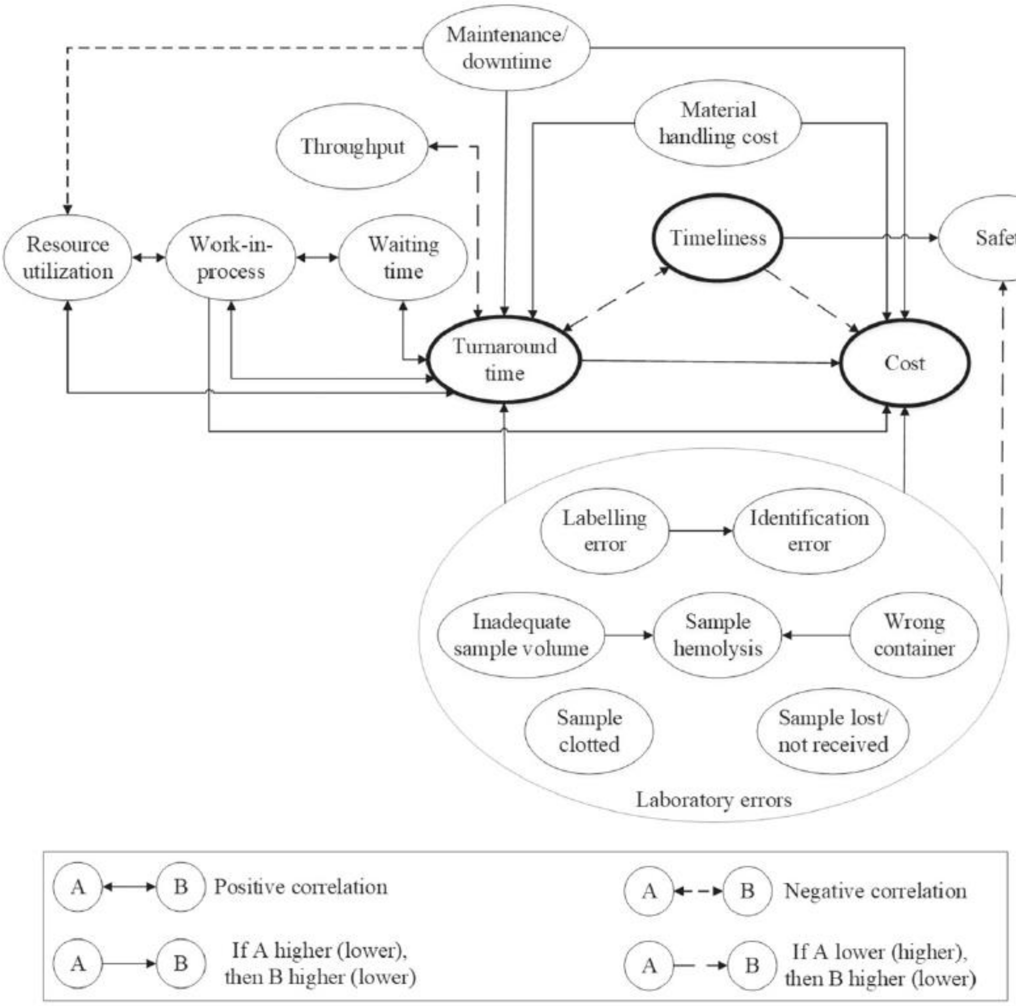


Figure 1.1 – Relation between quality indicators [5]

To achieve improvements in TAT, one can exploit relations between TAT and other quality indicators. For example, throughput negatively correlates with turn-around time. Thus, increasing system throughput would result in lower TAT. Similarly, reducing the amount of work-in-process, the average sample waiting time, or resource utilization in the system will result in TAT reduction. The positive correlation between resource utilization and turn-around time might be surprising. It is desirable to have high machine utilization in production processes, reflecting effective usage of the available resources. This is also true for laboratories, but high utilization is possible only with an increased workload. Higher workload will inevitably lead to increased waiting times of individual samples at analyzers, resulting in redundant delays of individual tests and increasing total sample TAT. This means that balanced workload distribution is more critical for laboratories than high resource utilization. Therefore, it is worthwhile for researchers to examine possible improvements in other quality indicators or find new meaningful laboratory performance measurements and their relation to turn-around time to achieve a decrease in laboratory TAT.

1.1.2 Statistical Analysis of Laboratory Data

Several papers performed statistical data analysis to identify possible redundancies and improvements in laboratory systems. It is common for laboratories to repeat tests of samples with positive results to minimize the possibility of erroneous analysis. However, this may unnecessarily increase the TAT of the production process and lead to the sample's higher processing cost. As was suggested by paper [6], for 99.3% of repeated tests, there was little difference between the result of the original test and the repeated one. This percentage is an average over all observed types of tests. The rate varies slightly depending on the testing method (calcium tests had the highest difference percentage with 4.9% different results). This result suggests that repetition of some tests is unnecessary and may be excluded from the process to achieve lower processing cost and sample TAT of the laboratory. As another example, Chien et al. analyzed in their work [7] the 90th percentile of analytical TAT of laboratory system and discovered that troponin-I test caused most TAT prolongations. Based on this observation, they experimentally prioritized troponin-I tests, resulting in 18 minutes reduction of TAT from 66 minutes to 48 minutes.

1.1.3 Optimization of Laboratory Automatization

The application of mathematical models to the laboratory environment is not currently thoroughly examined. However, the few existing works on the topic yield promising results in the optimization of laboratory automatization. In the paper [8], Yang et al. utilized the value stream mapping visualization method to identify limitations and bottlenecks of hospital laboratory in the medical center in southern Taiwan. The visualization uncovered surprisingly high waiting times at the three identical DXC analyzers installed in the automatization during peak demand caused by an uneven distribution of tubes among the analyzers. They proposed an approach that controls batch waiting time of centrifuge (how long the centrifuge waits for a whole batch) and distributes samples among the DXC analyzers. Tubes are separated based on their estimated processing time into 20 groups, each associated with an average processing time. These groups are distributed among the DXC analyzers based on their averages with respect to two defined value breakpoints. An integer linear program (ILP) is utilized to find optimal values for the two breakpoints and the centrifuge batch waiting time to maximize the number of samples with TAT below 60 minutes, achieving a significant 54.51% improvement of average sample TAT from 7320 seconds to 4222 seconds.

A different approach is suggested in [9], where authors perceive a patient's sample as a job waiting to be scheduled and analyzers as workers performing these jobs. Jobs release over time, and the information about their release time and priority is not available until their arrival. The authors focus their attention on biochemical analyzers. Thus all the jobs' processing time is the same (this is a property of some biochemical analyzers). They modeled the laboratory automatization as a parallel-batch online scheduling problem with an unbounded batch capacity and different job priorities based on the sample importance with the objective to minimize the maximum weighted flow time, maximizing utilization of biochemical analyzers, and improving the efficiency of sample processing.

1.1.4 Optimization Considering Laboratory Personnel

Optimization of laboratory performance is not limited only to its automatization. For instance, Boyd and Savory designed a genetic algorithm to set up laboratory personnel's task schedules for given work shifts, distributing individuals among workplaces and tasks based on their abilities and availability [11]. Their algorithm's fitness function is based on the priorities of individual tasks and weighted skills of laboratorians. The goal is to maximize the sum of the tasks weighted by the assigned individual's skill for the given shift. Therefore, the algorithm prioritizes the assignment of skilled individuals to high-priority tasks.

Laboratory personnel was considered from a different perspective by Leeftink et al. in their paper [13]. They devised an integer linear program (ILP) to optimally organize the histopathology laboratory's workload of University Medical Center Utrecht considering the laboratory operational data. The authors collected data over 12 months, from 1 January 2013 to 31 December 2013. The data analysis uncovered that in early hours between 8:00 and 9:30, the amount of workload exceeded the available workforce. An ILP model was designed to approximate the histopathology processes and evaluate them in terms of TAT and workload spread. The operational data was used to schedule specimen arrival to the histopathology laboratory with respect to the currently available personnel. As a result, the model achieved better workload leveling and accomplished a 25% decrease of TAT in early hours.

1.1.5 Optimization Outside the Laboratory Environment

The laboratory's performance can be influenced by any event in the total testing process (TTP) depicted in Figure 1.2. TTP describes individual phases and milestones in laboratory workflow, starting with patient diagnosis in the pre-analytic phase and ending with potential treatment in the post-analytic stage. It is meaningful to examine possible improvements in these events to improve the laboratory workflow, as demonstrated in works [10] and [3].

The authors of [10] focus on transportation in the total testing process, optimizing courier routes to improve sample transport efficiency and improving the laboratory's workload balance. The authors

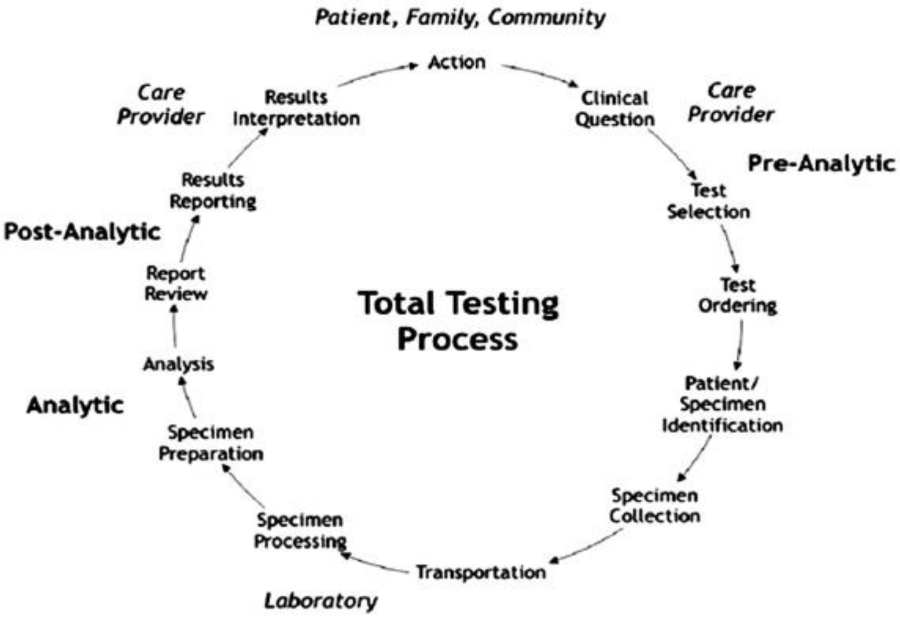


Figure 1.2 - Total Testing Process visualization [12]

constructed a simulation model to simulate couriers' movement and utilized it to optimize their routes. They achieved a nearly uniform workload balance in morning hours, reduced the amount of work during peak demand periods, and redistributed this work among hours with a lower workload. As a result, they achieved better utilization balance while avoiding the laboratory's potential overload.

Based on the literature reviews regarding outpatient appointment scheduling problem in hospitals [1-2], a problem where a patient needs to sequentially visit multiple types of resources available in the hospital to receive their treatment or diagnosis, we can observe an increasing interest in this topic in recent years. However, Ahmadi-Javid et al. noted in their review [2] that most of these studies neglect environmental factors and encourages researchers to study these factors as it would help to increase the applicability of proposed approaches in practice. An example of environmental factors is patient unpunctuality (time difference between patient's appointment and their arrival), random service times (time needed to diagnose the patient), or medical check-up interruptions (such as the arrival of an emergency patient).

An example of research regarding outpatient scheduling problem is a paper made by Suleyman Sevinc et al. [3]. They designed a two-phased heuristic approach to arrange a chemotherapy appointment schedule. The first phase solves patients' scheduling for laboratory tests, determining whether a patient needs to undergo the infusion or not. The second phase concerns the distribution of patients among the infusion seats. The first phase utilizes a negative feedback scheme to balance the laboratory's load around a pre-set utilization value over a set of time windows. For each time window, the laboratory workload is used to estimate laboratory utilization. The goal is to arrange an appointment schedule where the laboratory's utilization is at least as high as the targeted utilization for each time window. Therefore, the selection of appropriate value for the targeted utilization value is paramount. The second phase is formulated as a Multiple Knapsack Problem, where a set of patients is being distributed among multiple infusion seats to maximize their utilization. The first phase of this work demonstrates the possible influence of outpatient appointment scheduling on laboratory efficiency, where different appointment schedules result in various laboratory utilization among the time windows.

1.2 Contribution

This thesis aims to fill in the research gap regarding the optimization of laboratory automatization. Currently exists only a few studies that focus their attention on laboratory automatization. Therefore, it is difficult to find guidelines for its optimization. We provide ideas on how to achieve these improvements using the equipment currently available in the laboratory. We have performed data analysis of the laboratory operational data and identified opportunities for enhancements. According to our findings in the data analysis, we devised optimization approaches that are also applicable to other laboratory systems. Moreover, we thoroughly discuss considered constraints and relations between utilized criteria to assist other researchers in finding an appropriate approach to optimizing laboratory automatization.

1.3 Outline

In the remainder of the thesis, we first introduce the considered laboratory automatization, its characteristics, and the operational data in Section 2. Afterward, in Section 3, we performed an analysis of the operational data, identifying potential bottlenecks that are further studied in the following sections. Section 4 defines the first optimization task considered in this work regarding the maximization of laboratory system throughput and introduces two approaches to solve the problem. In Section 5, we describe the second optimization task minimizing average sample TAT. Results of all the performed experiments with the proposed algorithms are available in Section 6. The final Section 7 the research conducted in this work and provides some ideas for future research.

2 Laboratory

In this section, we discuss the background of our research. Firstly, we introduce the private laboratory Prevedig Medical and outline the differences between private and hospital laboratories in Section 2.1. In Section 2.2, we described the laboratory workflow and visualized it in a flow chart. Afterward, we explained the installed laboratory automatization in Section 2.3, emphasizing biochemical analyzers in Subsection 2.3.1 and immunology in Subsection 2.3.2. The final Section 2.4 discusses the operational data utilized in the following research, namely data formatting and information contained within.

2.1 Medical Laboratory Prevedig

Prevedig Medical is a Czech private laboratory founded in 1993 in Prague. The laboratory offers many biochemical, immunological, hematological, serological, or microbiological examinations of an organic sample. The samples are collected by clinicians or by laboratory personnel. For urgent cases, the laboratory offers sample testing with two different priorities. Routine samples have low priority, whereas Statims are high priority. Statim priority is reserved for samples of patients whose condition has abruptly changed. The laboratory guarantees that routine samples' results will be reported during the day they were delivered. On the other hand, statim samples are analyzed as soon as possible after the sample's reception.

The sample priorities are the core difference between private and hospital laboratories. Hospital laboratories are attached to a hospital or a medical center and work with three different sample priorities – Routine, Statim, and Vital. Routine and Statim have the same role as in a private laboratory. However, Vital priority is reserved for situations where timely medical intervention can save a patient's life. An example of such a critical situation is a patient who has suffered a heart attack. Such high-priority samples are not analyzed in private laboratories because they would need to be transported first. This transportation delay could be fatal to the patient. Therefore, the private laboratories' testing process is not as time-restricting, and the laboratory can focus on different efficiency indicators than turn-around time (TAT).

The Prevedig laboratory team provided us the operational data of their automatization and insight into their analyzers' complex processes. Their assistance helped us to approximate their system's behavior and design approaches that further improve the automatization capability considering the provided operational data.

2.2 Prevedig Laboratory Workflow

The processing of a single sample begins with the sample's collection from a patient. The sample is registered via an electronic order in the laboratory information system and is sent to the laboratory. Each sample is associated with several methods requested by the clinician. Some of the methods offered by the laboratory can be ordered with a statim priority, and the presence of a statim priority test determines the sample's overall priority. However, the amount of arrived samples is not uniformly distributed throughout the day. Clinicians typically draw blood samples from patients in the morning hours, and couriers then transport them to the laboratory. Thus, most samples arrive between hours 11 and 13, depending on how far the clinician is from the laboratory. This is illustrated in Figure 2.1, which shows the average number of samples that arrived during respective hours. For example, hour 11 stands for the time interval between 11:00 and 11:59, and the y-axis value describes the number of samples that arrived in the laboratory during this time interval, which is nearly 200 samples for hour 11. The blue vertical lines visualize standard deviations of the samples arrival number. It is clear that workload is unevenly balanced throughout the day and varies between individual days. This may be somewhat alleviated by optimizing courier paths, as was demonstrated in the paper [10], but this is beyond this thesis's scope.

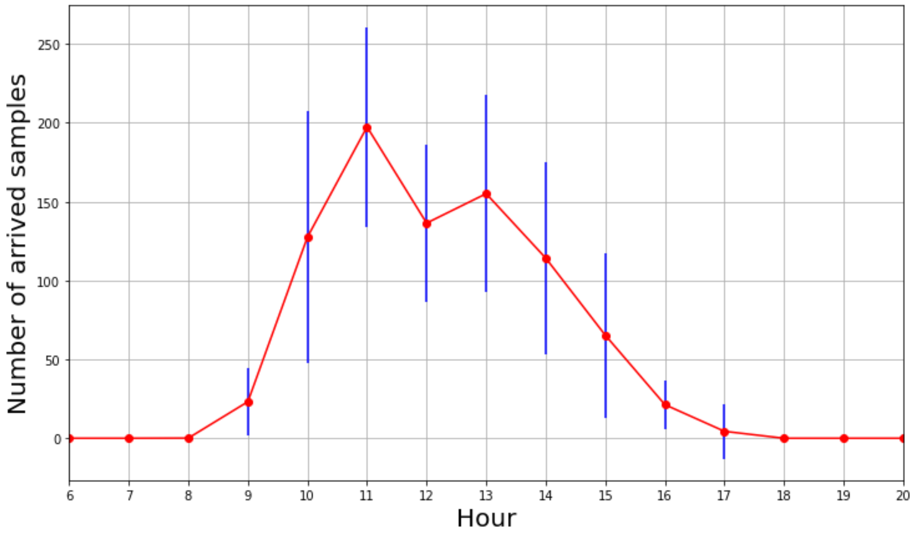


Figure 2.1 – The average number of arrived samples in individual hours

After the laboratory receives the sample, the laboratory personnel inserts it into the automatization input. Each sample is identified by a unique label attached to the tube. These labels are scanned at the input, and the information system is queried for methods associated with this sample. Based on the requested methods, the system plans the sample's route, determining analyzers the sample should visit. Afterward, it is sent to the centrifuge. Centrifugation separates samples into individual particles with centrifugal force. For instance, a blood sample is fractioned into three layers - blood plasma, the layer of platelets mixed with leukocytes, and the red blood cells. After centrifugation, the cap is removed from the tube containing the sample, and it is ready to be analyzed.

All automatization components, including input, centrifuge, analyzers, and sample storage, are connected by a transportation track. After centrifugation, the sample is placed on the transportation track. The track transports it to individual analyzers according to the sample's planned route, where the requested methods are performed. A method is a procedure that analyzes samples according to certain criteria, such as the presence of a specific substance, the amount of a substance in the sample, or the sample's reaction to artificial antibodies. Analyzers collect aliquots from the testing tube via a pipetting needle attached to a robotic arm and place them into several reaction slots to perform an analysis. Afterward, reagents are added to reaction slots depending on the test to be performed, initializing a sample's chemical response. The duration of the reaction varies among individual tests. For example, all biochemical reactions last about 10 minutes, but in immunology, the reaction time ranges from 15 minutes up to 75 minutes, depending on the method. After the reaction's completion, the analyzer reports the results to the information system.

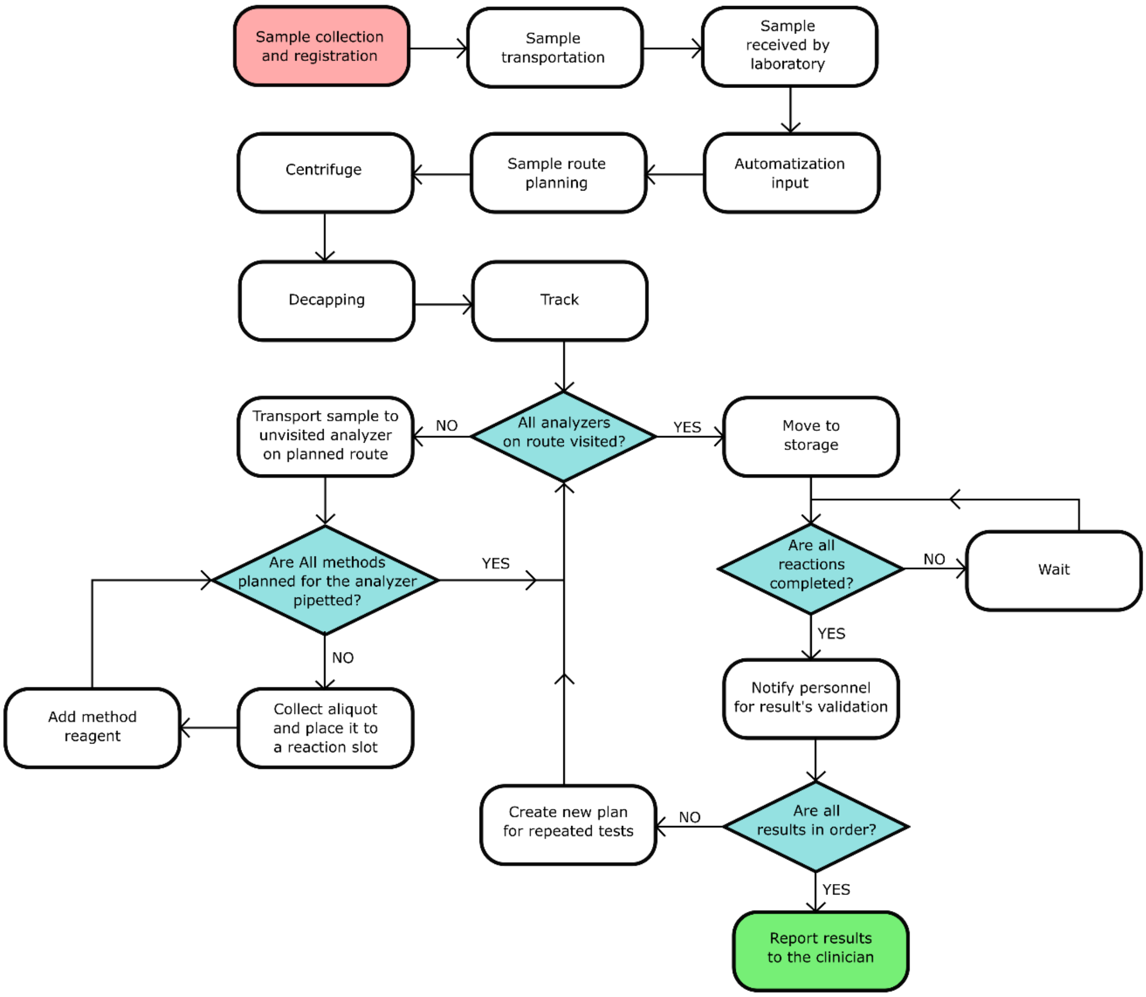

Sample collection and registration
Sample transportation
Sample received by laboratory
Automatization input
Sample route planning
Centrifuge
Decapping
Track
All analyzers on route visited?
NO
Transport sample to unvisited analyzer on planned route
YES
Move to storage
Are All methods planned for the analyzer pipetted?
YES
NO
Collect aliquot and place it to a reaction slot
Add method reagent
Are all reactions completed?
NO
Wait
YES
Notify personnel for result's validation
Are all results in order?
NO
Create new plan for repeated tests
YES
Report results to the clinician

Figure 2.2 – Flowchart of single sample's processing

Once all requested methods have been finished, the sample is transported to the storage. A laboratory employee validates that sample's results and may order reevaluation of particular methods. After all the results have been successfully validated, they are reported to the clinician who ordered the sample's analysis. Figure 2.2 depicts a flowchart showcasing described process on a single sample. The rectangles represent individual events. The red event marks the beginning of the process, whereas the rectangle highlighted in green represents the process's end. The blue rhombi stand for question events, shifting the flow throughout the diagram depending on the answer.

2.3 Prevedig Automatization

Laboratory Prevedig uses the DxA5000 automatization system designed and manufactured by the company Beckman Coulter. The automatization system provides various functionalities, notably a route planner, a sample scheduler, and a data manager. The route planner calculates the most efficient route for individual samples considering the analyzers' configurations. The priority-based sample scheduler regulates the amount of work in the process to avoid overload of the analyzers, and the data manager archives various events of the laboratory workflow. However, the exact decision process of the planner and the scheduler is unknown to us. The Prevedig laboratory installation is depicted in Figure 2.3.

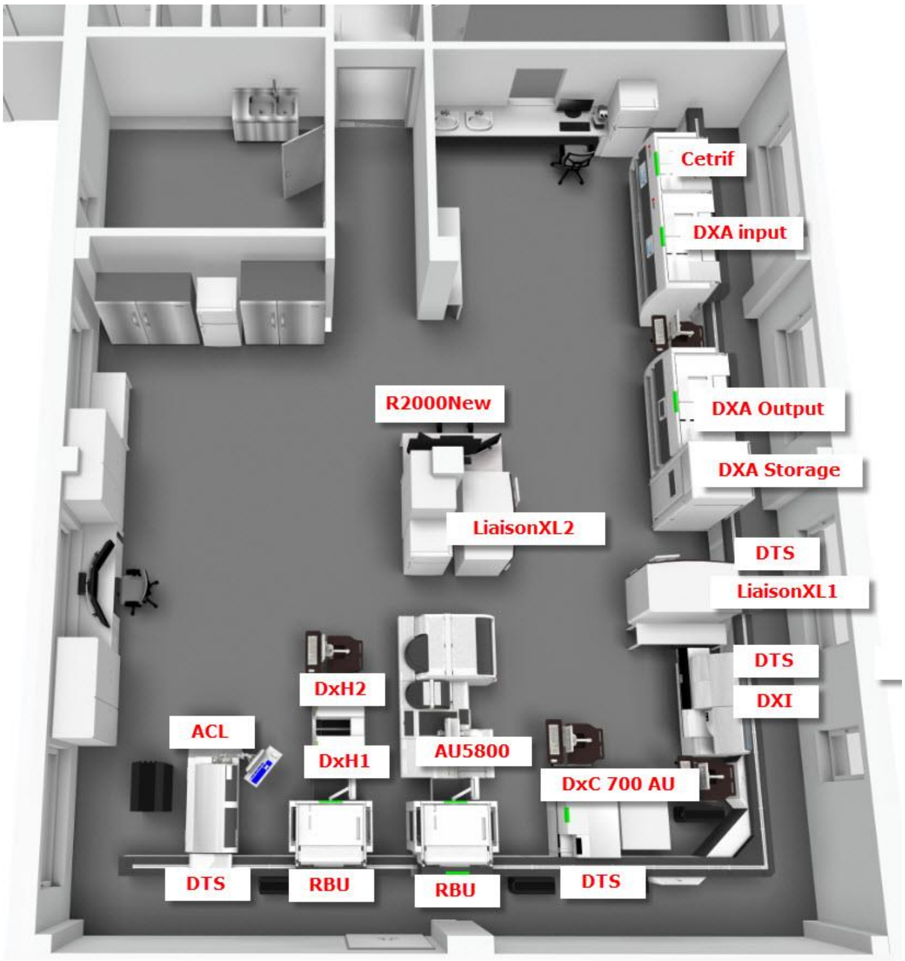


Figure 2.3 - Prevedig laboratory automation system

Individual samples are inserted into the system via DXA input, where the sample is identified by its label and is linked with its order recorded in the information system. Here, the scheduler decides which available samples will be dispatched for analysis, and the route planner calculates the samples' routes. The selected samples are moved to a batch where they wait for their centrifugation. The centrifuge (Centrif in Figure 2.3) operates in batches. Therefore, the samples wait until all the slots are occupied, or a certain timeout is reached before they are admitted to the centrifuge.

The automatization system consists of various types of analyzers, namely one coagulation analyzer (ACL), two hematology analyzers (DxH1 and DxH2), two biochemical analyzers (AU5800 and DXC 700 AU), and two immunological analyzers (LiaisonXL1 and DXI). The third immunological analyzer (LiaisonXL2) is not connected to the laboratory automatization and is operated manually by the laboratory personnel. Therefore, it is not of interest to our research. For simplification, we assume a second DXI analyzer instead of the LiaisonXL1. We may address the complete installation in future research.

This thesis concentrates on biochemical and immunological analyzers. The reason is these two types of analysis are the most time-consuming. That is especially true for immunology, where a reaction may last between 15 to 75 minutes. Moreover, the majority of samples request biochemical or immunological analysis, as is visualized in Figure 2.4. The figure shows the percentages of all the samples that ordered at least one analysis of a specific type. In Biochemistry, this means that 52.22% of samples requested at least one biochemical test. Naturally, a sample may request more than one type of analysis.

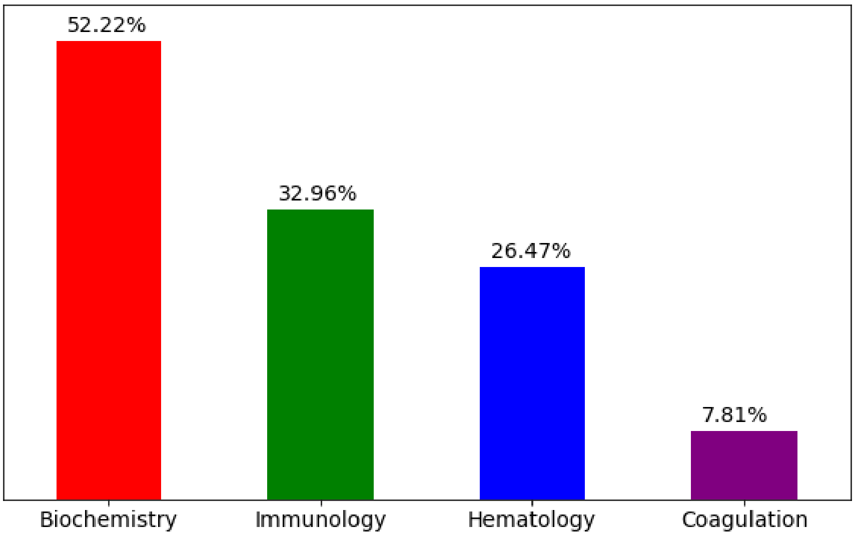


Figure 2.4 - Percentages of samples requesting a certain type of analysis

2.3.1 Biochemistry

The analyzers AU5800 and DXC 700 AU are the biochemical units installed in the Prevedig's automation system. We will refer to the AU5800 analyzer as the AU and the DXC 700 AU analyzer as the DXC.

AU is larger than the DXC analyzer, with quicker sample processing. The analyzer connects to the transportation track via the RBU unit. This unit collects whole tubes from the track and stores them in batches, each with ten samples. Once a batch is filled or a certain amount of time has passed, the batch moves to the AU analyzer. First, it visits a specialized unit designed for ISE tests to measure the samples' ion concentration. Afterward, the batch moves to the primary component that performs individual biochemical tests. This component contains two reaction carousels (inner and outer) with an equal number of slots, each accompanied by a pipetting arm. The analyzer processes samples one by one depending on their ordering in the batch. A pipetting arm collects aliquots from the samples for each requested biochemical method, stores them in a reaction carousel's slot, adds chemical reagent, and is washed afterward. This process lasts 3.6 seconds for individual arms. We will refer to this speed as the component's pipetting cycle. The arms take turns in aliquot pipetting, resulting in a theoretical speed of 1 test per 1.8 seconds. Thus, the AU's biochemical component can perform up to 2000 tests per hour. After pipetting is completed, the batch moves back to the RBU unit, where it waits until the results of all contained samples are known. All biochemical tests have 10 minutes reaction time on both biochemical analyzers (AU and DXC). The system automatically validates the results without the need for laboratory personnel. All samples with at least one invalid result are transferred to a different input batch. The RBU unloads the other samples to the transportation track.

The DXC analyzer precedes AU in the automatization. Unlike AU, this analyzer collects sample aliquots directly from the transportation track. The analyzer has one reaction carousel with one pipetting arm, and its pipetting cycle lasts 4.5 seconds. Therefore, the analyzer is able to execute up to 800 tests per hour. The functionality of the pipetting arm is identical to the AU's pipettors. Once the reaction of all the requested methods is initiated, the sample leaves the analyzer.

2.3.2 Immunology

In Prevedig's automation system, immunology tests are performed by two analyzers: LiaisonXL1 and DXI. However, we assume the automatization contains two identical DXI analyzers (DXI1 and DXI2, respectively) for simplification. Similar to the DXC analyzer, DXI collects sample aliquots directly from the transportation track. A single robotic arm collects aliquots with 9 seconds long pipetting cycle. However, unlike DXC, DXI collects a single aliquot of a greater volume instead of pipetting each requested method individually from the track. This aliquot is stored inside the analyzer, and several internal pipetting arms divide the aliquot into heated reaction slots. Then, a reagent is added to the reaction slot, causing a reaction from the pipetted sample. Unlike biochemical tests, the duration of immunology varies among individual methods, ranging from 15 minutes to 75 minutes. This makes the immunological tests the most time-consuming compared to the other method types, even in the case of methods with the shortest incubation.

2.4 Laboratory Operational data

The operational data provided by Prevedig has a vital role in this thesis. We used it to accurately approximate the automatization behavior, allowing us to measure the quality of a laboratory configuration quantitatively. Then we designed data-driven optimization techniques, finding the optimal system configurations based on information hidden in the data, such as the time spectrum of samples' arrivals or correlation between methods.

The DxA5000 data manager records the data. The data is archived in tabular form, where every row represents a single event in the automatization. The system identifies over 35 distinct event types, including sample input, route planning, the start of pipetting at an analyzer, end of pipetting, or results reporting. However, despite the wide variety of events, some are not recorded in the data, such as the arrival of a sample at a particular analyzer. Each archived event is accompanied by:

- **Sample identifier** – Identifies the sample that has caused the event.
- **Timestamp** – Time at which the event has occurred.
- **Event Type** – What kind of event has occurred.
- **Component** – On which automatization component the event has occurred.
- **Parameter** – Optional attribute, usually the name of a method. It is used especially with the "new result" event to identify the method for which the result has been reported.
- **Result** – Optional attribute containing the result of a method. The attribute is used exclusively with the "new result" event.
- **Comment** – Optional attribute reserved for additional information about the event. For example, the sample's route calculated in the "route plan" event.

Table 2.1 displays the structure of the data. However, note that this is a simplified artificial example to demonstrate the data formatting and information within it and not an actual record.

ID	Timestamp	Event Type	Component	Parameter	Result	Comment
12345S	08.04.2000 14:14:08	Route Plan	DXA	-	-	DXC – S_UREA, LIH; AU – BILK, S_GLU
12345S	08.04.2000 14:25:15	Pipetting Started	DXC	-	-	-
12345S	08.04.2000 14:25:24	Pipetting Finished	DXC	-	-	-
12345S	08.04.2000 14:36:56	New Result	DXC	LIH	0.0	-
12345S	08.04.2000 14:36:56	New Result	DXC	S_UREA	4.22	-

Table 2.1 - Data example

3 Operational Data Analysis

In this section, we discuss two significant findings we have uncovered with the analysis of operational data. The first Subsection 3.1 is dedicated to the routing of samples that need to visit both biochemical analyzers. The second Subsection 3.2 introduces the uneven workload of biochemical components that we address in our optimization algorithms in the later section.

3.1 Biochemistry Routing

The issue of biochemistry routing is related to the samples that need to visit both the biochemical analyzers, namely the DXC and the AU. We have analyzed the time required to finish the biochemical analysis of such samples. The results of this analysis are visualized in Figure 3.1. The figure depicts a histogram where the y-axis stands for the duration of biochemical testing in minutes. The duration is measured as the time between the sample's arrival at the first biochemical analyzer and the timestamp when all results are reported. The x-axis describes the number of samples whose analysis took the corresponding amount of time. For example, biochemical analysis required 40 minutes for 88 samples. The average biochemical processing time of samples is 39.5 minutes with a 7.60 standard deviation. However, the samples are organized into two clusters – one larger cluster containing 859 samples with durations ranging from 33 to 53 minutes, and a second smaller cluster of 67 samples with analysis time between 13 and 25 minutes. The difference between these two clusters is the order in which the samples visited the individual biochemical analyzers. The smaller cluster contains samples that have visited the analyzers in DXC→AU order, whereas in the larger cluster, the order is reversed to AU→DXC.

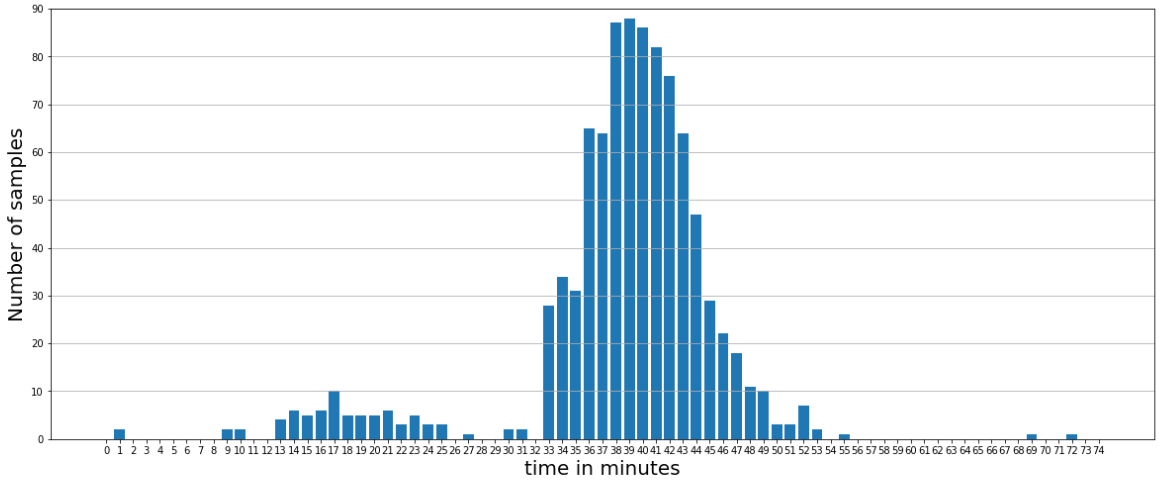


Figure 3.1 - Biochemical analysis duration histogram of samples visiting both analyzers

This significant difference between the clusters is caused by the distinction in sample processing of the individual analyzers. The DXC analyzer collects aliquots directly from the transportation track, and once the sample pipetting is finished, the tube leaves the analyzer immediately. On the other hand, AU removes whole tubes from the transportation track, and the tube is not released until all the requested tests have been finished. In DXC→AU ordering, the sample is pipetted at the DXC analyzer first, and the corresponding chemical reactions are initiated. Then the tube moves to the AU analyzer, where it is removed from the transportation track, requested methods are pipetted, and the sample waits until the results of all the AU methods are known. This way, the chemical reactions are carried out on both the analyzers in parallel. In AU→DXC ordering, the sample needs to wait at the AU until all the tests pipetted there are finished and validated. Only afterward, the sample's tube is released back to the track, and it can travel to the DXC analyzer. DXC collects the aliquots, initiates the reactions, and the sample is moved to storage, where it waits for the results of DXC methods. Therefore, in this order, the chemical reactions on the analyzers are performed sequentially. The time period during which the AU analyzer waits for the results introduces redundant delays in the sample processing, especially if any method needs to be repeated. The AU→DXC ordering occurs at 92.76% of the samples that need to visit both the analyzers, severely slowing down the testing process.

We have presented this finding to the laboratory technicians. They added additional rules that ensure the DXC→AU ordering of samples visiting both the biochemical analyzers and provided us with new operational data with these rules applied. The new situation is displayed in Figure 3.2. Now the samples are organized in a single cluster with processing time ranging from 11 to 29 minutes. The ordering rules resulted in a significant improvement with 17.5 minutes average sample biochemistry processing time with a 4.86 standard deviation. This is a 22 minutes decrease in comparison with the previous system configuration.

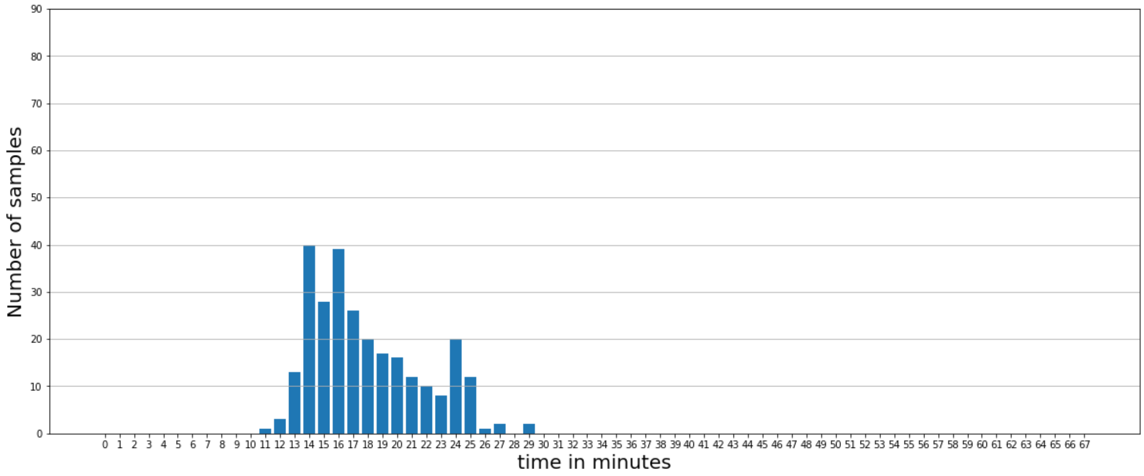


Figure 3.2 - Biochemical analysis duration histogram of samples visiting both analyzers after rules addition

3.2 Utilization of Biochemical Components

Another discovery we have made with our data analysis is the uneven workload distribution among the biochemical components. The situation is depicted in Figure 3.3, where each individual curve represents the utilization of one of the three biochemical reaction carousels. Recall from Section 2.3.1 that the AU has two reaction carousels (inner and outer), whereas the DXC has only one. The x-axis stands for individual hours throughout a day, and the y-axis describes the average utilization in respective hours. For instance, hour 11 represents the time interval from 11:00:00 to 11:59:59, and the average utilization of the carousels during this period is approximately 9.5% for the DXC, 34.5% for the AU's outer carousel, and 31% for the AU's inner carousel. We define hourly component utilization as the percentage ratio of component active time and availability time as

$$\text{Utilization} = \frac{at}{dt} * 100,$$

where:

- at (active time) stands for the amount of time the robotic arm of the carousel was actively pipetting aliquots.
- dt (availability time) stands for the amount of time the component was available for use. We assume no downtimes. Therefore this value equals one hour.

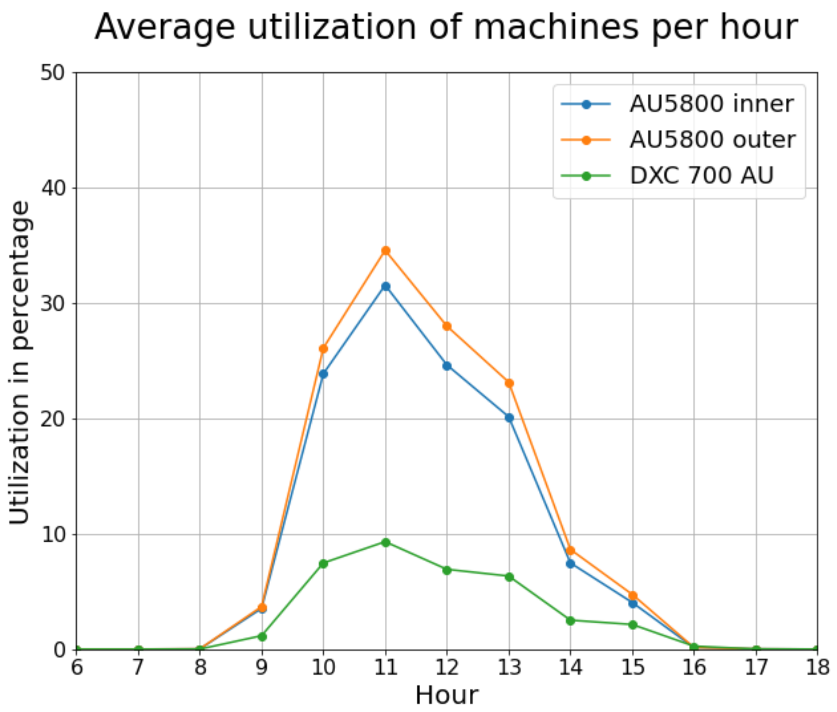


Figure 3.3 - Average utilization of laboratory machines throughout a day

Based on the research done in the review of quality indicators [5], the TAT positively correlates with machine utilization. Thus, the higher is the analyzer's utilization, the higher is the average sample TAT on the given analyzer, which is undesirable. Figure 3.3 shows a clear utilization disbalance between the AU components and the DXC analyzer, where the DXC's active time is less than one-third in comparison to any AU carousel. This represents potential issues in terms of system TAT as we have discussed earlier in Section 1.1.1, where we have concluded that balanced analyzer utilization is more important than high utilization of individual components for laboratories. It is necessary to move some of the tests performed on the AU analyzer to the DXC to achieve workload leveling among the components. The workload of individual carousels is determined by the set of methods assigned to the carousel. Thus, we need to find such an assignment of methods to components that balances their workload. Unfortunately, this problem can't be resolved as simply as the biochemistry routing, and its solution is discussed in detail in the following section.

4 System Throughput Maximization

This section introduces the first optimization problem solved in this thesis. Section 4.1 defines the optimization problem at hand. We designed two different approaches to solve this problem. The first approach utilizes integer linear programming and is explained in Section 4.2. The second approach based on the evolutionary algorithm is described in Section 4.3.

4.1 Problem Statement

In Section 3.2, we have uncovered imbalanced workload distribution among the biochemical carousels. This could represent a potential inefficiency in the laboratory automatization. Our goal is to achieve a uniform utilization leveling of the three biochemical components. Each carousel is associated with a set of methods that can be performed there. The assignment of methods determines the workload of the components. Thus, the task is to find a methods assignment for each carousel that minimizes the maximum workload among the analyzers, expressed as

$$\min \max_{c \in C}(l_c),$$

where C is the set of biochemical components (DXC, AU inner, and AU outer), c is a single component from this set and l_c is the workload of component c.

The difficulty of this task comes from the complexity of laboratory analyzers and the uncertainty in the laboratory process. The time of sample arrivals varies among individual days. One day, the sample arrival might be evenly balanced between morning and noon hours, whereas another day, there could be a sudden increase in one specific hour. Moreover, each sample is associated with a certain amount of requested tests. The number and the spectrum of the tests vary from sample to sample. Thus, the laboratory workload is a random variable generated by an unknown distribution. Therefore, we are solving a stochastic optimization problem. To solve the problem, we utilized our operational data to sample the unknown distribution and treated it as a deterministic optimization considering the sampling. This allows us to adapt our solutions to the unknown distribution.

4.1.1 Criteria

In the optimization, we consider three distinct criteria. The working time of individual components, the total throughput of the laboratory biochemistry, and the number of necessary transports between the DXC and the AU analyzers. The criteria are explained in detail in the following three subsections, starting with the component working time, followed by throughput, and transports being the last.

4.1.1.1. Component Working Time

The working time covers the amount of time the component's pipetting arm needed to collect aliquots to perform all requested tests. This includes both the active time, during which the pipetting arm collected aliquots, and the idle time when the arm was waiting. We assume that all the considered samples are available in a queue before the analyzer. Thus we eliminate the time intervals between individual sample's arrival. In this setting, the carousel's pipetting arm should operate without any pauses. This is indeed the case for the DXC analyzer. Therefore, the DXC working time can be defined as

$$w^{dxc} = \sum_{i \in I^{dxc}} \left(pt^{dxc} * \|J_i\|\right),$$

where w^{dxc} is working time of DXC component, I^{dxc} is the set of samples visiting the DXC analyzer, J_i is the set of all biochemical methods requested by sample i and pt^{dxc} is the DXC's pipetting duration. Effectively, it is the length of the DXC pipetting cycle multiplied by the number of conducted tests.

However, the same doesn't hold for the AU analyzer. The AU analyzer consists of two reaction carousels, each with one pipetting arm. However, both arms process the same sample in turns. Let us consider three different methods M1, M2, and M3. Methods M1 and M3 are assigned to the inner carousel (AU1), and M2 is mapped to the outer carousel (AU2). Then we consider a sample batch with 4 distinct samples with indices 1, 2, 3 and 4 requesting a certain set of methods – {M1, M2}, {M1, M3}, {M2} and {M1} respectively. The example is demonstrated in Table 4.1. The first column represents individual samples. Each row is identified by the sample's index and the set of requested methods in braces. The second and third columns are reserved to demonstrate the processing of individual samples on AU's carousels in the following examples.

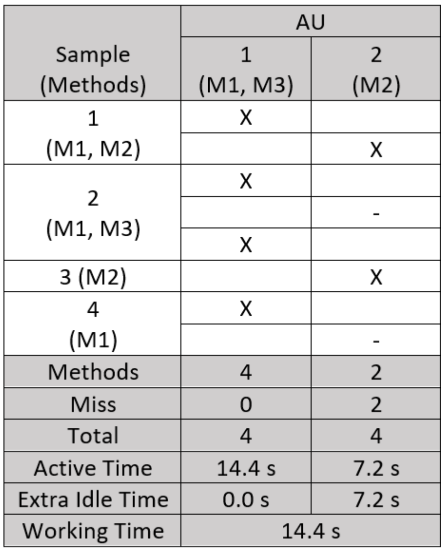
Sample (Methods)	AU	
	1 (M1, M3)	2 (M2)
1 (M1, M2)		
2 (M1, M3)		
3 (M2)		
4 (M1)		

Table 4.1 – Considered AU processing example

The samples visit the analyzer in the ascending ordering of their indices. The first sample requires methods M1 and M2. The method M1 is available on the AU1 carousel. Thus the carousel's pipetting arm will collect an aliquot and place it into a reaction slot. While the AU1's arm is transporting the aliquot, the arm of AU2 is free to do its own pipetting. The next sample's request is M2. This method is assigned to AU2, so its arm collects the aliquot. The first sample is now fully processed, and the sample batch is shifted to move sample 2 underneath the analyzer. The batch movement is swift enough not to cause any delays in the operation.

Now, AU2's pipetting arm is transporting collected aliquot. Meanwhile, the first pipetting arm finished its previous operation and can pipet another method. The second sample requests methods M1 and M3, both assigned to the first component. Therefore, its arm collects the aliquot for one of the methods. Now it is the second arm's turn. However, none of the methods requested by sample 2 is available on AU2. Therefore, there is no task for the second pipetting arm. The arm needs to wait, skipping a pipetting cycle and introducing redundant idle time to the workflow. Once the first arm finishes its operation, it can process the other requested method, and the batch is shifted. The second arm then processes the single method M2 requested by the third sample without any delays. The same happens for the last sample 4, whose method M1 is processed by the first arm. However, there are no more requested methods or samples in the queue, so the second arm misses another pipetting cycle. Thus, the second arm missed two pipetting cycles in total, resulting in 7.2 second idle time.

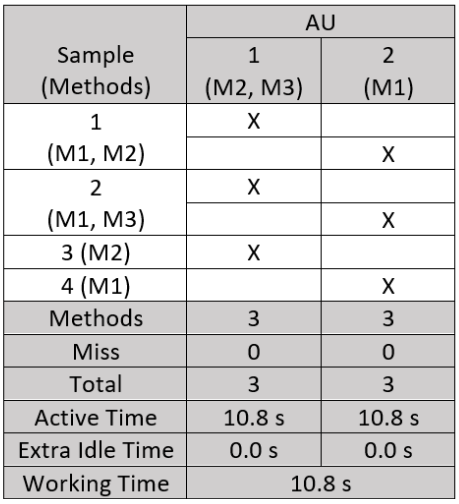
The described sample processing is shown in Table 4.2. The second and third columns demonstrate whether a carousel missed a pipetting cycle or not, where "X" stands for cycle hit and "-" represents cycle miss. Samples requested six methods in total. Four of those were processed by AU1 and two by AU2. Thus, AU1 actively worked for 14.4 seconds, whereas AU2 was active for 7.2 seconds and spent another 7.2 seconds waiting. Therefore, the whole pipetting process took 14.4 seconds since the individual arms operate in parallel.



Sample (Methods)	AU	
	1 (M1, M3)	2 (M2)
1 (M1, M2)	X	
		X
2 (M1, M3)	X	
		-
	X	
3 (M2)		X
4 (M1)	X	
		-
Methods	4	2
Miss	0	2
Total	4	4
Active Time	14.4 s	7.2 s
Extra Idle Time	0.0 s	7.2 s
Working Time	14.4 s	

Table 4.2 - AU sample processing

However, appropriate methods assignment can reduce the length of this process. Let us change the assignment to (M2, M3) on AU1 and (M1) on AU2. The AU sample processing with this method assignment is demonstrated in Table 4.3. Now, beginning with sample 1, method M2 is processed by the first arm. Then method M1 is pipetted by the second arm. The processing moves to the second sample, where the AU1's arm collects the aliquot for method M3, and the second arm perform its collection for method M1. Sample 2 is now wholly processed without any missed pipetting cycles, unlike in the previous configuration. Afterward, the first arm processes the single method M2 requested by the third sample, and without any delays, the second arm performs the M1 test ordered for sample 4. The sample processing is now completed without any missed pipetting cycles. Both carousels performed three tests, so their active time is 10.8 seconds. The whole operation took 10.8 seconds, which is 3.6 seconds less compared to the previous configuration.



Sample (Methods)	AU	
	1 (M2, M3)	2 (M1)
1 (M1, M2)	X	
		X
2 (M1, M3)	X	
		X
3 (M2)	X	
4 (M1)		X
Methods	3	3
Miss	0	0
Total	3	3
Active Time	10.8 s	10.8 s
Extra Idle Time	0.0 s	0.0 s
Working Time	10.8 s	

Table 4.3 - AU sample processing with proper method assignment

We define this behavior as the **problem of odd methods**. It is a situation where any of the carousels can introduce additional idle time to the sample's processing based on the assignment of methods to the AU's components. However, the occurrence of odd methods also depends on the order in which samples visit the analyzer. Consider the previous example where we eliminated the idle time. If the samples visited the analyzer in order 1→4→2→3, it would cause one pipetting cycle skip when moving from sample 1 to sample 4. The processing of sample 1 ends with method M1 assigned to the second carousel, but the single method requested by sample 4 is also M1. Therefore, the AU1's arm would skip one pipetting cycle. This adds another layer of complexity to the optimization task. However, the sample order is determined by the DxA5000 automation system and is beyond our control. Thus, we do not consider the sample ordering and operate with a simplified version of the odd methods. We compute the number of odd methods caused by each sample as the difference between the number of tests on AU1 and AU2. This represents the worst-case scenario of the miss occurrences because sample ordering can never worsen the situation. Thus, we define the working time of the AU's first component as

$$w^{AU,1} = \sum_{i \in I^{AU}} \left(pt^{AU} \cdot \left(\|J_i^1\| + \max\left(0, \|J_i^2\| - \|J_i^1\|\right)\right)\right),$$

and similarly, the working time of the AU's second component as

$$w^{AU,2} = \sum_{i \in I^{AU}} \left(pt^{AU} \cdot \left(\|J_i^2\| + \max\left(0, \|J_i^1\| - \|J_i^2\|\right)\right)\right),$$

where $w^{AU,c}$ for $c \in \{1, 2\}$ is the working time of component c, I^{AU} is the set of samples visiting AU analyzer, pt^{AU} is the pipetting duration of one pipetting arm, and J_i^c is the set of methods requested by sample i assigned to carousel c.

4.1.1.2. Biochemistry Throughput

Biochemistry throughput numerically describes the efficiency of biochemical analysis. It is equal to the total number of methods conducted by the biochemical analyzers over a time period. We assume that the system has processed all considered samples. Therefore, to compute biochemistry throughput, we divide the total number of all considered biochemical tests by the working time of the biochemistry system. The biochemical operation is completed when all components have finished aliquots collection. Therefore, the working time of biochemistry is equal to the maximum over the working times of individual components, and the hourly throughput can be calculated as

$$\text{Throughput} = \frac{n}{\max\limits_{c}(w_c)} \cdot 3600,$$

where n is the total number of all requested tests, c is a biochemical component and w_c is the working time of a component c. The component working time is identical to the criterion described in the previous subsection. Thus, the system throughput depends on the workload balancing. The more balanced a load of biochemical components is, the lower is the denominator in throughput calculation, resulting in system throughput increases. Component working time is computed in seconds, so the throughput needs to be multiplied by 3600 to express throughput in hours.

4.1.1.3. Transports

We have discussed earlier that the primary quality indicator for hospital laboratories is low TAT of high priority samples. High throughput is associated with a low processing time of individual samples. A decrease in a sample's processing time results in shorter waiting times for the remaining tubes in the queue. Thus, effective processing of a single sample affects the testing process' length of the following samples. Therefore, maximizing biochemistry throughput will improve the TAT of all considered samples on average. However, it might cause an unintended increase for some samples. This depends on whether the sample needs to be transported between the DXC and the AU analyzers. Such a sample would first visit the DXC analyzer, where it needs to wait in a queue until all preceding tubes are processed. Afterward, it moves to the AU, where the sample waits for its turn again, and it needs to undergo all the AU manipulation processes discussed earlier. This may cause delays in a sample's processing. Therefore, it is beneficial to create a method's assignment without sample transport between analyzers. One way of sample transport computation is to sum the product of two binary variables over all samples where each variable indicates whether the sample requests at least one test on a biochemical analyzer, i.e.,

$$transports = \sum_{i \in I} (t_i^{AU} \cdot t_i^{DXC}),$$

where I is the set of all samples and t_i^{AU}, t_i^{DXC} is a binary value equal to one if at least one requested method of sample i is conducted by the AU or the DXC analyzer, respectively.

Unfortunately, the number of necessary transports negatively correlates with the workload balancing. The more evenly the available work is distributed among the components, the more likely a transport will occur. A trivial solution to the minimization of transports would be to assign all methods to one analyzer and ignore the other analyzer altogether. However, this is hardly an ideal configuration, as it would likely overload the single working analyzer. To cope with the trade-off, we use the system throughput as the main optimization criterion and introduce the number of transports as a side criterion limited by a fixed upper bound. Thus, we can change the importance of the criteria depending on the upper bound value. This allows us to either emphasize system throughput or to stress the individual sample's TAT.

4.1.2 Constraints

In this subsection, we discuss the constraints we consider in the methods assignment problem. All the biochemical methods offered by the laboratory Prevedig can be performed by both the DXC and the AU analyzers. Therefore, one could consider a trivial solution, where all available methods are triplicated and assigned to all the components. However, such a solution is not ideal for multiple reasons. Configuring components in such a way is more costly compared to a configuration where the methods are distributed without multiplication. Materials and reagents required to perform a test would have to be provided to all the components, potentially leading to higher operational costs. Moreover, the automatization efficiency would be determined solely by the routing and scheduling algorithms in DxA5000 that are beyond our control. Thus, we have enforced a constraint that forces each method to be assigned to precisely one biochemical component. However, there are a few exceptions to this rule - methods LIH and ISE.

LIH methods (Lipemia, Icteric, Hemolysis) monitor the quality of a blood sample. Sample hemolysis, lipemia, or icteric can significantly influence the results of photometric analysis, causing erroneous results. Therefore, LIH methods are performed for every blood sample requesting the biochemical analysis. For this reason, the tests need to be always available on all the biochemical components to avoid unnecessary sample transportations. Moreover, if a sample visits both biochemical analyzers, LIH is always performed by the DXC. The ISE methods are performed only if a clinician orders them. These methods are duplicated for the same reason – to avoid unnecessary transportations. The AU analyzer has a specialized component performing ISE tests. Therefore, ISE does not introduce additional load to the AU's biochemical components. If a sample visits the AU analyzer, ISE methods are performed there. Otherwise, the methods are pipetted by DXC's carousel, adding more work to the DXC analyzer. Although LIH and ISE both stand for a set of methods, they are all performed from a single aliquot. Thus, the processing of LIH and ISE requires only one pipetting cycle.

Another constraint concerns the washing of the pipetting needles. A pipetting cycle of a robotic arm consists of multiple operations – an aliquot collection, reagent addition, and needle washing. The needle needs to be washed to avoid sample contamination or an unintended reaction between reagents. However, some pairs of reagents are more sensitive to each other. A minimal number of performed washings is defined for each pair of interfering methods, and their pipetting cannot be completed until the requested number of washings has been performed. Let us assume a pair of methods M1 and M2, that require three washing cycles between their processing. Let S be a sample requesting methods M1 and M2 both assigned to a single component. If the methods are processed in order M1→M2, the component must wash the needle thrice after processing method M1 before it can pipet M2. One washing is done during the cycle that pipetted M1. However, two still remain. Therefore, the component skips two additional pipetting cycles to wash the needle before processing method M2, introducing additional idle time in the process. The interference is symmetric, therefore ordering M2→M1 leads to the same situation. If the sample requested two other methods M3 and M4, processed in order M1→M3→M4→M2, the cycle skipping would no longer occur since the washing was performed before M2's processing began. Thus, the number of skipped cycles caused by the method's interference depends on the assignment of methods and their execution order. Unfortunately, we cannot determine the processing order of the sample's tests. Thus, we instead consider a constraint forbidding the assignment of interfering methods on the same component to avoid interference.

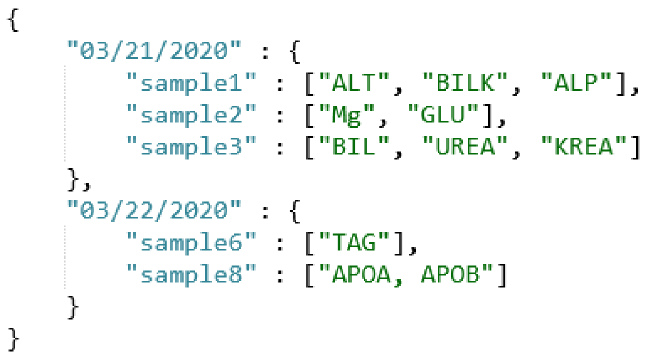
The last constraint is associated with the particular requirements of some methods. The methods BIL and BILK are based on the subtraction of two different results. The components collect two aliquots and perform two versions of the test - BIL/BILK and BIL blind/BILK blind - to perform the methods. The result is equal to the difference between the blind and non-blind versions. Moreover, these two methods cannot be assigned to the AU's inner carousel. Thus, they can be present on either the AU's outer carousel or on the DXC. The unique requirements of these methods are considered in our optimization.

4.2 Integer Linear Program

One of the optimization approaches designed to solve the methods assignment problem is based on integer linear programming (ILP), a mathematical optimization technique based on a set of linear equations and inequalities. Subsection 4.2.1 describes the design of the ILP model. Subsection 4.2.2 offers an overview of utilized sets, parameters, and variables. Finally, Subsection 4.2.3 contains the mathematical representation of the designed ILP model

4.2.1 ILP Model

The designed ILP model minimizes the biochemical system working time w.r.t. the constraints mentioned above. The optimization utilizes the laboratory operational data in a key-value format. The data is split into individual days, denoted as a set D. Each day consists of a set of samples I_d, where i represents individual samples from this set. A list of biochemical methods J_i accompanies every sample. Figure 4.1 shows an example of the data expressed in a JSON format. Data in this format determines values for the ILP parameters and the number of its decision variables and constraints. Therefore, the model's computational complexity is based on the data size.



```
{
    "03/21/2020" : {
        "sample1" : ["ALT", "BILK", "ALP"],
        "sample2" : ["Mg", "GLU"],
        "sample3" : ["BIL", "UREA", "KREA"]
    },
    "03/22/2020" : {
        "sample6" : ["TAG"],
        "sample8" : ["APOA, APOB"]
    }
}
```

Figure 4.1 - Json Example of Optimization Data

Our task is to find an optimal assignment of biochemical methods to the three available components in the system. To express the method assignment in the model, we introduce a new binary decision variable $x_j^{a,b}$ where j is a certain method, a is the analyzer's index, and b represents the analyzer's component. The variable is equal to one if method j is assigned to carousel b of analyzer a, and 0 otherwise. There are three limitations regarding the methods assignment. First, each method has to be assigned to exactly one analyzer. Thus, we add constraint for each method, and sum $x_j^{a,b}$ over all biochemical components and the sum must be equal to one,

$$\sum_{a \in A} \sum_{b \in B^a} x_j^{a,b} = 1, \qquad \forall j \in J,$$

where A is the set of analyzers, B^a is the set of components of analyzer a and J is the set of all biochemical methods. Because of the multiplicated assignment of methods LIH and ISE, they are excluded from the set J and handled individually. Note that $a = 1$ represents the AU analyzer, whereas $a = 2$ stands for the DXC analyzer. AU analyzer has two carousels, and b can be either 1 for the inner carousel or 2 for the outer carousel. DXC contains only 1 component, so b is always equal to 1 for the DXC.

Second, we cannot assign interfering pairs of methods on the same component. Therefore, for each pair of interfering methods and each component, we add constraint,

$$x_j^{a,b} + x_{j'}^{a,b} \leq 1, \qquad \forall (j, j') \in F, \forall a \in A, \forall b \in B^a,$$

with F representing the set of interfering pairs of methods. We sum the assignment of methods j and j' on component b and upper bound it by one. Therefore, at most one of those methods can be assigned to component b of analyzer a.

The last constraint regarding the methods assignment is the limitation of methods BIL and BILK that can't be assigned to the AU's inner carousel. Thus, we set the respective assignment variables to zero

$$x_{BIL}^{1,1} = 0;\ x_{BILK}^{1,1} = 0.$$

The LIH methods are by default assigned to all biochemical analyzers. Moreover, the methods are also available on both AU's components. Thus, to accurately approximate the load caused by LIH, we have to first decide which analyzer will execute them. Fortunately, this decision is based on a simple rule. If a sample visits the DXC analyzer, LIH are performed there. Otherwise, it is executed at the AU analyzer. Thus, for each sample, we introduce new binary decision variable $h_{d,i}$ where d represents a particular day in the optimization data and i is a specific sample from day d. Variable $h_{d,i}$ is equal to one if the LIH method will be performed on the AU analyzer. To enforce the LIH decision rule, we introduce two new constraints for each sample

$$\frac{\|J_i\| - \sum_{J_i} x_j^{2,1}}{\|J_i\|} \geq h_{d,i}, \qquad \forall d \in D, \forall i \in I_d,$$
$$1 - \sum_{J_i} x_j^{2,1} \leq h_{d,i}, \qquad \forall d \in D, \forall i \in I_d.$$

Recall that $a = 2$ represents the DXC analyzer, which has only one components. In the first constraint, we take the total number of methods requested by the sample i and subtract the number of the requested methods assigned to the DXC analyzer. Then we divide the subtraction by the total number of requested tests. The $h_{d,i}$ is a binary variable. Therefore, if at least one of the requested methods is assigned to the DXC analyzer, the division numerator will be less than the denominator and the $h_{d,i}$ is forced to be zero. Thus, the LIH method will be performed on the DXC analyzer. If none of the requested methods is assigned to the DXC, the constraint allows $h_{d,i}$ to be either zero or one. However, the second constraint subtracts the number of requested tests assigned to the DXC from one. Therefore, if none of the methods is assigned to the DXC, $h_{d,i}$ is forced to be equal to one, and LIH are performed on the AU analyzer. The two constraints together enforce the LIH decision rule.

However, we have only decided whether the DXC or the AU performs the LIH. Should the AU perform LIH, we also need to decide on which component. For this purpose, we introduce two new binary decision variables $h_{d,i}^{1,1}$ and $h_{d,i}^{1,2}$ for each sample, deciding on which AU's component LIH will be executed. If $h_{d,i}^{1,1}$ is equal to one, LIH are pipetted on the inner carousel. Variable $h_{d,i}^{1,2}$ represents the outer carousel.

To determine their value, we sum the values of $h_{d,i}^{1,1}$ and $h_{d,i}^{1,2}$ and put them equal to $h_{d,i}$ in constraint

$$h_{d,i}^{1,1} + h_{d,i}^{1,2} = h_{d,i}, \qquad \forall d \in D, \forall i \in I_d,$$

introduced for each sample. Thus, if $h_{d,i}$ is equal to one, one of the components' variables needs to be equal to one as well. If the DXC performs LIH, $h_{d,i}$ is zero and both component variables are also zero. Earlier, we have discussed the effects of odd methods on the component's working time. The number of occurred odd methods depends on the distribution of methods among the AU's components, which is expressed by the decision variable $x_j^{1,b}$. Thus, the number of odd methods is a variable dependent on the variable $x_j^{1,b}$. To estimate the number of odd occurrences, we establish for each sample two new integer variables $n_{d,i}^{1,1}$ and $n_{d,i}^{1,2}$ denoting the number of aliquot collections performed by the first AU's component and the second AU's component, respectively. The value of these variables is computed as the number of aliquots pipetted by the component needed to process the sample i plus the decision variable determining which component performed LIH. The resulting equalities for the individual components are

$$n_{d,i}^{1,1} = \sum_{J_i} p_j x_j^{1,1} + h_{d,i}^{1,1}, \qquad \forall d \in D, \forall i \in I_d,$$
$$n_{d,i}^{1,2} = \sum_{J_i} p_j x_j^{1,2} + h_{d,i}^{1,2}, \qquad \forall d \in D, \forall i \in I_d,$$

where p_j is an integer parameter defining the number of aliquots needed to execute method j. Utilizing these two values, the number of odd occurrences for each AU's component b is

$$\max\left(0, n_{d,i}^{1,b'} - n_{d,i}^{1,b}\right), b' \neq b,$$

where b' represents the other AU's component. Therefore, to determine the number of odd occurrences at the first component, we subtract the number of methods on the first component from the number of methods processed by the second component and take a maximum from this value and zero.

Now we have everything to express the criterion of the model. We search for an assignment of methods with the most evenly distributed workload. Thus, we seek a solution minimizing system working time, which is expressed as the maximum component's working time.

Because the optimization data is split into individual days, we compute the system workload for each day and minimize the sum

$$\min \sum_{D} \max\left(w_d^{1,1}, w_d^{1,2}, w_d^{2,1}\right),$$

where $w_d^{a,b}$ is the working time of component b of analyzer a in day d.

However, the working time computation differs slightly for each component. Therefore, each component needs to be expressed individually. The working time of the AU's inner carousel is equal to the number of aliquot collections performed for regular methods in set J plus the number of LIH methods, multiplied by the pipetting duration. The active pipetting time of the component is expressed as

$$at_d^{1,1} = pt^1 \left(\sum_{J} p_j s_{d,j} x_j^{1,1} + \sum_{I_d} h_{d,i}^{1,1} \right),$$

where $at_d^{a,b}$ is the component's active time b in day d, pt^a is the pipetting duration of analyzer a and $s_{d,j}$ is the number of samples requesting method j in day d. The first sum represents the number of aliquots pipetted for the regular methods and the second sum stands for the number of LIH methods.

Then we add the idle time as the number of missed pipetting cycles caused by the odd methods multiplied by AU's pipetting duration

$$it_d^{1,1} = pt^1 \sum_{I_d} \max\left(0, n_{d,i}^{1,2} - n_{d,i}^{1,1}\right).$$

AU's inner carousel is idle only if the outer carousel collects more aliquots. Thus, the number of skipped pipetting cycles is equal to the difference between collections performed by the outer carousel minus collections done by the inner carousel. If the outer carousel performs more collections, the inner carousel needs to wait until the process is finished. Otherwise, the number of skipped cycles is equal to zero.

Having expressed computations for active and idle time, we can finally define the inner component working time as

$$w_d^{1,1} = at_d^{1,1} + it_d^{1,1} = pt^1 \left(\sum_{J} p_j s_{d,j} x_j^{1,1} + \sum_{I_d} h_{d,i}^{1,1} \right) + pt^1 \sum_{I_d} \max\left(0, n_{d,i}^{1,2} - n_{d,i}^{1,1}\right).$$

The working time for the outer carousel is expressed similarly with different indices as

$$w_d^{1,2} = at_d^{1,2} + it_d^{1,2} = \; pt^1\left(\sum_J p_j s_{d,j} x_j^{1,2} + \sum_{I_d} h_{d,i}^{1,2}\right) + pt^1 \sum_{I_d} \max\left(0, n_{d,i}^{1,1} - n_{d,i}^{1,2}\right).$$

The DXC working time calculation is simpler as it has only one component. Thus, the odd methods do not occur at the analyzer due to odd methods. However, unlike AU, the DXC doesn't have a specialized unit performing ISE methods. Therefore, the ISE has to be processed by the biochemical component. Similarly to LIH, the number of ISE performed by the DXC depends on the assignment of other methods that determines whether a sample needs to visit the DXC or not. However, we do not know the process of deciding on which analyzer ISE will be performed. Therefore, we have decided to approximate the rate of ISE methods performed by the DXC analyzer from the operational data and utilized this rate to compute a constant ise_d representing the number of ISE methods performed by the DXC analyzer in day d. The working time of the DXC analyzer is calculated as the number of aliquots collected by the analyzer plus the LIH methods performed by the DXC plus the ISE constant, all multiplied by the DXC's pipetting time, resulting in equality

$$w_d^{2,1} = pt^2\left(\sum_J p_j s_{d,j} x_j^{2,1} + \sum_{I_d}(1 - h_{d,i}) + \; ise_d\right).$$

Now that we have expressed working time calculations for all the components, we can substitute the calculations and express the model's criterion as

$$\min \sum_D \left(\max\left(pt^1\left(\sum_J p_j s_{d,j} x_j^{1,1} + \sum_{I_d} h_{d,i}^{1,1}\right) + pt^1 \sum_{I_d} \max\left(0, n_{d,i}^{1,2} - n_{d,i}^{1,1}\right), \right.\right.$$
$$pt^1\left(\sum_J p_j s_{d,j} x_j^{1,2} + \sum_{I_d} h_{d,i}^{1,2}\right) + pt^1 \sum_{I_d} \max\left(0, n_{d,i}^{1,1} - n_{d,i}^{1,2}\right),$$
$$\left.\left. pt^2\left(\sum_J p_j s_{d,j} x_j^{2,1} + \sum_{I_d}(1 - h_{d,i}) + \; ise_d\right)\right)\right).$$

We have discussed the minimization of the biochemistry working time. However, we would also like to limit the number of necessary transports, because the high amount of transports may result in high TAT of the transported samples. Thus, we need to decide for each sample whether it was transported or not.

For each sample, we establish additional binary decision variable $t_{d,i}$ equal to one if the sample i needs to be transported. To determine the variable's value, we add a new constraint for each sample, for each pair of methods, and for each pair of analyzers, which in this case is only the AU and the DXC:

$$t_{d,i} \geq \sum_{B^a} x_j^{a,b} + \sum_{B^{a'}} x_{j'}^{a',b} - 1, \qquad \forall d \in D; \forall i \in I_d;\ \forall j, j' \in J_i:\ j \neq j';\ \forall a, a' \in A, a < a'$$

If the sample requests at least one pair of methods j and j' where $j \neq j'$, j is assigned to one component and j' is assigned to the other component, then $t_{d,i}$ has to be greater or equal to one. However, $t_{d,i}$ is a binary variable and is forced to be one. Otherwise, $t_{d,i}$ can be either 0 or 1. The model can decide to "transport" a sample that doesn't need to be transported. However, the goal is to identify all necessary transports and limit their amount by an upper bound. Therefore, it won't affect the model's functionality. Lastly, we limit the average daily number of transported samples by an upper bound. Transport limitations aim to improve the sample's TAT. However, this is meaningful only if the biochemical analysis determines the sample's TAT. This is the case only if the sample doesn't request a long immunological analysis. Immunology tests require up to 75 minutes. According to our analysis performed earlier, depicted in Figure 3.2, the longest biochemical analysis took 30 minutes. Therefore, we pre-computed for each sample a binary parameter $st_{d,i}$ equal to one if the sample requests an immunological test with a reaction lasting at least 35 minutes. If $st_{d,i}$ is equal to one, we do not consider sample's transport as it does not affect the sample's TAT. Thus, the transportation constraint is defined as the sum of the sample transportation indicator variable $t_{d,i}$ multiplied by one minus the binary transportation skipping parameter $st_{d,i}$:

$$\alpha \geq \frac{1}{\|D\|} \sum_{I_d} (1 - st_{d,i}) t_{d,i}$$

Where α is the upper bound of average daily transport occurrences.

The two following subsections provide an overview of the discussed ILP model. Subsection 4.2.2 defines the meaning of all sets, parameters, and decision variables utilized in the model. Subsection 4.2.3 contains its mathematical representation.

4.2.2 Overview of ILP Sets, Parameters and Decision Variables

Sets

- D – The set of all considered days.
- I – The set of all samples. I_d denotes a set of samples processed in day d.
- J – The set of all biochemical methods except LIH and ISE. These two methods are handled individually because of their multiplicity. J_i represents regular methods requested by sample i.
- A – The set of biochemical analyzers (AU and DXC).
- B^a – The set of components of analyzer a.
- F – The set of all pairs of interfering methods.

Parameters

- pt^a – The pipetting operation length of analyzer a.
- p_j – The number of aliquot collections needed to process method j.
- $s_{d,j}$ – The total quantity of samples requesting method j in day d.
- ise_d – Estimation of the ISE methods performed on the DXC analyzer.
- $st_{d,i}$ – Binary parameter equal to one if sample i requests long immunological method. If so, the transport of such a sample is not considered.
- α – Upper bound on the average daily amount of occurred transports.

Decision Variables

- $x_j^{a,b}$ – A binary variable equal to 1 if method j is assigned to a component b of analyzer a. Note that AU has two components, whereas DXC has only 1.
- $h_{d,i}$ – A binary variable determining whether the LIH methods of sample i from day d will be performed on the AU analyzer or not. The variable is equal to 1 if LIH are performed by the AU and 0 otherwise.
- $h_{d,i}^{a,b}$ – If AU performs the LIH methods, it is necessary to decide on which component. This binary variable is equal to 1 if LIH of sample i are performed by component b of the AU analyzer.
- $n_{d,i}^{a,b}$ – The number of aliquots collected by the component b of analyzer a in order to process sample i. This variable determines the number of odd methods.
- $t_{d,i}$ – A binary variable. Equal to 1, if sample i from day d needs to visit both biochemical analyzers.

4.2.3 ILP Mathematical Representation

$$\begin{aligned}\min\sum_{D}\Bigg(\max\Bigg(& pt^{1}\left(\sum_{J} p_j s_{d,j} x_j^{1,1} + \sum_{I_d} h_{d,i}^{1,1}\right) + pt^{1}\sum_{I_d}\max\left(0, n_{d,i}^{1,2} - n_{d,i}^{1,1}\right), \\ & pt^{1}\left(\sum_{J} p_j s_{d,j} x_j^{1,2} + \sum_{I_d} h_{d,i}^{1,2}\right) + pt^{1}\sum_{I_d}\max\left(0, n_{d,i}^{1,1} - n_{d,i}^{1,2}\right), \\ & pt^{2}\left(\sum_{J} p_j s_{d,j} x_j^{2,1} + \sum_{I_d}(1 - h_{d,i}) + \ ise_d\right)\Bigg)\Bigg)\end{aligned}$$

St.

1) $\sum_{A}\sum_{B^a} x_j^{a,b} = 1$ $\qquad \forall j \in J$

2) $x_j^{a,b} + x_{j'}^{a,b} \leq 1$ $\qquad \forall (j, j') \in F, \forall a \in A, \forall b \in B^a$

3) $x_{BIL}^{1,1} = 0;\ x_{BILK}^{1,1} = 0$

4) $h_{d,i}^{1,2} + h_{d,i}^{1,1} = h_{d,i}$ $\qquad \forall d \in D, \forall i \in I_d$

5) $\frac{\|J_i\| - \sum_{J_i} x_j^{2,1}}{\|J_i\|} \geq h_{d,i} \geq 1 - \sum_{J_i} x_j^{2,1}$ $\qquad \forall d \in D, \forall i \in I_d$

6) $n_{d,i}^{1,2} = \sum_{J_i} p_j x_j^{1,2} + h_{d,i}^{1,2}$ $\qquad \forall d \in D, \forall i \in I_d$

7) $n_{d,i}^{1,1} = \sum_{J_i} p_j x_j^{1,1} + h_{d,i}^{1,1}$ $\qquad \forall d \in D, \forall i \in I_d$

8) $t_{d,i} \geq \sum_{B^a} x_j^{a,b} + \sum_{B^{a'}} x_{j'}^{a',b} - 1$ $\qquad \forall d \in D, \forall i \in I_d$

$\forall j, j' \in J_i\colon\ j \neq j'$

$\forall a, a' \in A, a < a'$

9) $\alpha \geq \frac{1}{\|D\|}\sum_{I_d}(1 - st_{d,i})t_{d,i}$

4.3 Evolutionary Algorithm

This section describes the evolutionary approach we have designed to solve the methods assignment problem as an alternative to the ILP model. First, we will describe the general idea behind evolutionary algorithms in Subsection 4.3.1. Then we discuss our implementation of the algorithm in Subsection 4.3.2.

4.3.1 Evolutionary Algorithms

Evolutionary algorithms (EAs), or sometimes also called genetic algorithms (GAs), are randomized optimization models inspired by natural evolution [15][16]. A potential solution is encoded as a chromosome, and a fitness function evaluates the solution's quality. A set of various chromosomes forms a population. EAs are iterative algorithms that are initialized with a randomly generated population with pre-defined size. EAs select a subset of chromosomes from the population according to their fitness function and create new ones by combining features of the selected chromosomes with each iteration. This process approximates the optimal solution. A local search is applied to each chromosome in the population to help the algorithm converge faster. The local search is a simple search strategy that attempts to modify the chromosome to improve its fitness value. The algorithms apply randomized mutations to the generated chromosomes, significantly increasing the algorithm's coverage with the goal to avoid convergence to a local optimum. The flowchart depicts the whole evolutionary process in Figure 4.2. The respective evolutionary steps are as follows:

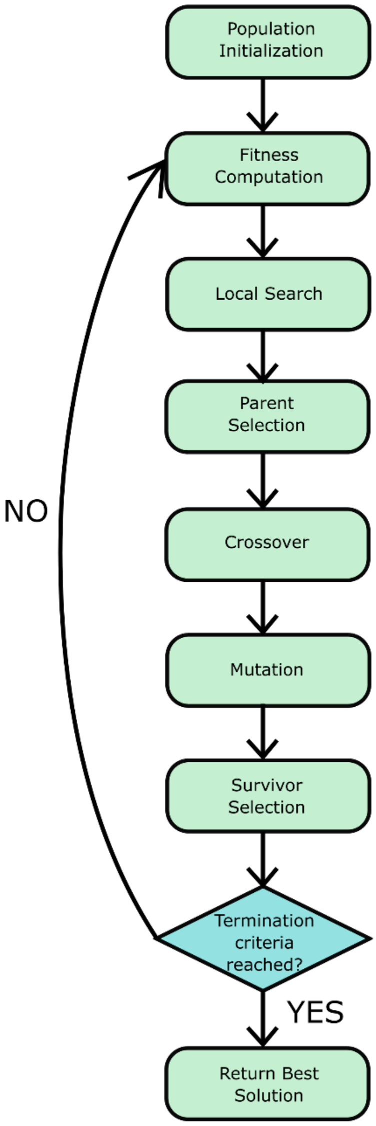


Figure 4.2 - EA Flowchart

- **Population initialization** – Population is a set of solutions in the current EA generation. It is usually defined as an array with a fixed size. The selection of an appropriate size is crucial to the EAs' efficiency. It needs to be large enough to cover the domain of fitness function. However, EAs computational complexity scales with its size. The population is typically initialized either randomly or with a domain-specific heuristic generator.
- **Fitness Computation** – In this phase, each solution in the current generation is evaluated w.r.t. a fitness function. This function evaluates the quality of individual solutions and defines the

optimization objective. Depending on the problem, the calculation of the fitness function may be costly.

- **Local Search** – The algorithm iterates over solutions in the current generation and attempts to improve them w.r.t. the fitness function during the local search. The improvement is made via a search strategy that could find an optimal solution, given enough time. This phase aims to modify the current solutions and shift them closer to the optimum.
- **Parent Selection** – This phase marks the selection of individual solutions as parents. The features of the selected parents are utilized to generate new solutions that will form the next generation. The parent selection is usually probabilistic, and the probability of a solution is proportional to its fitness function. Therefore, the fittest solutions have the highest selection probability. However, even an unfit solution may be selected to maintain variability in the population and avoid convergence to a local optimum.
- **Crossover** – As the name indicates, the idea of this operation is to combine features of two different parents to generate new solutions. The choice of parent features and their combination is usually made randomly.
- **Mutation** – Mutation is a stochastic operation that slightly modifies the solutions created during the crossover. The main goal of mutations is to maintain diversity in the population and is typically applied with a low probability. With high mutation probability, the evolutionary algorithm reduces to a random search.
- **Survivor Selection** – In this phase, the genetic algorithm decides for each solution whether it will be allowed to the next generation or not based on its fitness value. Unlike parent selection, the survivor selection is often deterministic. One approach to the selection is to sort all the available solutions by its fitness value and select the ones with the highest fitness.
- **Termination -** At some point, a genetic algorithm needs to be terminated and the best solution extracted. If the optimal fitness value is known beforehand, the evolution can be halted after any solution reaches optimal fitness. However, because of the random nature of the algorithm, this may never happen. Thus, it is reasonable to introduce additional termination conditions, such as an upper bound on the number of iterations, maximum elapsed CPU time, or termination after the improvement between generations drops below a threshold [16].

4.3.2 Proposed Evolution Strategy

Chromosome and Population

We need to decide on an assignment for 42 different biochemical methods in our optimization, excluding ISE and LIH methods. Thus, we encode a chromosome as an array of 42 integer values with one value for each method. Each value encodes the assignment of a method to the components. The value is either

0 (AU inner), 1 (AU outer), or 2 (DXC). The chromosome encoding enforces the constraint limiting method's assignment to exactly one component. The size of the population is set to 50 solutions, where each chromosome is generated randomly w.r.t. a discrete uniform distribution, maintaining equal generation probability for each value. Methods BIL and BILK can't be assigned to the AU's inner component. Thus, the assignment generation for these methods is repeated until it is different from zero. A solution is represented as a tuple containing a chromosome and its fitness value. The population is a sorted array of solutions, where its fitness value determines the solution's position. The fitness function computes the solution's throughput w.r.t. its methods assignment to measure its quality. The algorithm's goal is to maximize the biochemistry throughput. Thus, the population is sorted in descending order.

Fitness Function

The fitness computation is based on the laboratory operational data. Similarly to the ILP approach, the optimization data is separated into individual days containing a set of samples, and each sample is associated with a set of requested methods. According to our definition of throughput in Subsection 4.1.1.2, we need to estimate the working time of individual components. The working time is equal to the active time of the component plus the component's idle time. The working time of a component depends on the methods assignment. To utilize the assignment in calculations, we define an indicator variable $x_{c,j}^{a,b}$, that is equal to one if method j is assigned to the component b of the analyzer a in the chromosome c.

However, we need to decide for each sample where its LIH methods are performed. The decision between DXC and AU analyzers follows the same rule as before. If a sample requests at least one method other than LIH and ISE assigned to the DXC analyzer, the LIH is performed there. For this purpose, we will use an indicator variable $h_{c,d,i}$ equal to one if the AU analyzer performs the LIH expressed as

$$h_{c,d,i} = 1 - \mathbf{1}\left[\sum_{j \in J_i} x_{c,j}^{2,1} > 0\right],$$

where J_i is the set of methods requested by the sample i, $x_{c,j}^{2,1}$ is an indicator variable equal to one if method j is assigned to the DXC analyzer in the chromosome c, and $\mathbf{1}\left[\sum_{j \in J_i} x_{c,j}^{2,1} > 0\right]$ is an indicator function returning one if at least one requested method is assigned to the DXC.

Should LIH be performed by the AU, it is necessary to determine on which component. To do so, we exploit the knowledge of odd methods. For each sample, we compute the number of aliquots collected by the individual AU components. We denote this as an integer variable $n_{c,d,i}^{1,b}$ computed as

$$n_{c,d,i}^{1,b} = \sum_{j \in J_i} p_j x_{c,j}^{1,b},$$

where $n_{c,d,i}^{1,b}$ is the number of aliquots collected by the AU's component b for sample i w.r.t. chromosome c and p_j stands for the number of aliquot collections needed to perform method j (BIL and BILK require two collections). LIH are performed by the component with lower $n_{c,d,i}^{1,b}$, effectively lowering the number of occurred odd methods. Should it happen that $n_{c,d,i}^{1,1} = n_{c,d,i}^{1,2}$, the LIH decision is made randomly. The LIH decision for AU's component b is encoded by a binary variable $h_{c,d,i}^{1,b}$ equal to one if this component executes LIH.

The ISE methods are handled identically to the ILP model – the number of ISE performed by the DXC analyzer in day d is approximated from the data as a parameter ise_d. The DXC's component working time in day d w.r.t. chromosome c is equal to the number of aliquot collections plus all the LIH methods performed by the DXC plus the ise_d approximation multiplied by DXC's pipetting time, leading to an equality

$$w_{c,d}^{2,1} = pt^2 \left(\sum_{j \in J} p_j s_{d,j} x_{c,j}^{2,1} + \sum_{i \in I_d} (1 - h_{c,d,i}) + \; ise_d \right),$$

where pt^a is the pipetting time of analyzer a, and $s_{d,j}$ is the number of samples requesting method j in day d. This is identical to the calculation in the ILP model. The working time of AU's components is computed similarly with the addition of idle time caused by the odd methods and without ISE. Therefore, AU's working time is calculated as

$$w_{c,d}^{1,b} = pt^1 \left(\sum_{j \in J} p_j s_{d,j} x_{c,j}^{1,b} + \sum_{i \in I_d} h_{c,d,i}^{1,b} \right) + pt^1 \sum_{i \in I_d} \max\left(0, (n_{c,d,i}^{1,b'} + h_{c,d,i}^{1,b'}) - (n_{c,d,i}^{1,b} + h_{c,d,i}^{1,b})\right),$$

where b' represents the other AU's carousel. In comparison with the ILP model, the difference is the handling of odd methods computation. In the ILP model, the LIH methods are included in the $n_{c,d,i}^{1,b}$ value. However, in this case, the variable determines where LIH will be performed. Thus, the LIH methods have to be added to the computation separately. Following from the earlier throughput definition, the calculation

$$q_{c,d} \;=\; \frac{n_d}{\max(w_{c,d}^{1,1}, w_{c,d}^{1,2}, w_{c,d}^{2,1})} * 3600$$

computes the daily biochemistry throughput using the component working times, where $q_{c,d}$ is the biochemistry throughput and n_d is the number of samples in day d. The overall biochemistry throughput is equal to the average of the daily throughputs. Let q_c be the average biochemistry throughput of a chromosome c and D be the set of days. The overall throughput is

$$q_c \;=\; \sum_{d \in D} \frac{q_{c,d}}{\|D\|}.$$

The transports in the fitness function are considered via a penalty to the system throughput that is applied if the total number of transports exceeds the upper bound. Furthermore, we want to encourage the algorithm to prioritize solutions with a low number of exceeded transports. Thus, we penalize the solution for each additional transport above the upper bound. Let $t_{c,d,i}$ be an indicator variable for each sample i equal to one if the sample i is transported w.r.t. the chromosome’s assignment. The sample is transported if it requests at least one method assigned to the DXC and at least one method assigned to any AU component, excluding ISE and LIH methods. Once again, if a sample requests a long immunology method, we do not consider its transport. Let $st_{d,i}$ be a binary variable equal to one if sample i requests immunology method with the duration of at least 35 minutes and t_c be the average number of daily transports calculated as

$$t_c = \frac{1}{\|D\|} \sum_{d \in D} \sum_{i \in I_d} (1 - st_{d,i}) \cdot t_{c,d,i}.$$

Then, the fitness value e_c of chromosome c is computed as

$$e_c = q_c - \; r \cdot \mathbf{1}[t_c > \alpha] - s \cdot \max(0, t_c - \alpha\,),$$

where α is the upper bound of average daily transports, r is the initial penalty applied if t_c is greater than α, s represents the additional penalty for each transport above the upper bound, and $\mathbf{1}[t_c > \alpha]$ is an indicator function returning one if the amount of transports exceeds the upper bound.

The last constraint we haven't considered is the interferences of methods. The constraint is enforced similarly by imposing a penalty for every interfering pair of methods assigned to the same component. Let f_c be the number of such pairs assigned to the same biochemical component and v be a penalty applied for each interfering pair. Thus, the fitness value the algorithm maximizes is calculated as

$$e_c = q_c - \ r \cdot \mathbf{1}[t_c > \alpha] - s \cdot \max(0, t_c - \alpha\ ) - v \cdot f_c.$$

However, there is a potential danger when imposing penalties on the fitness function – the function might return a negative value. This may be troublesome for some selection strategies. Therefore, depending on the EA's implementation, it might be important to set the penalties to such values that won't allow a negative fitness value. The maximum possible penalty must be less than or equal to the minimal possible system throughput to guarantee this. We can achieve minimal system throughput by assigning all biochemical methods to the slowest component, which is the DXC, with 800 methods per hour.

In the worst-case scenario, we transport every sample in the optimization data. In such a case, the daily number of transported samples equals the daily number of samples. Let m be the daily average number of samples considered in the optimization data expressed as

$$m = \frac{||I||}{||D||}.$$

If we assume that the transportation limit α is less than m, then the maximum transportation penalty γ equals

$$\gamma = r + s(m - \alpha),$$

where $s(m - \alpha)$ represents the maximum penalty for each transport above the limit α, and r is the initial penalty for violating the transport limit. The maximum penalty for method interference occurs if both methods from each pair are assigned to the same component. The maximum interference penalty δ is

$$\delta = v \cdot ||F||,$$

where F is the set of interfering pairs and v is the penalty for each pair assigned to the same component. Let q_{min} be the minimal possible throughput. The fitness function is guaranteed to be non-negative if the sum of the maximum possible penalties is less than or equal to the minimum possible throughput expressed by inequality

$$q^{min} \geq \gamma + \delta.$$

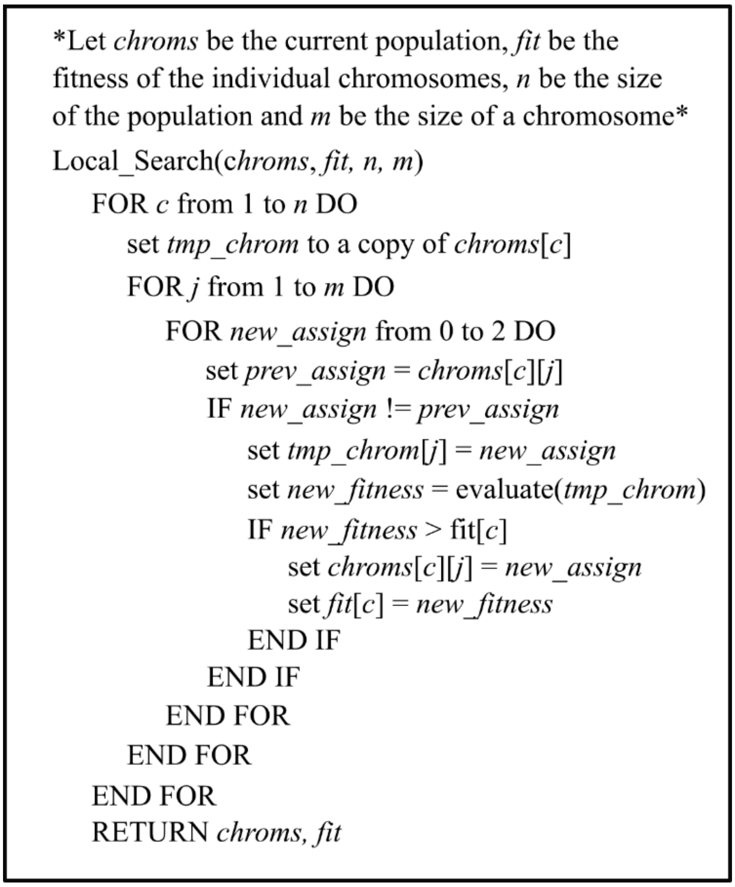
Let us consider an average amount of samples $m = 500$, transportation limit $\alpha = 100$, and $\|F\| = 10$. The minimum system throughput is equal to the DXC working rate, which is 800. Therefore, to guarantee non-negative fitness value, we need to find real numbers r, s and v that satisfy inequality

$$800 \geq r + s(500 - 100) + 10v = r + 400s + 10v.$$

which can be satisfied, for example, with $r = 200$, $s = 1$, $v = 15$. With these values of penalties, the maximum possible penalty is equal to 750, which is less than the minimum possible system throughput, and the fitness is guaranteed to be non-negative.

Note that we had to assign all methods to the DXC analyzer to achieve the minimal throughput. However, with such an assignment, no transports can occur. Thus, there is a certain tolerance to the values of penalties. An alternative method dealing with the negative fitness values is discarding all such chromosomes from the population and replacing them with new ones before applying selection.

Local Search



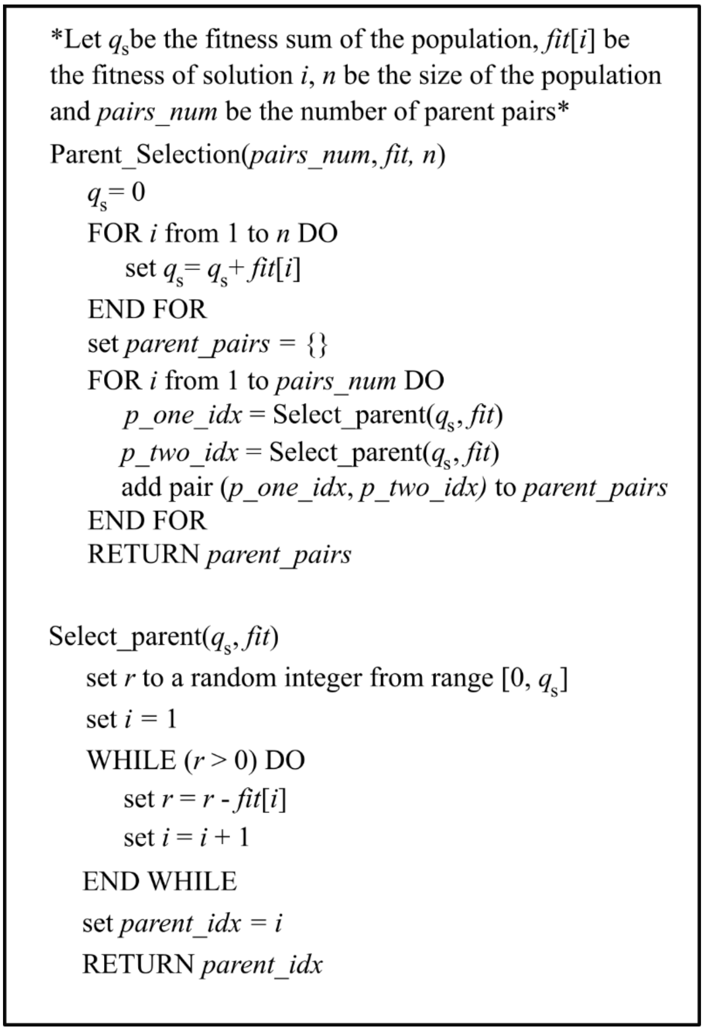
```
*Let chroms be the current population, fit be the
fitness of the individual chromosomes, n be the size
of the population and m be the size of a chromosome*
Local_Search(chroms, fit, n, m)
    FOR c from 1 to n DO
        set tmp_chrom to a copy of chroms[c]
        FOR j from 1 to m DO
            FOR new_assign from 0 to 2 DO
                set prev_assign = chroms[c][j]
                IF new_assign != prev_assign
                    set tmp_chrom[j] = new_assign
                    set new_fitness = evaluate(tmp_chrom)
                    IF new_fitness > fit[c]
                        set chroms[c][j] = new_assign
                        set fit[c] = new_fitness
                    END IF
                END IF
            END FOR
        END FOR
    END FOR
    RETURN chroms, fit
```

Figure 4.3 - Local Search Pseudocode

The local search is applied to every solution in the population. During the local search, the proposed evolutionary algorithm iterates over individual methods in the chromosome. Each method is experimentally assigned to the other components and evaluated with the fitness function. If the assignment change improved the fitness function, the newly acquired chromosome is kept. Consider a chromosome with method GLU assigned to the AU's inner carousel. Thus, the assignment value of the method GLU is 0. The local search creates a local copy of the chromosome and changes the GLU assignment in the copy to 1. The copy is evaluated with the fitness function. If its fitness is higher than the fitness of the original chromosome, the assignment change is applied to the original chromosome. The same process is repeated for GLU assignment equal to 2. Figure 4.3 shows the pseudocode of the local search process. Because we keep the modifications only if the fitness value increased, the local search can either improve the solution or the solution won’t change. Thus, it can only push the chromosome closer to the optimum but never farther from it.

Parent Selection

In our evolutionary strategy, we generate 35 new solutions using the crossover procedure. For each such chromosome, we select a pair of parents utilizing the roulette wheel algorithm [16]. The probability of a solution becoming a parent is proportional to its fitness function. Therefore, the fittest solutions have the highest probability, but even a solution with a low fitness can be selected. The parent selection pseudocode is depicted in Figure 4.4. First, we pre-compute the fitness sum of all solutions in the population. The goal is to generate *n* children. Thus we need *n* parent pairs. Each parent is selected w.r.t. a randomly generated number ranging from 0 to the fitness sum. Solutions in the population are sorted in descending order w.r.t. their fitness value. The parent selection iterates over all the solutions, subtracting the solution’s fitness from the generated number until it is less than or equal to zero. The index of a solution that lowers the random value to zero or below is returned. The same process is repeated for the index of the second parent, and together they form a single parent pair.



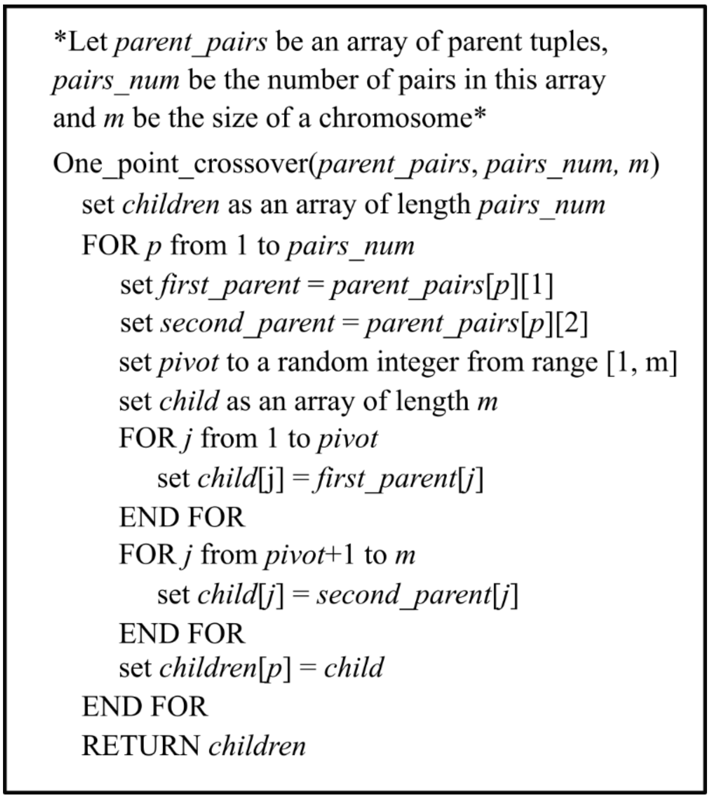
```
*Let $q_s$ be the fitness sum of the population, $fit[i]$ be
the fitness of solution $i$, $n$ be the size of the population
and $pairs\_num$ be the number of parent pairs*
Parent_Selection($pairs\_num$, $fit$, $n$)
    $q_s = 0$
    FOR $i$ from 1 to $n$ DO
        set $q_s = q_s + fit[i]$
    END FOR
    set $parent\_pairs = \{\}$
    FOR $i$ from 1 to $pairs\_num$ DO
        $p\_one\_idx$ = Select_parent($q_s$, $fit$)
        $p\_two\_idx$ = Select_parent($q_s$, $fit$)
        add pair ($p\_one\_idx$, $p\_two\_idx$) to $parent\_pairs$
    END FOR
    RETURN $parent\_pairs$

Select_parent($q_s$, $fit$)
    set $r$ to a random integer from range $[0, q_s]$
    set $i = 1$
    WHILE ($r > 0$) DO
        set $r = r - fit[i]$
        set $i = i + 1$
    END WHILE
    set $parent\_idx = i$
    RETURN $parent\_idx$
```

Figure 4.4 - Parent selection pseudocode

Crossover

The crossover algorithm iterates over all pairs of parents selected during the selection phase. To create a child, we generate a random pivot that splits the child's chromosome into two parts. The first part is inherited from the child's first parent and the second part is inherited from its second parent. Therefore, in the child's chromosome, the assignment of all methods with an index lower than or equal to the pivot is set to the assignment in the first parent's chromosome. Similarly, the assignment of methods with a higher index than the pivot is equivalent to their assignment in the second parent's chromosome. This crossover process is described by pseudocode in Figure 4.5.

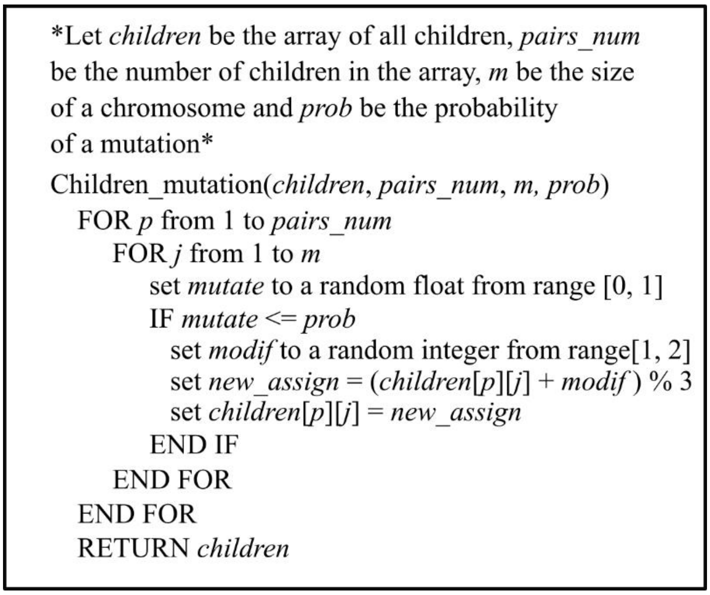


```
*Let parent_pairs be an array of parent tuples,
pairs_num be the number of pairs in this array
and m be the size of a chromosome*
One_point_crossover(parent_pairs, pairs_num, m)
  set children as an array of length pairs_num
  FOR p from 1 to pairs_num
    set first_parent = parent_pairs[p][1]
    set second_parent = parent_pairs[p][2]
    set pivot to a random integer from range [1, m]
    set child as an array of length m
    FOR j from 1 to pivot
      set child[j] = first_parent[j]
    END FOR
    FOR j from pivot+1 to m
      set child[j] = second_parent[j]
    END FOR
    set children[p] = child
  END FOR
  RETURN children
```

Figure 4.5 - Crossover pseudocode

Mutation

Each generated child might undergo a slight mutation of its chromosome. We iterate over each method's assignment and apply the mutation with a pre-defined probability for each child. The mutation reassigns the method to a different component, with a very low probability. Thus, a single child can mutate multiple assignments, but it may also happen that no assignment change has occurred. The described mutation process pseudocode is depicted in Figure 4.6.



```
*Let children be the array of all children, pairs_num
be the number of children in the array, m be the size
of a chromosome and prob be the probability
of a mutation*
Children_mutation(children, pairs_num, m, prob)
  FOR p from 1 to pairs_num
    FOR j from 1 to m
      set mutate to a random float from range [0, 1]
      IF mutate <= prob
        set modif to a random integer from range[1, 2]
        set new_assign = (children[p][j] + modif ) % 3
        set children[p][j] = new_assign
      END IF
    END FOR
  END FOR
  RETURN children
```

Figure 4.6 - Mutation pseudocode

Survivor Selection

In addition to the solutions from the previous generation and chromosomes created via the crossover procedure, we randomly generate five random solutions to maintain higher variability. All these solutions undergo the survivor selection that determines whether it is allowed to the next generation or not. The ten fittest solutions from the previous generation are allowed immediately. For the remainder of the population, we sort the remaining solutions according to their fitness value and select 40 chromosomes with the highest fitness.

Termination

The genetic algorithm is terminated, and the best solution is extracted after a certain amount of CPU time has passed. The reasoning behind this termination criterion is that we cannot exploit the knowledge of theoretical maximum system throughput because, in a practical setting, it is very difficult or even impossible to achieve. This is caused by the high variability in demand for methods. One method may be barely ever requested, whereas every second sample might require a different one. Thus, it is impossible to distribute the workload among the components perfectly. We could also enforce a fitness threshold and terminate the EA if the best fitness difference between two subsequent generations is below the threshold. However, it is a common occurrence for the best solution to not change for tens of generations. Thus, such termination criteria could prematurely terminate the algorithm, despite being able to further improve the best assignment.

5 Average TAT minimization

In the previous sections, we focused our attention on the system throughput maximization, and TAT was only considered as a side criterion. We want to explore further the effect of methods assignment on the sample average TAT, which is paramount for hospital laboratories. This section is a continuation of Karel Gavenčiak's research performed in his diploma thesis [17]. In his thesis, he analyzed operational data of a laboratory in Faculty Hospital Královské Vinohrady. He designed several rules deciding the assignment of individual samples in batches, resulting in a 21.7% statim TAT decrease in morning hours and a 36.6% decrease during the forenoon. In the following optimization, we will use anonymized data from Královské Vinohrady and apply it to the Prevedig's DxA5000 automatization system. In the remainder of this section, we will first define the average TAT minimization problem in Subsection 5.1. Afterward, we describe in Subsection 5.2 the proposed ILP model solving this optimization task.

5.1 Problem Statement and Criteria

In hospitals, laboratory systems operate with three different priorities of samples – Routine, Statim, and Vital. Hospital laboratories typically enforce requirements on the sample's TAT based on the sample's importance. In the case of Královské Vinohrady [17], 80% of samples with Statim priority must be examined within 60 minutes and 98.5% within 120 minutes after the sample's arrival. The tests for Vital priority must be carried out within 30 minutes for 80% of samples and 60 minutes for 95.5% of samples. Lastly, 80% of routines should be processed within five hours. However, these requirements are associated with laboratory TAT – a time interval between the sample's reception and reporting of tests results. This includes sample handling by the laboratory personnel and centrifugation. Instead, we consider analysis TAT that begins with the release of the sample from the centrifuge and ends with the results reporting. Note that the analysis TAT is included in the laboratory TAT. Therefore, lowering analysis TAT will also lower the laboratory TAT.

For this task, we consider the two most time-consuming types of analysis – biochemistry and immunology – since they are the main determinants of the sample's turn-around time. Let p_i^a be the processing time of sample i on analyzer a, representing the time needed by the analyzer to process all the requested methods. Sample's analysis TAT is determined by the last released result. Therefore, its TAT is equal to

$$TAT_i = \max_{\mathrm{a} \in \mathrm{A}} (p_i^a),$$

where TAT_i is the turn-around time of sample i and A is the set of biochemical and immunological analyzers. In this task, we no longer differentiate between AU's inner and outer carousel, and instead, we only consider the analyzer as a whole. Therefore, we only decide whether a biochemical method should be assigned to the DXC or the AU. The distribution among the AU's components can be done separately with another algorithm similar to those proposed earlier in Section 4.

Our goal is to find an assignment of methods that minimizes the average sample TAT. Therefore, the objective is expressed as

$$\min \frac{1}{\|I\|} \sum_{i \in I} TAT_i,$$

where I is the set of all considered samples. However, we want to satisfy the laboratory TAT requirements based on sample's priority. Let us consider a laboratory requirement c for priority p with TAT limit $\alpha_{p,c}$. Let $v_{i,p,c}$ be a binary variable indicating whether sample i with priority p violated requirement c. Thus, $v_{i,p,c}$ for sample i is equal to one if TAT_i is greater than requirement's limit $\alpha_{p,c}$, expressed as

$$v_{i,p,c} = \mathbf{1}\left[TAT_i > \alpha_{p,c}\right],$$

where $\mathbf{1}\left[TAT_i > \alpha_{p,c}\right]$ is an indicator function returning one if $TAT_i > \alpha_{p,c}$ holds. Let $q_{p,c}$ be the percentage of samples that should satisfy the required limit. The number of samples that violated the requirement, expressed as the sum of $v_{i,p,c}$ over all samples $i \in I_p$, are upper bounded by constraint

$$\sum_{i \in I_p} v_{i,p,c} \leq (1 - \alpha_{p,c}) * \|I_p\|,$$

where I_p is the set of samples with priority p and $(1 - \alpha_{p,c})$ expresses the percentage tolerance of laboratory requirement. Therefore, the right side of the inequality represents the maximum number of samples that are allowed to have their TAT higher than the limit.

Thus, we search for an assignment of methods to the analyzers that minimizes the average sample TAT while satisfying all the laboratory requirements. The problem's objective can be expressed as

$$\min \frac{1}{\|I\|} \sum_{i \in I} TAT_i$$

subjected to constraints

$$\begin{aligned} v_{i,p,c} &= \mathbf{1}\left[TAT_i > \alpha_{p,c}\right] & \forall p \in P, \forall c \in C_p, \forall i \in I_p, \\ \sum_{i \in I_p} v_{i,p,c} &\leq \left(1 - \alpha_{p,c}\right) * \|I_p\| & \forall p \in P, \forall c \in C_p, \end{aligned}$$

where P is the set of all priorities, C_p is the set of all considered TAT requirements associated with priority $p \in P$ and TAT_i is the TAT of sample $i \in I$ that is determined by the assignment of methods.

5.1.1 Constraints

Once again, we operate with biochemical analyzers. Thus, all the constraints described previously in Subsection 4.1.2 hold here as well. However, some of the constraints are slightly modified because we no longer differentiate between the AU's components. Odd methods cannot be considered and need to be solved individually with a specialized algorithm. The same applies to the limitation of BIL and BILK methods. The pairs of interfering methods cannot be both assigned to the DXC analyzer. However, it is possible to assign interfering pairs to the AU analyzer since it has two components. As for the LIH methods, we still need to decide whether the DXC or the AU will perform them. To our knowledge, the addition of the immunology analyzers doesn't introduce any new constraints. Similarly to biochemistry, each immunological method has to be assigned to exactly one analyzer. This time, there are no exceptions. ISE methods were performed by a specialized unit in the laboratory system of Královské Vinohrady. For our optimization, we utilized the biochemical and immunological sample data only. However, since a separate unit performed the ISE in the hospital laboratory, these methods are not included in the optimization data.

5.2 Integer Linear Program

In this section, we introduce the ILP approach designed to solve the defined optimization task. The remainder of this section has the following structure. Subsection 5.2.1 describes the structure and functionality of the ILP model in detail. Subsection 5.2.2 provides an overview of the model's sets, parameters, and decision variables. The mathematical representation of the model is available in Subsection 5.2.3.

5.2.1 ILP Model

The approach designed to solve this optimization task is based on integer linear programming. Similarly to the throughput maximization, the approach utilizes the laboratory operational data to adapt the solution to the time spectrum of the sample's arrivals. The optimization data is formatted identically as in the throughput maximization problem in Subsection 4.2.1. However, instead of being separated into individual days, the samples are distributed into 15 minutes long time intervals. Each time interval contains a set of samples that were released during the interval, and each sample is associated with a set of requested methods.

We are searching for an assignment of methods on the analyzers. Therefore, we define a binary decision variable x_j^a for each method and analyzer equal to one if method j is assigned to analyzer a. However, we operate with two different types of methods – immunological and biochemical – determining to which analyzers the methods can be assigned. For this reason, we forbid the assignment of biochemical methods to the immunology analyzers and vice versa by fixing the values of the particular variables to zero with constraints

$$\begin{aligned} x_j^{DXI1} = 0, x_j^{DXI2} = 0 \qquad & \forall j \in J^{bio}, \\ x_j^{DXC} = 0, x_j^{AU} = 0 \qquad & \forall j \in J^{immuno}. \end{aligned}$$

Each method, except LIH methods, has to be assigned to exactly one analyzer. To enforce this rule, we introduce a new constraint for each method that sums over the assignment variable of all the analyzers and puts it equal to one

$$\sum_{a\in A} x_j^a = 1 \quad \forall j \in J\backslash\{LIH\}.$$

The LIH methods are by default assigned to both biochemical analyzers, expressed by equalities

$$x_{LIH}^{AU} = 1,$$
$$x_{LIH}^{DXC} = 1.$$

The final constraint limiting the assignment of methods is the interfering pairs. As we have discussed earlier, the interfering methods cannot be assigned to an analyzer with only one analytic component. Thus, we define A^{II} as the set of all components with two reaction carousels and V as the set of interfering pairs. We introduce a new constraint for each analyzer with one component and each interfering pair, limiting the sum of the assignment variables to at most one. Thus, at most one of the interfering methods can be assigned to an analyzer with a single component, expressed as

$$x_j^a + x_{j'}^a \leq 1 \qquad \forall (j,j') \in V, \forall a \in A\backslash A^{II}.$$

To assess the TAT of a single sample, we need to determine two factors. The first is the route through the system the sample undertakes. If a sample requests methods assigned to the DXC and the DXI1 analyzers, there is no need to visit the DXI2 and the AU. Thus, we need to identify which route the sample should take. The second factor is the sample's TAT on the individual analyzers it has visited. Sample's TAT on an analyzer represents the time between the release of a sample from the input and the results reporting of all requested methods assigned to the analyzer.

The assignment of methods determines a sample's routing. Let L be the set of all permutations with repetition of values 0 and 1 and length 4 representing all 16 possible routes. Number 1 on position p indicates that the p-th analyzer will be visited. The analyzers have DXI1→DXI2→DXC→AU order. For example, $l = (0, 1, 1, 0)$ denotes a route l where analyzers DXI2 and DXC are visited. Let y_i^l be a binary variable equal to one if sample i undertakes route l. Thus, for each sample, exactly one route has to be selected. This is ensured by constraint

$$\sum_{l\in L} y_i^l = 1 \qquad \forall h \in H, \forall i \in I_h,$$

where H is the set of time intervals into which the samples are separated and I_h is the set of samples in the time interval $h \in H$. However, we haven't determined which routes the sample can take. The sample has to visit all the analyzers with at least one sample's method. Thus, we introduce a new constraint for each sample i and each analyzer a forcing the model to select one of the routes containing the analyzer on which the sample's methods are assigned. Let M be a big-M parameter – a value high enough to guarantee the constraint's satisfaction given a specific condition – and L_a be the set of all routes containing analyzer a. Consider constraint

$$\left(\sum_{l \in L_a} y_i^l\right) * M \geq \sum_{j \in J_i \setminus \{LIH\}} x_j^a \qquad \forall h \in H, \forall i \in I_h, \forall a \in A.$$

On the right side, we sum the assignment variable x_j^a over each method requested by sample i. If at least one of the requested methods is assigned to analyzer a, the sum is a positive number. On the left side, we sum the route selection variable y_i^l over all the routes containing analyzer a. If the right side is greater than zero, then at least one of the y_i^l variables containing analyzer a has to be one to satisfy the inequality. This constraint is created for each analyzer a. Therefore, if a sample requires methods on the DXI1 and AU analyzers, the model must select precisely one route containing both of these analyzers. Naturally, the model is allowed to select routes where the sample unnecessarily visits other analyzers, such as $l = (1,1,1,1)$. However, choosing such a route would increase the sample's TAT, and because of the minimization nature of the task, the model will not choose such a route. The TAT computation will be presented shortly.

There is one more consideration to be made regarding the route selection. The LIH methods are assigned to both biochemical analyzers by default. Therefore, according to the previous constraint, a sample would need to visit both the DXC and the AU if it requests the LIH methods. To avoid this, the LIH are excluded in the previous constraint when summing over the sample's methods. However, samples requesting only LIH biochemical analysis would no longer visit the biochemistry. To solve this issue, we define a binary parameter LIH_i that is equal to one if a sample i requests LIH analysis. Let L_{BIO} be the set of routes containing any biochemical analyzer. We add a new constraint for each sample requesting LIH that sums route selection variables y_i^l over all routes in L_{BIO} and puts it equal to one. Thus, if a sample requests LIH methods, it needs to undertake a route visiting at least one biochemical analyzer, expressed as

$$\left(\sum_{l \in L_{BIO}} y_i^l\right) = 1 \qquad \forall h \in H, \forall i \in I_h : LIH_i = 1.$$

When a sample reaches the analyzer, it may need to wait in a queue and undergo a series of manipulations before the sample can be processed. Recall the AU's sample processing is described in Subsection 2.3.1. Although the sample manipulations allow the analyzer to process a high amount of samples, it comes at the cost of additional delays. On average, the delay caused by the sample's manipulation is 5 minutes. The DXC analyzer collects aliquots directly from the transportation track.

However, the analyzer operates in two states – an active state and an idle state. If the analyzer is in the idle state and a sample arrives, the DXC needs to switch to the active state. The DXC changes to the idle state if no sample has arrived for some time. This represents another layer of complexity because the delays on the DXC analyzer depend on the number of samples the analyzer processes. If the analyzer performs methods for a few samples, it is very likely the DXC will be in the idle state when the sample arrives, which will result in a processing delay. On the other hand, if there are many samples available at the DXC analyzer, the analyzer will always be active. Still, the samples will have to wait in a queue until all preceding tubes are processed.

According to the queueing theory [18, 19], an analyzer can be viewed as an M/M/1 queueing model – a system with one serving unit, exponential service time, and arrivals determined by the Poisson process [18]. The analyzer is associated with a service rate, representing the number of tests it can process over time. The theoretical DXC's service rate is 800 tests per hour. As long as the test arrival rate is below the analyzer's service rate, the sample's average queue waiting times are negligible. Naturally, this depends on the granularity of the time intervals. If the DXC analyzer performs 600 tests over an hour, the arrival rate is lower than the analyzer's service rate, and the average waiting time is close to zero. However, it may happen that 500 out of the 600 tests arrived over 30 minutes, but the DXC's 30-minute service rate is 400 tests. Thus, the DXC will be overwhelmed, and the average sample waiting time will be high for the 30 minutes long interval.

For this reason, we have decided to distribute the samples into 15-minutes long intervals. According to our data analysis, there is no time interval with enough tests to exceed the service rates of AU, DXI1, or DXI2 analyzers. However, it is possible to overload the DXC. Therefore, we enforce a constraint limiting the number of tests performed by the DXC analyzer in the particular time intervals to be lower than the DXC's 15-minute service rate. Let $f_{h,dxc}$ be an integer variable representing the number of tests processed by the DXC analyzer in time interval h. This value is equal to the sum of the assignment variables x_j^{dxc} multiplied by the number of samples requesting the method j in time interval h, expressed as a parameter $s_{h,j}$, over all methods J. However, this summation also includes LIH methods, and as a result, all LIH would be included in this calculation. Thus, we subtract LIH methods of all the samples in the time interval h taking a route that doesn't have the DXC analyzers. The resulting constraint is

$$f_{h,dxc} = \sum_{j \in J} (s_{h,j} \cdot x_j^{dxc}) - \sum_{i \in I_h} (LIH_i \cdot \sum_{l \in L_{A \setminus \{DXC\}}} y_i^l) \qquad \forall h \in H.$$

In the optimization, we consider sample data from Královské Vinohrady and apply it to the DxA5000 system in Prevedig. However, the advantage of the DxA5000 is its sample scheduler and route planner. In the operational data from Královské Vinohrady, the amount of new requested tests is very skewed among the individual time intervals, as shown in Figure 5.1. The figure depicts the number of arrival tests in particular time intervals. We can observe a significant difference in arrival rate between 7:00 and 7:15, where the number of new tests is nearly three times lower in 7:15. Such skewness could cause problems in the optimization because the DXC could easily get overloaded with the abnormally high arrival rate at 7:00. Moreover, the DxA5000 scheduler would consider this and schedule the sample's releases so that the DXC would not be overloaded. Thus, the arrival rate would be more balanced among the individual time intervals. To incorporate this idea into the ILP model, we increased the test arrival rate upper bound of the DXC analyzer for individual intervals. Furthermore, we also introduced an additional binary variable o_h for each time interval h that allows increasing the limit of the DXC to reduce the impact of the outliers with abnormally high service rate on the final solution. Variable o_h is equal to one if the DXC upper bound should be increased in the time interval h. Let γ be the increase of the DXC limit and β be the maximum number of allowed increases. The constraints

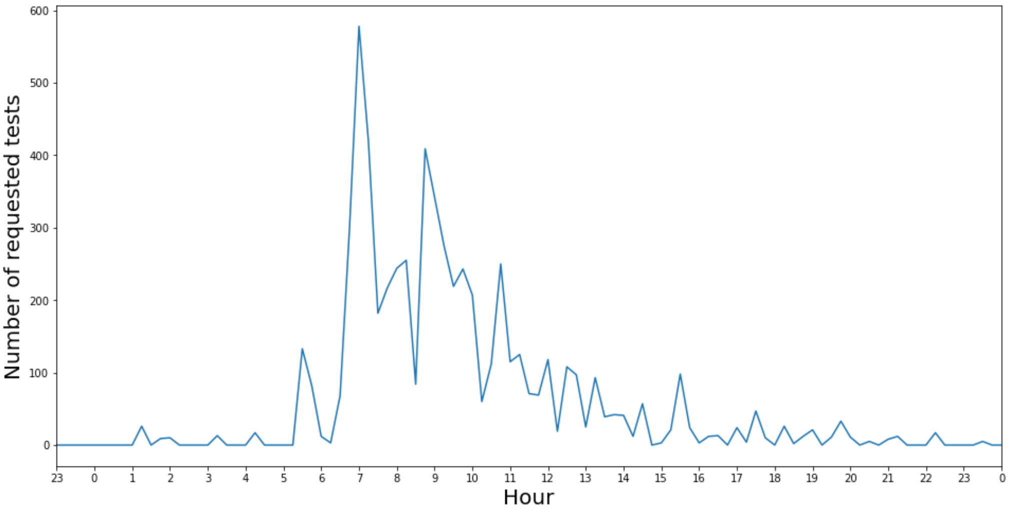


Figure 5.1 - Test requests in Královské Vinohrady

$$f_{h,dxc} \leq k^{dxc} + \gamma \cdot o_h \quad \forall h \in H,$$

$$\sum_{h \in H} o_h \leq \beta,$$

limit the DXC load. With routes selected for each sample and average queue waiting time under control, we can now calculate the sample's TAT on the individual analyzers. There are multiple factors we consider when computing the TAT. First, the sample needs to be transported to the analyzer. Thus, we estimated a parameter t^a from the Prevedig's operational data equal to the transportation time from the system's input to the analyzer a. Another considered aspect is the processing delays caused by the functionality of individual analyzers. Because queue waiting times are now negligible, we estimate the processing delays on the analyzers from the operational data as a constant parameter w^a. Suppose that a sample needs to visits analyzers DXI1, DXC, and AU. Before the sample arrives at the AU analyzer, it visits the DXI1 and the DXC. The sample was subjected to delays at the DXI1 and DXC, which further delays the sample's TAT at the AU analyzer. Therefore, the analyzer's total delay is computed as the analyzer's delay plus the delays on all the previously visited analyzers on the route, leading to

$$\sum_{a' \leq a} w_{a'} b^{l,a'},$$

where $b^{l,a}$ is a Binary parameter equal to one if analyzer a is present on route l, a is the index of the current analyzer.

The analyzer needs to collect a number of aliquots to perform the requested methods. The pipetting time needed to perform all the collections is another considered factor. It is computed as the sum of method assignment variables x_j^a multiplied by the number of needed aliquots p_j to perform method j and the pipetting time pt^a of analyzer a over all the methods requested by sample i, expressed as

$$\sum_{j \in J_i} x_j^a p_j pt^a.$$

After the aliquots are collected and reactions initiated, the analyzer waits until the chemical reaction is completed. The length of the responses depends on the methods. The sample's TAT is determined by the method with the longest reaction time. Therefore, the analysis duration is

$$\max_{j \in J_i} \left(x_j^a d_j^a \right),$$

where d_j^a is the reaction time of method j on analyzer a. Lastly, the TAT computation on analyzer a is meaningful only if the analyzer is visited on the route taken by sample i. With all these factors added together, the constraint

$$TAT_i^a \geq t^a + \textstyle\sum_{a' \leq a} w_{a'} b^{l,a'} + \textstyle\sum_{j \in J_i} x_j^a p_j pt^a + \max_{j \in J_i} \left(x_j^a d_j^a \right) - M(1 - y_i^l)$$
$$\forall h \in H, \forall i \in I_h, \forall a \in A, \forall l \in L$$

calculates the sample's TAT on analyzer a, expressed as TAT_i^a. Note that this constraint is created for each sample, each analyzer, and each route. If analyzer a isn't present on route l, the big-M is subtracted from the TAT value, and the constraint is always satisfied. Only one route can be selected for each sample. Therefore, only one of these constraints will be active for each analyzer, and the remaining 15 will always be satisfied. The total sample's TAT equals

$$TAT_i = \max_{a \in A}(TAT_i^a) \qquad \forall h \in H, \forall i \in I_h,$$

where TAT_i is the total turn-around time of sample i.

Lastly, we need to consider the hospital TAT requirements. Let $r_{i,p,c}$ be a binary decision variable equal to one if sample i violates the requirement c associated with priority p. Let $q_{p,c}$ be the TAT limit of the requirement c and $\alpha_{p,c}$ be the percentage of the samples that have to fulfill the requirement. For each sample, we determine whether it met the requirement with constraint

$$TAT_i \leq q_{p,c} + (r_{i,p,c} * M) \qquad \forall p \in P, \forall i \in I_p, \forall c \in C_p$$

created for every priority p, every requirement c and each sample with priority p.

If the sample's TAT is greater than the limit, then the constraint can be satisfied only if $r_{i,p,c}$ is equal to one. Then, for each requirement, we sum $r_{i,p,c}$ over all the samples with priority p and upper bound it by the total number of samples with such priority multiplied by the tolerance of the requirement, expressed by constraint

$$\sum\nolimits_{I_p} r_{i,p,c} \leq (1 - \alpha_{p,c}) * |I_p| \qquad \forall p \in P, \forall c \in C_p.$$

All that remains is to define the optimization criterion of the ILP model. The goal of the task is to minimize the average sample TAT. Thus, the model's criterion is the sum of TAT_i over all the samples divided by the total number of considered samples, expressed as

$$min \frac{1}{|I|} \sum_{h \in H} \sum_{i \in I_h} TAT_i.$$

The mathematical representation of the model can be seen in Subsection 5.2.3, and an overview of all the used sets, parameters, and variable is available in Subsection 5.2.2.

5.2.2 Overview of ILP Sets, Parameters and Variables

Sets

H – Set of time interval containing individual samples

I – Set of all samples. I_h denotes samples in time interval h and I_p stands for the set of samples with priority p.

J – The set of all methods. J_i represents the methods requested by sample i. The set includes LIH methods. J^{bio} stands for all biochemical methods and J^{immuno} is the set of immunological methods.

A – Set of all analyzers DXI1, DXI2, DXC, AU in this order. A^{II} denotes analyzers with two components, which is only the AU.

V – The set of interfering pairs of methods

L – Set of all considered routes. L_a is the set of all routes containing analyzer a and L_{BIO} is the set of all routes with at least one biochemical analyzer.

P – The set of priorities – Routine, Statim and Vital.

C_p – The set of all requirements associated with priority p. The requirement is a pair of a TAT limit and a percentage of samples that should satisfy the limit.

Parameters

t^a – Time needed to transport a sample from the automatization input to the analyzer a.

w^a – Estimation of average sample waiting time at the analyzer a.

$b^{l,a}$ – Binary parameter equal one if analyzer a is present on the route l.

p_j – Number of aliquot collections needed to executed method j.

pt^a – The pipetting time of the analyzer a.

d_j^a – Reaction duration of the method j on the analyzer a.

$s_{h,j}$ – The number of samples requesting method j in time interval h.

LIH_i – A binary parameter equal to one if sample i requests LIH methods.

k^{dxc} – The upper bound on the number of methods executed by the DXC analyzer in any time interval.

β – The maximum number of allowed increases of the DXC load upper bound.

γ – The amount by which the k^{dxc} the upper bound can be increased for a time intervals.

M – Big M. A parameter with high enough value guaranteeing constraint satisfaction if a specific condition is met.

$\alpha_{p,c}$ – The percentage of requirement c associated with priority p.

$q_{p,c}$ – The TAT limit of requirement c associated with priority p.

Decision Variables

TAT_i – A continuous variable describing the turn-around time of the sample i.

TAT_i^a – TAT of sample the i on the analyzer a.

x_j^a – A binary variable encoding method assignment. If the method j is assigned to the analyzer a the variable is equal to one.

y_i^l – A binary variable that represents route selection of a sample. Equal to one if sample i takes the route l.

f_h^{dxc} – An integer variable describing the number of tests processed by the DXC analyzer in the time interval h.

o_h - A binary variable equal to one if the DXC limit should be increased in the interval h. This variable is utilized to limit the impact of intervals with an abnormal number of requested tests.

$r_{i,p,c}$ – A Binary variable encoding whether the sample i satisfied the requirement c associated with the priority p or not. If the sample didn't fulfill the requirement, the variable is equal to one.

5.2.3 ILP Mathematical Representation

$$\min \frac{1}{|I|} \sum_{H} \sum_{I_h} TAT_i$$

s.t

$$TAT_i = \max_{\mathrm{a}}(TAT_i^a) \qquad \forall h \in H, \forall i \in I_h$$

$$TAT_i^a \geq t^a + \textstyle\sum_{a' \leq a} w_{a'} b^{l,a'} + \sum_{J_i} x_j^a p_j pt^a + \max_{J_i}\left(x_j^a d_j^a\right) - M(1 - y_i^l)$$

$$\forall h \in H, \forall i \in I_h, \forall a \in A, \forall l \in L$$

$$f_{h,dxc} = \textstyle\sum_J (s_{h,j} \cdot x_j^{dxc}) - \sum_{I_h} (LIH_i \cdot \sum_{L_{A \backslash \{DXC\}}} y_i^l) \quad \forall h \in H$$

$$f_{h,dxc} \leq k^{dxc} + 70 \cdot o_h \qquad \forall h \in H$$

$$\textstyle\sum_H o_h \leq \beta$$

$$\left(\textstyle\sum_{L_a} y_i^l\right) * M \geq \sum_{J_i \backslash \{LIH\}} x_j^a \qquad \forall h \in H, \forall i \in I_h, \forall a \in A$$

$$\left(\textstyle\sum_{L_{BIO}} y_i^l\right) = 1 \qquad \forall h \in H, \forall i \in I_h : LIH_i = 1$$

$$\textstyle\sum_L y_i^l = 1 \qquad \forall h \in H, \forall i \in I_h$$

$$\textstyle\sum_A x_j^a = 1 \qquad \forall j \in J \backslash \{LIH\}$$

$$x_j^{DXI1} = 0, x_j^{DXI2} = 0 \qquad \forall j \in J^{bio}$$

$$x_j^{DXC} = 0, x_j^{AU} = 0 \qquad \forall j \in J^{immuno}$$

$$x_{LIH}^{AU} = 1$$

$$x_{LIH}^{DXC} = 1$$

$$x_j^a + x_{j'}^a \leq 1 \qquad \forall (j, j') \in V, \forall a \in A \backslash A^{II}$$

$$\textstyle\sum_{I_p} r_{i,p,c} \leq (1 - \alpha_{p,c}) * |I_p| \qquad \forall p \in P, \forall c \in C_p$$

$$TAT_i \leq q_{p,c} + (r_{i,p,c} * M) \qquad \forall p \in P, \forall i \in I_p, \forall c \in C_p$$

6 Experimental Results

This section presents the results of experiments performed with the proposed approaches. First, we describe the tools and computational system utilized to perform the experiments in Subsection 6.1. Afterward, we present the results for individual optimization problems, starting with throughput maximization in Subsection 6.2, and followed by the average TAT minimization in Subsection 6.3. The throughput maximization subsection is further decomposed into two subsections – the experiments in 6.2.1 and the comparison of optimization techniques in 6.2.2.

6.1 Experimental Setup

The experiments were performed utilizing the computational cluster of the Czech Institute of Informatics, Robotics, and Cybernetics [20]. Specifically, we used the compute nodes with 256 or 512 GB RAM and Intel Xeon E5-2690 v4 CPUs with 14 cores per CPU and 2.6 GHz processor frequency. For each performed experiment, we reserved 64 GB of RAM and 12 cores. Both ILP models are solved using the Gurobi Solver [21], and their mathematical representations are implemented via Python. The genetic algorithm was implemented in the C++ programming language, using the standard libraries and the JsonCpp library [23] for JSON data manipulation. The optimization process of all the proposed approaches is based on the laboratory operational data. The data were analyzed and prepared for optimization using pandas data analysis tool [22] in a Jupyter Notebook. For the ILP models, the optimization data is stored as a python dictionary object in a Pickle file. The information for the evolutionary algorithm is stored and loaded via files in JSON format.

6.2 Biochemistry throughput Maximization

In this section, we first present the experiments performed with the ILP optimization approach in Subsection 6.2.1 and compare the results with the original Prevedig configuration. Afterward, we compare the results of the ILP and the genetic algorithm in Subsection 6.2.2.

6.2.1 Experiments

For the experiments, we split the laboratory operational data into two sets. An optimization set containing data from 5 days, one for each workday, and an evaluation set consisting of the remaining data to evaluate the resulting configurations. For each configuration, we compute the utilization of the individual components, their biochemistry throughput, the number of occurred odd methods, and the amount of transported samples. The resulting assignments are compared with the original Prevedig's configuration. We present the results of two particular experiments. The first experiment does not impose any limitation on the number of transported samples. The second experiment limits the number of average daily transports to 40, which is very similar to the number of transports that have occurred in the Prevedig's configuration. In this subsection, we present and compare the results of the ILP model.

Experiment without Transportation Limit

In this experiment, we set the parameter α, denoting the allowed number of transports, to infinity. Therefore, the transportation side criterion does not affect the optimization process, and the ILP can focus on the maximum component working time minimization. Figure 6.1 depicts the average daily utilization of biochemical components of Prevedig's configuration according to the evaluation data set. The x-axis denotes individual hours throughout a day, where any hour "HH" represents the time interval from HH:00:00 to HH:59:59. The y-axis represents the average component utilization in the given time interval. Thus, one can observe that most tests are performed from 11:00 to 11:59. The three lines each represent individual components. The blue line described the utilization of the AU's inner carousel (AU1). The orange line stands for the AU's outer carousel (AU2), and the green line represents the DXC component. According to the plot, the workload of the AU's components is reasonably balanced, with about a 3% difference in the peak interval. However, the DXC utilization is less than one-third compared to any of the AU's components. The throughput of the Prevedig configuration is 1905.14.

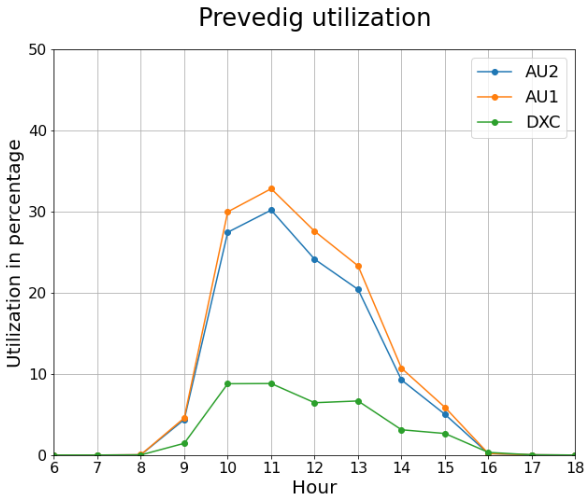


Figure 6.1 - Prevedig average machine utilization

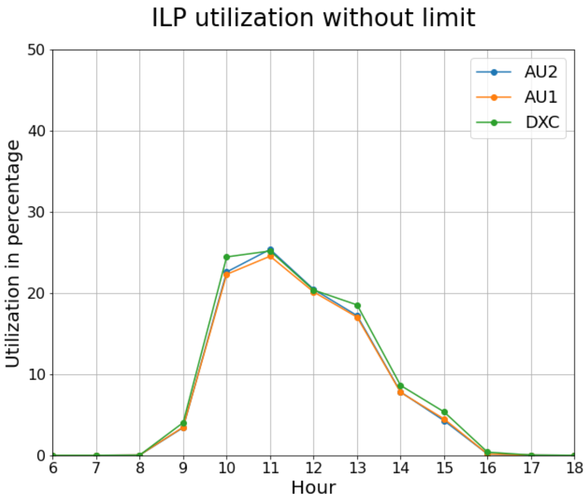


Figure 6.2 - unlimited ILP average machine utilization

Figure 6.2 shows the average component utilization of the configuration found by the ILP model. The utilization of the AU's components decreased from 32% to 24.5% for the AU1 and 30% to 25% for the AU2. However, the DXC utilization significantly increased from 9% to 25%, achieving nearly uniform workload distribution among the components. Note that the optimization was performed on different data than the evaluation. Thus, the optimization result can be applied to unknown data with promising performance. The resulting throughput is 2535.83 samples per hour, representing a 33.10% increase compared to the Prevedig configuration with 1905.14 samples per hour.

The increase in the system throughput is caused by more uniform utilization leveling and a significant decrease in occurrences of odd methods, as demonstrated in Figure 6.3, depicting odd occurrences with the Prevedig configuration, and Figure 6.4 representing the ILP solution. Earlier, we have defined component utilization as the percentage ratio of a component's active time to its availability time. Therefore, the idle time caused by the odd methods is not reflected in the machine utilization. However, it does impact the biochemistry throughput. The figures depict bar charts of average odd methods occurrences in respective hours. The x-axis denotes time intervals similarly to the previous figures. The y-axis represents the average number of occurred odd pipetting. The percentage above each bar indicates the ratio between the odd pipetting occurrences and the total number of requested tests. For example, in the Prevedig configuration, 106 skipped pipetting cycles have occurred during the eleventh hour. Thus, 15.5% of all the performed tests caused a skipped cycle. The average daily number of the odd methods is 425.86 for the Prevedig configuration. The configuration found by the ILP model significantly lowered the occurrences of odd methods in individual hours. The daily average number of odd pipetting decreased to 174.86, resulting in a 58.94% improvement.

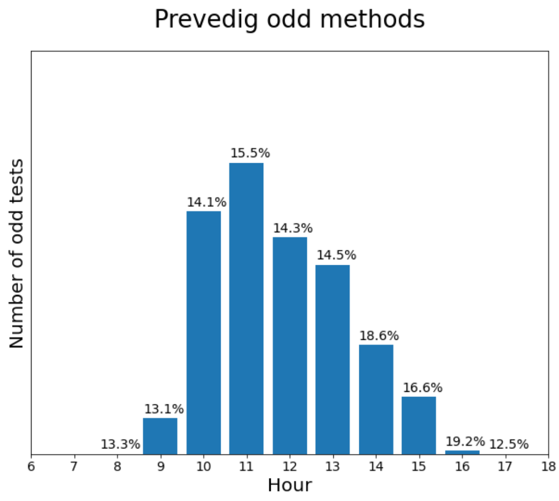


Figure 6.3 – Prevedig average odd methods occurrences

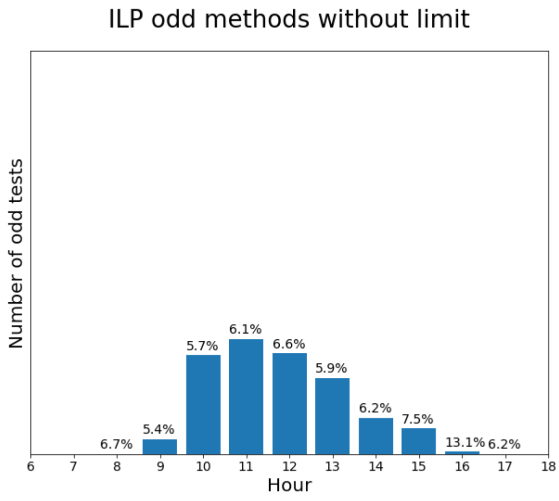


Figure 6.4 - unlimited ILP average odd occurrences

However, moving methods from the AU's component to the DXC will inevitably lead to a higher amount of necessary transports between the biochemical analyzers. The bar chart in Figure 6.5 (for Prevedig) and Figure 6.6 (for the ILP) have a similar structure as the bar charts of the odd methods. The y-axis represents the number of transported samples, and the percentage above the bars describes the ratio between transported samples and their total number. The advantage of the Prevedig configuration is its low amount of transports, with 15.5% transported samples in peak hour and 63.0 average daily transports. This is caused by the low number of methods assigned to the DXC analyzer. In the ILP configuration, nearly every second sample needs to be transported to be fully processed. With the ILP configuration, the average daily amount of transports is equal to 174.0, which is a 176.20% increase compared to the Prevedig configuration. This will likely increase TAT of certain samples. However, it is difficult to determine whether the increase in the number of transports would have a high enough impact on sample TAT to offset the significant improvement in system throughput. A process simulation, which is currently not at our disposal, would be necessary to accurately assess whether the sample TAT improved or not.

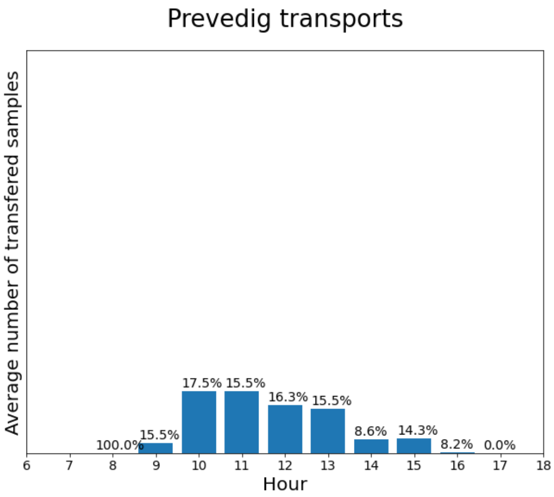


Figure 6.5 - Prevedig average daily sample transports

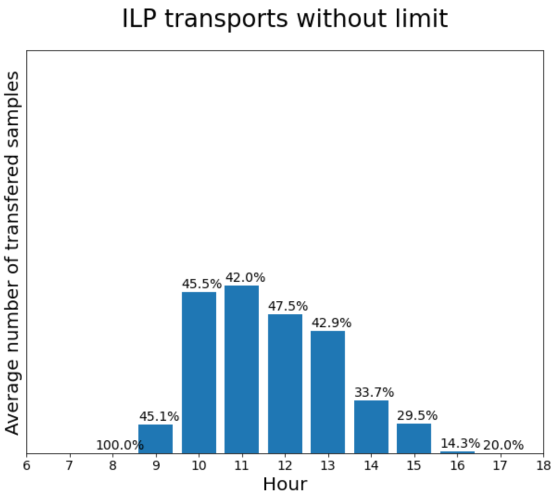


Figure 6.6 – unlimited ILP average daily sample transports

Experiment with Transportation Limit

In this experiment, we set the transportation upper bound α to a value that would preserve the transportation qualities of the Prevedig configuration. We carried out multiple experiments with different α settings and selected the value for which the amount of occurred transports was the closest to Prevedig's configuration. The results presented here were found by the model with α set to 40.

Figure 6.7 depicts the Prevedig component utilization, and Figure 6.8 shows the component utilization of the ILP method assignment. The figures have identical structures to the ones shown in the previous experiment. There is a little difference between the two plots. The ILP achieved a slightly better balance between the AU's components, but the DXC utilization hasn't changed. Therefore, in terms of utilization, nothing is different. However, the throughput of the Prevedig configuration is 1905.14 samples per hour. But the throughput of the ILP configuration is 1999.50 samples per hour, achieving a 4.95% improvement.

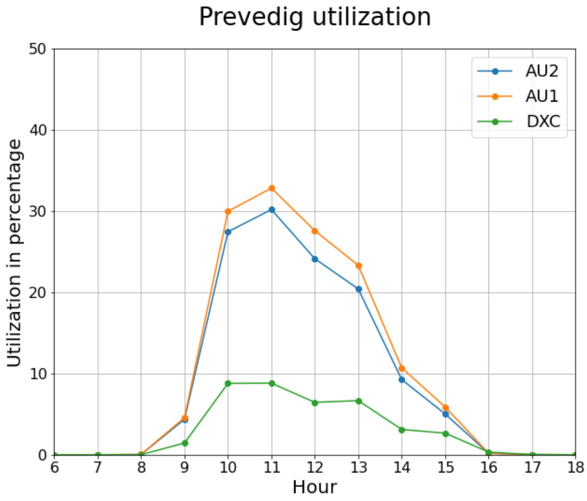


Figure 6.7 - Prevedig average daily utilization

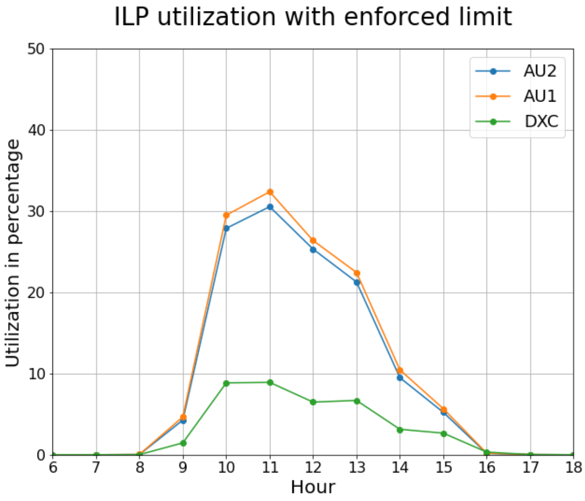


Figure 6.8 - Limited ILP average daily utilization

As we mentioned earlier, the utilization value is not affected by idle time. However, the system throughput is. Although the machine utilization barely changed, the number of occurred odd methods has significantly decreased, as is demonstrated by Figure 6.9 for Prevedig and Figure 6.10 for the ILP. We can observe a decrease in the number of odd pipetting with a 5.0% difference in the peak hour. The average daily number of odd occurrences improved by 32.92%, decreasing from 425.86 to 285.67. This result demonstrates the impact of odd methods on system performance and the importance of such complex criteria.

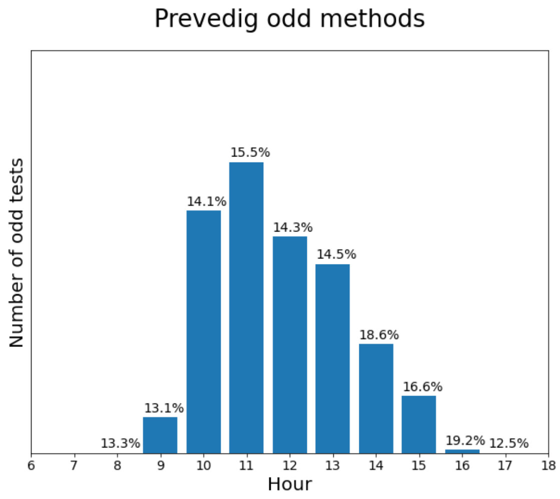


Figure 6.9 - Prevedig odd methods occurrences

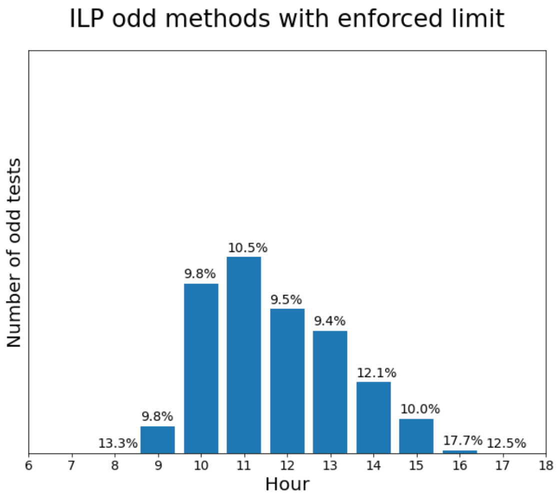


Figure 6.10 - limited ILP odd methods occurrences

Lastly, let us discuss the sample transports. The ILP transportation results are depicted in Figure 6.12 and the transports of the original configuration are in Figure 6.11. We can observe that the number of transports is precisely the same. This is expected as we have chosen α that enforces a similar number of transports. Therefore, the ILP model was able to find a configuration that preserves the advantage of the original assignment and still improves the biochemistry system throughput. An overview of individual criteria and the genetic algorithm results are available in the next section in Table 6.1.

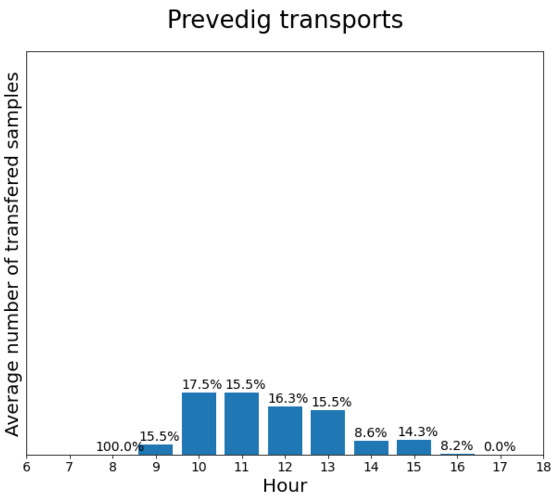


Figure 6.11 - Prevedig average daily transports

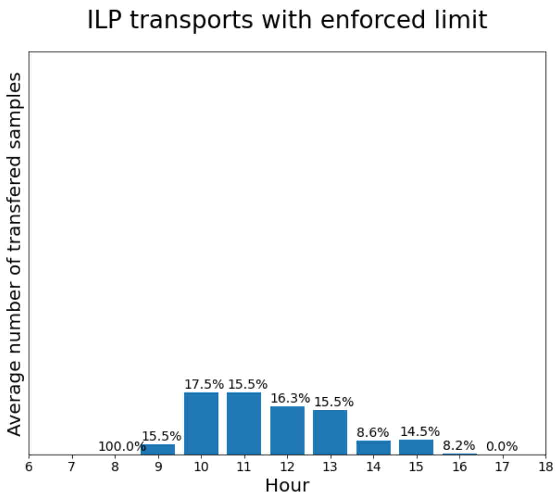


Figure 6.12 - limited ILP average daily transports

6.2.2 Methods Comparison

The same experiments presented in the previous section were also carried out with the evolutionary algorithm. The resulting configurations of the evolutionary algorithm were evaluated with the same evaluation process. Figures 6.13 and 6.14 demonstrate the utilization difference with unlimited transports between the ILP model and GA, respectively. The difference between the configurations is barely perceivable. The situation is similar in Figures 6.15 and 6.16 that present the utilization plots of the experiments with daily transports limited to 40. The DXC utilization seems to be identical, and the only difference is a slightly higher disbalance between AU1 and AU2 in the configuration obtained by the GA. The numerical evaluations of all the conducted experiments are available in Table 6.1, along with the Prevedig configuration for comparison.

		Unlimited Experiments		Limited Experiments	
	Prevedig	ILP	GA	ILP	GA
Throughput	1905.14	2535.83	2534.37	1900.83	1903.84
Odd methods	425.86	174.86	180.48	285.67	282.20
Transports	63.0	174.05	173.43	63.0	63.0

Table 6.1 - Criterion overview of all configurations

In the experiment without transportation limit, the ILP found a slightly better configuration in both system throughput and odd occurrences. However, the differences are negligible. The same holds for the limited experiments. However, the situation is reversed. The configuration of the GA performs better in both system throughput and odd occurrences compared to the ILP model. Therefore, the decrease in the odd occurrences was influential enough to offset the higher maximum component working time and achieve better system throughput.

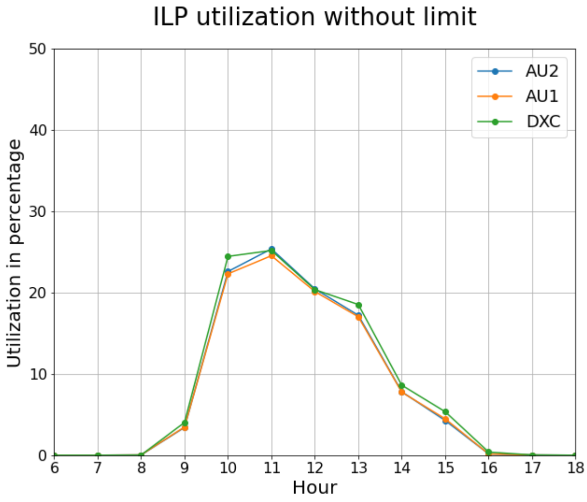


Figure 6.13 - Unlimited ILP utilization

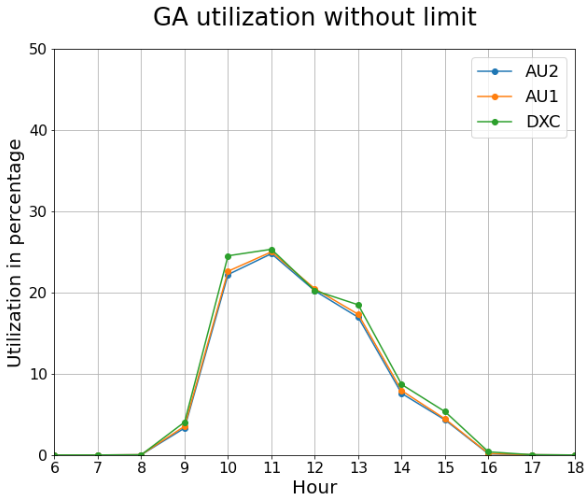


Figure 6.14 - Unlimited GA utilization

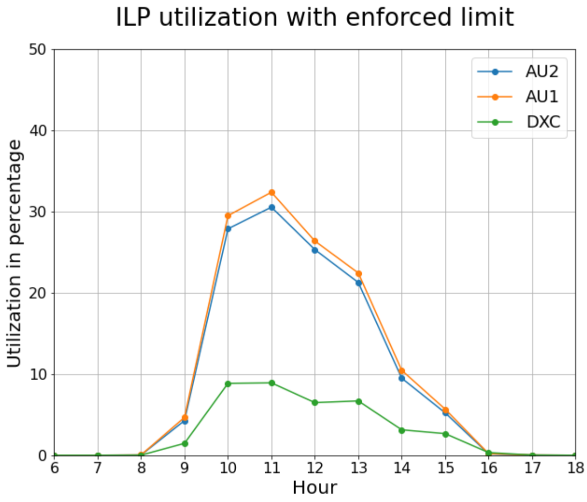


Figure 6.15 - Limited ILP utilization

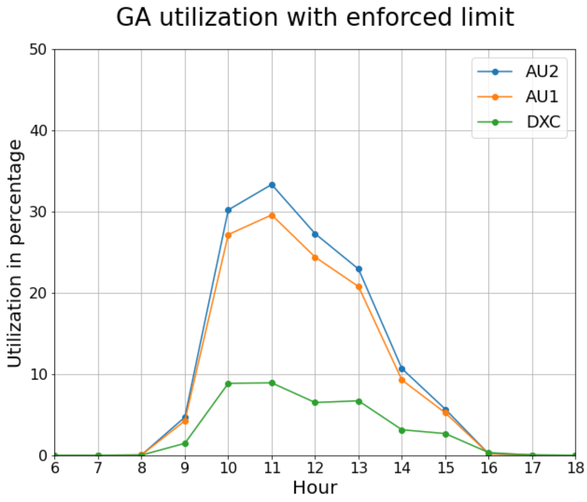


Figure 6.16 - Limited GA utilization

In terms of transports, the difference is even lower. The number of transports in the limited experiment is identical for both optimization approaches and even the Prevedig configuration.

To further explore the differences between the designed approaches, we have conducted several experiments with optimization datasets of various sizes. Each dataset has a predefined size that determines the number of randomly selected days contained within. We have generated datasets with sizes from 1 to 18. The same 13-hour timeout limits both approaches, and the best-achieved solutions are extracted. This experiment aims to determine whether the optimization strength of the individual approaches is comparable with the increasing number of considered samples. Individual solutions are evaluated w.r.t. the used optimization set. In these experiments, the upper bound of the average daily transports is equal to 50 for both algorithms. The configuration quality is evaluated by the biochemistry system throughput.

Figure 6.17 captures the outcome of this experiment. The x-axis denotes the size of an optimization set, and the y-axis represents the throughput of the best-achieved configuration. The throughput value itself is not important, as it highly depends on the days within the optimization set. We are interested in the difference between the individual lines, where the blue line represents the GA, and the red line stands for the ILP model. The highest difference occurred with the testing set containing three days, where the GA found a configuration with throughput equal to 2147.7, and the ILP achieved 2131.3 samples per hour. This is a 0.75% difference between the configurations. For the other optimization sets, the difference is a few units. Therefore, the optimization capabilities of both approaches are very similar. The ILP model seems to perform slightly better for sets with 11 and more days. However, this may be caused by the randomized nature of both the experiment and the optimizations. The experiment would need to be repeated multiple times, ideally with larger optimization sets, to carry out a more accurate assessment and minimize the effect of randomness.

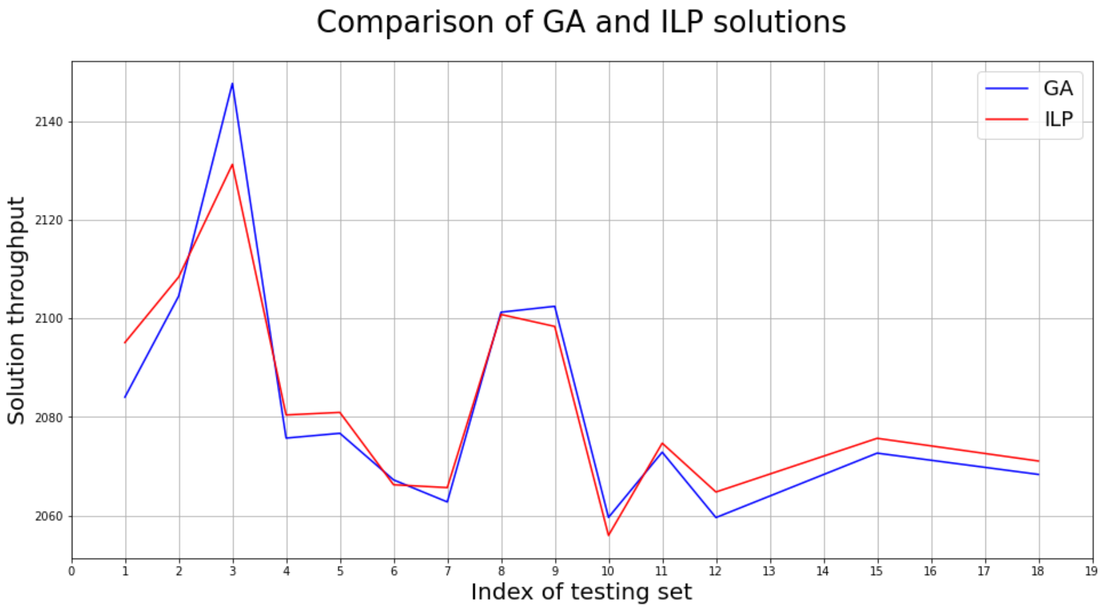


Figure 6.17 - Biochemistry throughput comparison of both approaches with the increasing number of considered samples

Although the performance of both proposed approaches is very similar, the time needed to find comparable solutions is different. In general, the GA finds reasonable solutions much quicker than the ILP model. To demonstrate this, we have performed a series of 5 experiments, each consisting of 6 optimization sets. Each set in an experiment contains several randomly selected days from 1 up to 6 days. First, we used our ILP model to find solutions with a 1% gap from the optimum, guaranteeing that the criteria value of such solution is at most 1% worse than the optimal value. Moreover, the ILP model was limited by a timeout set to 13 hours. If the timeout was reached, we instead extract the best solution. Afterward, the GA found solutions that are comparable to the ILP result. Comparable means that the throughput of the GA is not lower by more than five tests per hour.

The results of this experiment are available in Table 6.2. The columns represent the number of days contained within the optimization sets. We report the time in seconds required by the individual algorithms to find the solution and the absolute throughput difference between the solutions for each experiment and each set. Moreover, the absolute throughput difference is highlighted by a background color representing the approach that achieved the better solution. The green background stands for the GA, whereas the blue background represents the ILP. If the ILP time is equal to 46800, the ILP didn't find a solution with 1% optimum gap within the given time limit. There are a few occasions when the GA found a significantly better solution than the ILP. However, the ILP was stopped when the solution was guaranteed to be within a 1% gap from the optimum. Therefore, the reported solution might not be optimal but close to it. The GA reaches comparable solutions much quicker than the ILP model. Especially in the cases where the ILP reached its 13-hour time limit, whereas the GA found similar solutions within the first 20 minutes.

		Size 1	Size 2	Size 3	Size 4	Size 5	Size 6
1st Exp.	ILP	462s	1644s	1076s	5248s	46800s	1767s
	GA	38s	19s	86s	311s	552s	496s
	Difference	16.34	8.63	2.81	0.15	0.57	4.88
2nd Exp.	ILP	211s	3257s	2298s	8366s	2522s	3626s
	GA	18s	1359s	303s	144s	147s	759s
	Difference	2.8	3.89	0.36	15.33	3.16	0.96
3rd Exp.	ILP	270s	1913s	9657s	4446s	1965s	46800s
	GA	22s	748s	1586s	638s	45s	395s
	Difference	2.0	2.67	4.48	2.3	0.43	1.52
4th Exp.	ILP	46800s	461s	2943s	46800s	46800s	46800s
	GA	356s	413s	38s	762s	1417s	464s
	Difference	0.58	10.25	0.32	0.07	0.08	0.3
5th Exp.	ILP	170s	1261s	12563s	46800s	46800s	10881s
	GA	44s	96s	1838s	328s	185s	759s
	Difference	0.0	4.47	4.65	2.88	6.38	5.47

Table 6.2 - Comparison of Genetic and ILP optimization time

Moreover, we can observe from the table the varying optimization times of the ILP model. The expected behavior is that the time needed to perform the optimization increases with the number of considered samples. However, this is not always the case. Although the experiments with a higher number of considered days reached the time limit more often, the timeout occurred even in an experiment with only one day (4th experiment, size 1). Therefore, the optimization time is not based only on the number of considered samples but also on the spectrum of the sample's requests, which differ every day.

Although the genetic algorithm finds reasonable solutions very quickly in comparison with the ILP model, the algorithm might struggle to improve it as the optimization is mostly randomized. On the other hand, the ILP offers a systematic optimization process. Therefore, we believe that the optimization could be improved by combining the two approaches. The genetic algorithm would be utilized to construct a solution close to the optimum quickly, and this solution would be passed to the ILP model as an initial method assignment. The ILP would then attempt to improve the solution further and move it closer to the optimum.

6.3 Average TAT Minimization

This section discusses the experiments carried out with the TAT minimization ILP model and their results. These experiments are based on the operational data from Faculty Hospital Královské Vinohrady applied to the Prevedig automatization. Similarly to the throughput maximization, the optimization is done w.r.t the samples in the operational data. The samples are separated into individual 15-minutes long time intervals, depending on their arrival timestamps. Each sample is associated with its requested biochemical and immunological methods. In the performed experiments, we first found the configuration without any TAT limitations. Then, we have enforced numerous TAT requirements to adapt the solution to the hospital's needs. We simultaneously considered various limitations for each priority for 80% and 95% of the samples and selected the most strict satisfiable requirements. However, note that the sample TAT in conducted experiments reflects the time between the release of the sample from the input on the transportation track and the completion of the last sample's test. This does not consider the sample's centrifugation or its handling by laboratory personnel, which is typically included in the TAT reports of the laboratory.

The goal of the model is to achieve minimal average sample TAT. Because of the processing delays at the AU analyzer, it is reasonable to assume that as many methods as possible should be assigned to the DXC. The situation is similar for immunology. We assume two identical DXI immunology analyzers. Therefore, to accomplish the best sample TAT, most of the immunological methods should be assigned to the DXI1 as it is closer to the input. The first experiment without any hospital requirements supports these claims. The bar chart in Figure 6.18 shows the number of tests performed by the individual analyzers. The blue bars represent the number of methods associated with Routine samples, the green bar represents Statim priority, and the red bar stands for Vitals. According to the bar chart, 65.29% of biochemical tests were performed by the DXC analyzer, and 59.35% of immunological tests were executed by the DXI1. Therefore, the model found a configuration that utilizes the DXC analyzer as much as possible while respecting its capacity.

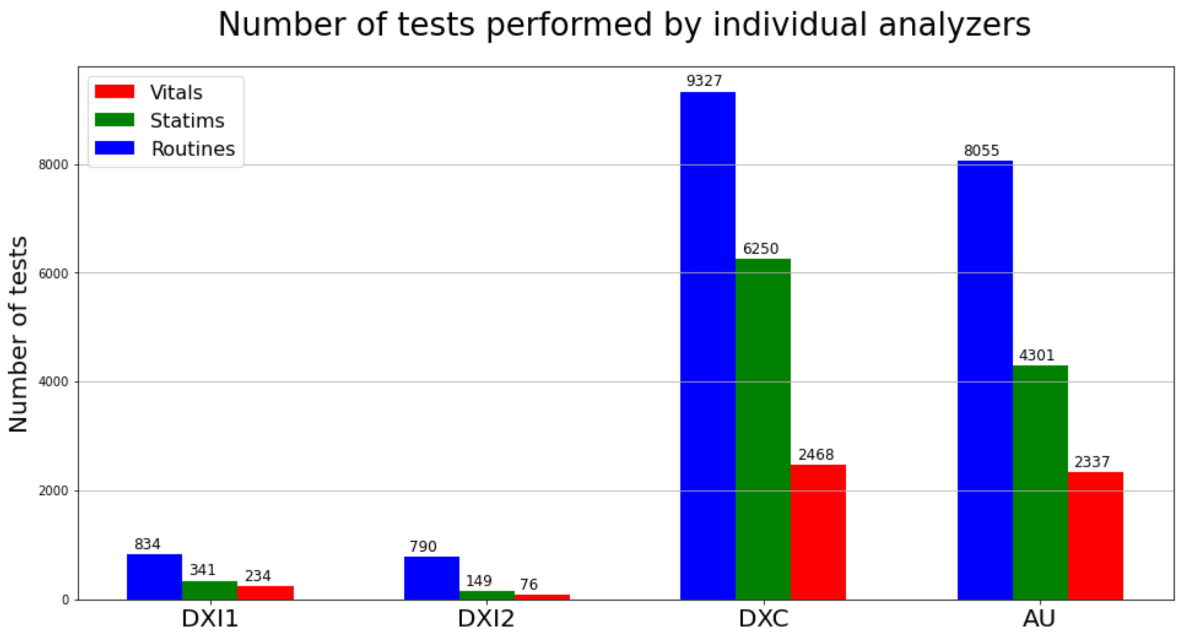


Figure 6.18 - Tests performed on the analyzers w.r.t. sample's priority

The remaining biochemical methods that the DXC could not perform are assigned to the AU. As for immunology, the majority of the tests are performed by the DXI1 analyzer. However, despite not enforcing any workload balancing criteria, the distribution of performed tests is not as unbalanced as expected. The reason might be that the analyzers are close to each other, and the sample transportation takes only a few seconds. Therefore, because of the low transportation time and high variability in length of immunological methods, it might be more appropriate in terms of TAT to assign methods with prolonged reaction times to the DXI1 and the short tests to the DXI2. The most protracted method determines the sample's TAT and should be initiate as fast as possible. If a quicker method is present, it may be pipetted before the long one, delaying its processing. However, delaying methods with shorter reaction times has no impact on the sample's TAT in such cases. Thus, by assigning them to the DXI2, we guarantee they are never pipetted before a method with a long reaction time.

This claim is supported by the correlation matrix heatmap of the immunology methods depicted in Figure 6.19. The correlation of the two methods represents the likeliness they are requested together. The methods written in red color are assigned to the DXI1 analyzer, and the blue methods are assigned to the DXI2. The red line visually separates the methods based on their assignment. The numbers on the diagonal denote the method's occurrences. On the DXI2, we can observe a group of highly correlated methods starting with S_AHIVc and ending with S_AHCVc. However, these methods are often requested along with S_HBsAgc and S_AHBsc available on the DXI1 analyzer. These two methods have the most extended reaction times. Therefore, they are assigned to the DXI1 to be initiated as soon as possible. These methods will likely be requested together with some of the correlated methods on the DXI2. Thus, to avoid a potential delay, the quicker correlated methods are assigned to the second analyzer, so the method pipetting order can no longer affect the sample's TAT. The situation is very similar to another pair of highly correlated methods – S_T4V and S_TSH. These two methods are almost always requested together, but S_T4V has a longer reaction time than S_TSH. Therefore, S_T4V is assigned to the DXI1, and S_TSH is available on the other analyzer.

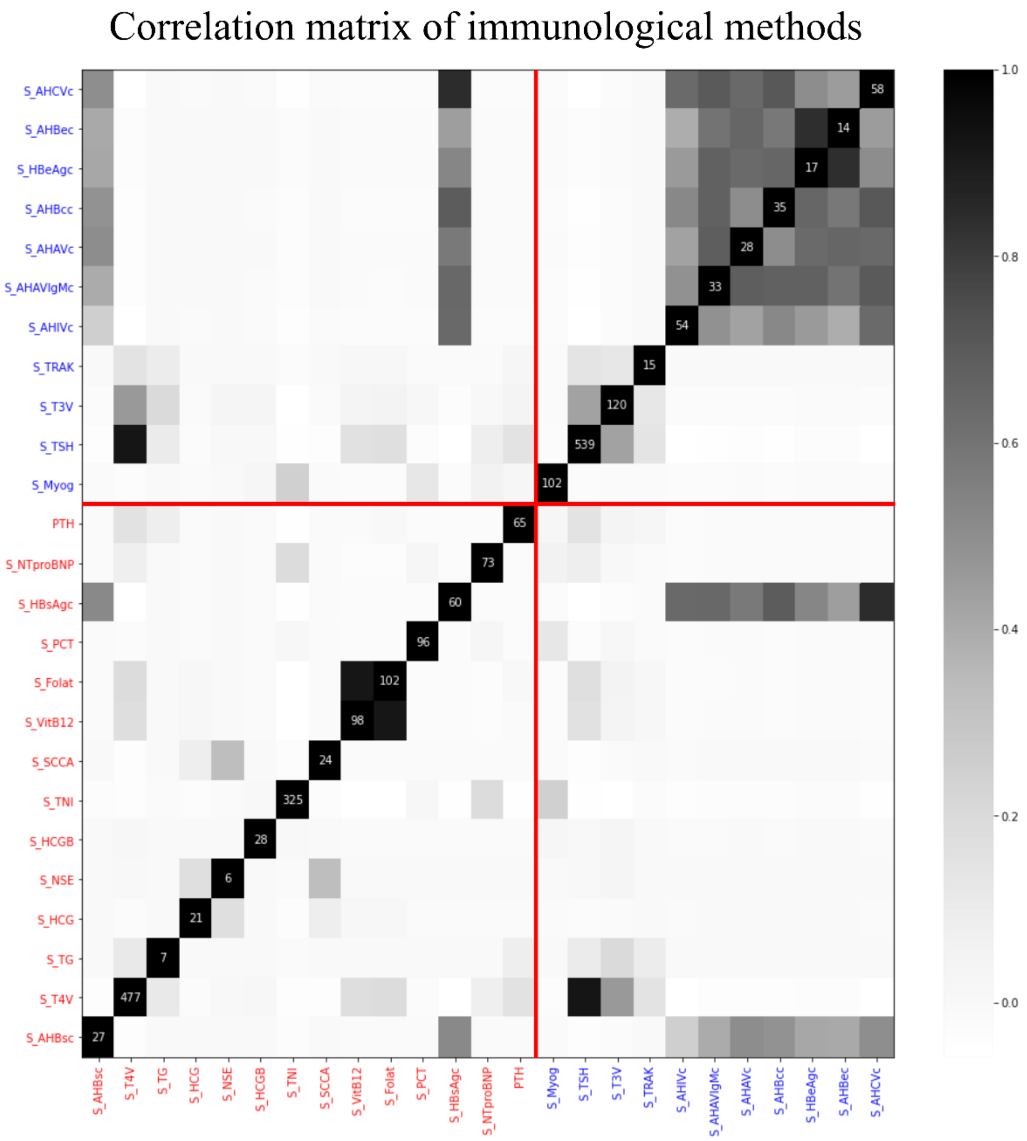


Figure 6.19 - Correlation matrix of biochemical methods without any hospital requirements

In biochemistry, the situation is different. The method's durations are identical, and the main decision factor is the 5-minute processing delay of the AU analyzer. Unfortunately, the DXC has limited capacity. Therefore, it is reasonable to assign highly correlated groups of methods to the DXC so that the samples are more likely to visit the DXC analyzer without traveling to the AU. The biochemistry correlation matrix of the model's configuration is depicted in Figure 6.20.

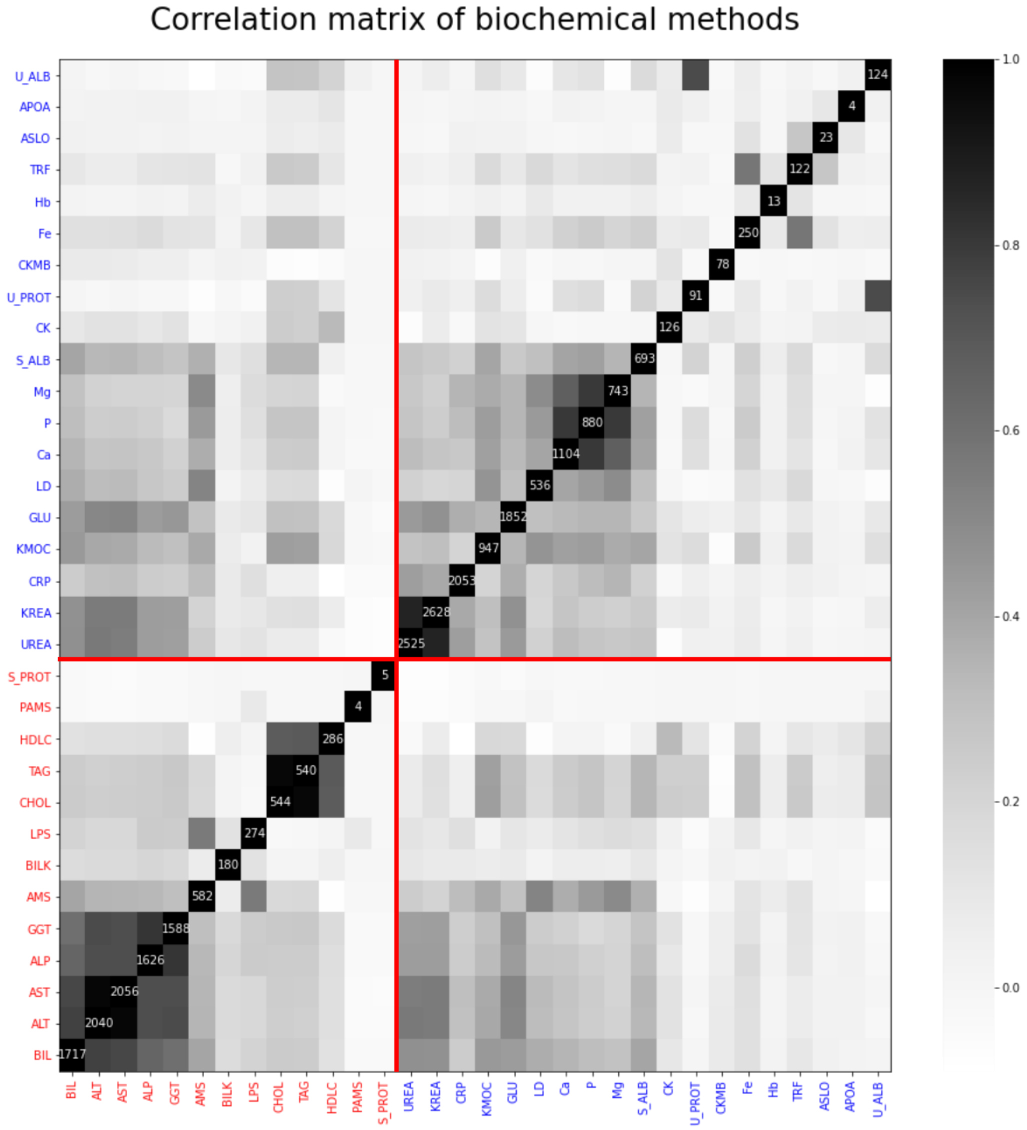


Figure 6.20 - Correlation matrix of immunology methods without any hospital requirements

We can observe a few highly correlated groups assigned to the DXC analyzer. Especially the pair UREA and KREA that are almost always requested together, or the group Ca, P, and Mg. On the other hand, highly correlated methods BIL, ALT, AST, ALP, and GGT are on the AU analyzer. The cause for this assignment is likely their very high number of occurrences. Assigning these methods to the DXC would introduce an enormous additional workload, potentially overloading the DXC. However, putting only a fraction of these methods to the DXC is not beneficial either. Because the methods are highly correlated, assigning any of them to the DXC analyzer would force a significant amount of samples to unnecessarily visit the DXC. The samples would be delayed either by the queue at the DXC or the transition between its states. Even if reassignment would cause a fraction of samples to end their processing at the DXC analyzer, the improvement is likely not high enough to offset the additional delays it caused for the majority of samples. Because the DXC can't process all these methods, it is more sensible to assign them all to the AU analyzer and avoid visiting the DXC altogether.

Until now, we have discussed the results of an experiment without any hospital requirements. We have conducted series of experiments to identify the most restricting satisfiable TAT requirements. Naturally, offsets exist between the limitation, and it is impossible to satisfy them all at once. Imposing a strict requirement on the Vital TAT will likely increase the average TAT of Statim and Routine samples. In a hospital environment, samples with Vital priority are usually inserted manually into the analyzer by the laboratory personnel to avoid potential delays in the automatization. Therefore, we are searching for the most restrictive Statim requirements instead. Table 6.3 contains the considered limitations of the results we will present shortly. The percentage represents the ratio of samples that have to fulfill the limit expressed in minutes. Limits with blue background represent a lower bound that is still satisfiable. For example, requesting completion of 80% Statim samples in 20.77 minutes is no longer satisfiable. The limits highlighted in red had to be increased, as their lower bounds cannot be satisfied with the Statim lower bounds. Therefore, to guarantee the lowest TAT of Statim samples, we had to increase the TAT guarantees for both Vital limitations and 95% Routines.

Routine		Statim		Vital	
Percentage	Limit	Percentage	Limit	Percentage	Limit
80%	34.10	80%	20.78	80%	21.42
95%	42.93	95%	34.10	95%	22.33

Table 6.3 - TAT limitations setting according to the sample's priority

In the resulting configuration, the assignment of immunological methods is identical to the configuration of the unlimited experiment. Thus, the situation has not changed in comparison with the correlation matrix in Figure 6.19. However, the biochemistry assignment has changed significantly. Surprisingly, almost every biochemical method is assigned to the AU analyzer, as is demonstrated by the correlation matrix in Figure 6.21. Only three methods – U_ALB, U_PROT, and S_PROT – with a total of 224 occurrences are available on the DXC analyzer. This result is surprising because the solution doesn't exploit the faster sample processing of the DXC analyzer at all. A possible explanation of this result is that the correlation between methods is slightly higher for samples with Statim priority than other priorities. Thus, it is more likely for Statim samples to travel to both the DXC and the AU analyzers. However, visiting both analyzers is not ideal as the sample's TAT will be affected by the processing delays of both analyzers. Thus, we want samples to either travel to the DXC or the AU, but not both. By imposing strict TAT limits on samples with Statim priority, the model is forced to shift its focus on the average TAT of Statim samples at the expense of Vitals and Routines. However, although it is possible to finish the processing of some Statim samples at the DXC analyzer, it might not introduce high enough improvement to offset the TAT detriment of samples that now need to visit both analyzers. Therefore, it is more beneficial for the average Statim TAT to assign all the biochemical methods to the AU analyzer than distributing them.

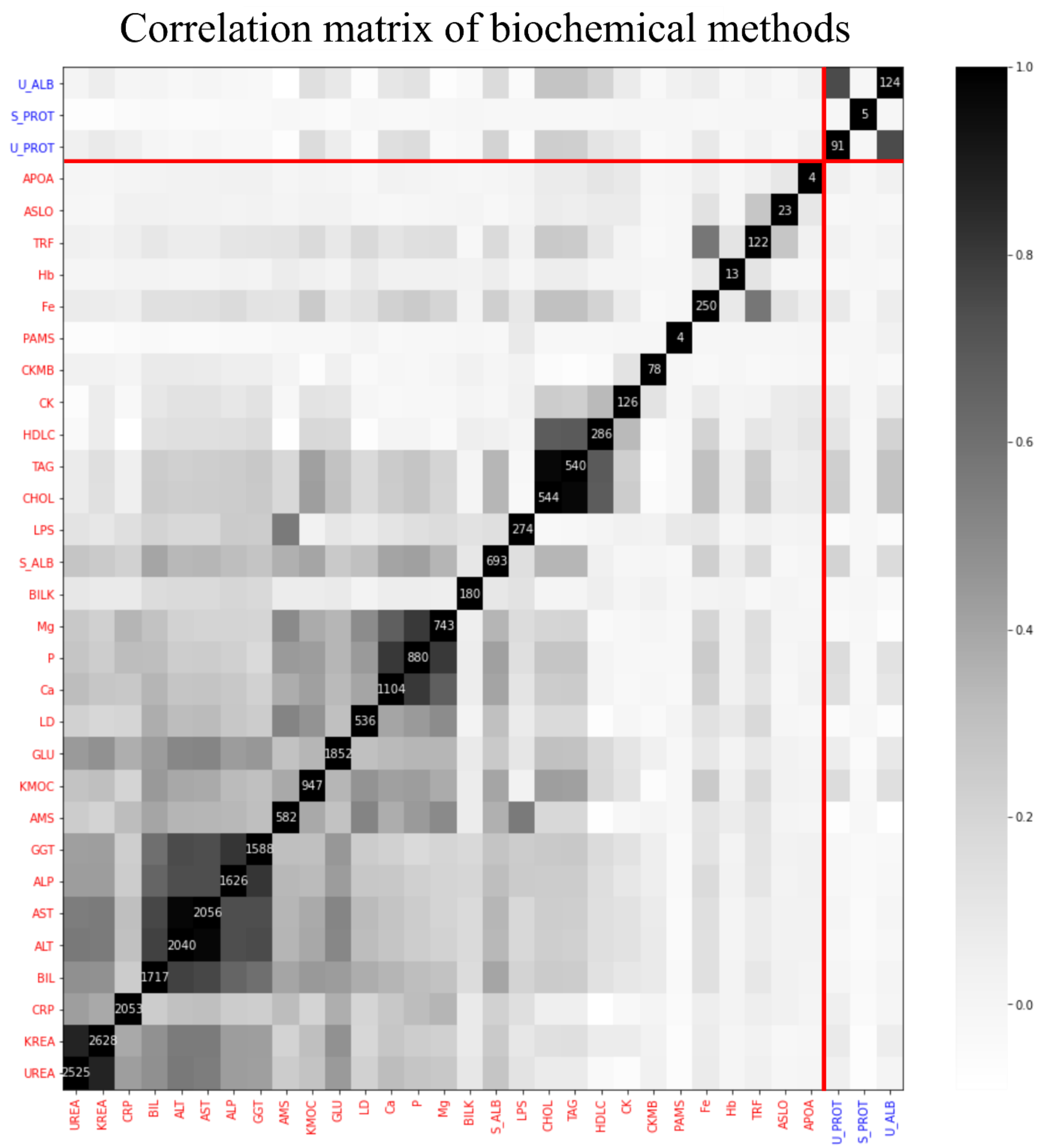


Figure 6.21 - Biochemistry method correlation matrix with imposed TAT requirements on samples with Statim priority

Naturally, all this comes down to the approximation of the waiting times at the DXC analyzer. We decided to approximate the DXC waiting time with a constant number. However, the waiting time at the analyzer highly depends on its workload. With a low amount of samples to process, the analyzer will often be in its idle state. Thus, the processing delay of samples is very high because the analyzer needs to transition between the idle and active states. However, the analyzer has very limited capacity and can be easily overloaded. Therefore, with a very high workload, the samples will spend a significant amount of time in a queue before the analyzer, resulting once again in high processing delays. Therefore, it would be ideal to perceive DXC processing delay as a variable that depends on the available workload of the analyzer and design a predictive model predicting the delays for the individual time intervals. However, because of the parabolic nature of the DXC delays, which are high for both low and high workload, it would be difficult to employ such a predictive model in the ILP without violating the linearity. Thus, a different optimization technique would be necessary. However, accurate predictions of the DXC delays could significantly improve the capabilities of the proposed optimization model. Unfortunately, the operational data is generated w.r.t. the Prevedig configuration. As we have demonstrated earlier, the DXC analyzer performs a fraction of the available biochemical methods. Therefore, we cannot accurately approximate the relation between the DXC waiting times and the amount of available work because the workload is always low.

7 Conclusion

The thesis researched the usage of optimization techniques in the Private laboratory Prevedig Medical intending to improve its automatization. The laboratory provided us its operational data and insight into the individual processes occurring during sample processing. We analyzed the operational data and identified two bottlenecks.

The first bottleneck regarded sample routing on biochemical analyzers. More than 90% of all the samples prioritized visiting the AU analyzer before the DXC. However, the samples undergo series of manipulations at the AU analyzer, delaying the sample’s processing at the DXC. Based on this finding, the laboratory introduced new rules for samples to prioritize visiting the DXC before the AU. The addition of these rules resulted in a 55.7% decrease in average biochemistry processing time.

The second discovery was the uneven workload balancing among the three biochemical components. Namely, the DXC’s component operated for less than one-third of the time compared to any of the two AU’s components. We defined the component workload balancing as an optimization task, searching for an assignment of biochemical methods to the individual components, minimizing the maximum component working time. We devised two data-driven approaches that solve this optimization task, first based on integer linear programming and the second based on an evolutionary algorithm. Both approaches utilize the operational data of the laboratory to adapt the configurations to the spectrum of test requests contained within it. Moreover, we included a side criterion to the algorithms with an adjustable limit to the amount of occurred transports between the biochemical analyzers. Based on this transportation upper bound, the algorithms focus their optimization on either workload balancing or sample TAT.

We split the data into two disjunctive sets to evaluate the solutions - an optimization set and an evaluation set. The approaches found optimal configurations w.r.t. to the optimization data, and the results were evaluated using the evaluation set. We conducted two experiments, one without any transport limitations and the other limiting the number of transports to the number of transport occurrences in the current Prevedig configuration.

In the first experiment, the optimization approaches found an assignment that achieves nearly uniform workload leveling among the biochemical components. Moreover, the resulting configuration improves the overall biochemistry throughput by 33.10%. However, this came at the cost of a 176.2% increase in sample transportations between the analyzers. Nevertheless, since Prevedig is a private laboratory, high system throughput is prioritized over low sample TAT. Therefore, w.r.t. the laboratory needs, this system configuration is significantly better than the previous one. The second experiment preserved the number of transportations of the original laboratory configuration while increasing the system throughput by 4.95%. Therefore, the optimization can maintain the advantages of the current laboratory configuration while still increasing the overall system throughput.

Lastly, we have defined a different optimization task regarding the minimization of average sample TAT, which is more critical for hospital laboratories. We designed a data-driven ILP model to solve this task. To simulate the hospital environment, we utilized operational data provided by Faculty Hospital Královské Vinohrady and applied it to the prevedig automatization. The approach finds an optimal method assignment that minimizes average sample TAT while respecting the laboratory TAT requirements based on sample priorities. The designed algorithm considers the limited capacity of the DXC analyzer and finds configurations that do not allow the analyzer to be overloaded.

The approach can exploit the correlation between occurrences of methods in the operational data. However, the results of experiments indicate that all the methods should be assigned to the larger, yet slower, analyzer to achieve the minimal average TAT of high priority samples. We believe this is not ideal. However, we imposed simplifications to waiting times at the DXC analyzer. Accurate approximation could significantly improve the model's capabilities. Unfortunately, the provided operational data does not contain sufficient information to make these predictions. This model will be further improved in the following research.

7.1 Future Work

In the throughput maximization approaches, we have approximated the additional load on the DXC analyzer caused by the ISE methods with a parameter computed from the data. However, the amount of performed ISE on the DXC depends on the assignment of other methods as they decide which analyzers should be visited. It is unknown how the system determines which analyzer performs the methods. However, accurate approximation of ISE execution w.r.t. to methods assignment could further improve the throughput maximization models.

In the average TAT minimization model, the waiting times at the individual analyzers were approximated by a constant. In the operational data, the available workload on individual analyzers is always below their service rates. Therefore, it is difficult, or even impossible, to accurately approximate the behavior of individual analyzers with a high enough workload. However, the waiting times at the analyzers depend on the amount of performed tests and should be computed w.r.t. the analyzer's load. We believe this would significantly improve the TAT model's capabilities.

In terms of constraints, we have enforced a hard constraint to the assignment of interfering pairs of methods, forbidding the assignment of such methods on the same component. Although the interference of methods introduces additional idle time to the system, it might be beneficial to assign them to the same component to achieve better workload leveling in some scenarios. However, whether the additional washing needs to occur or not depends on the order in which the sample's methods are performed. Therefore, it would be necessary to approximate the decision process behind the processing order of the individual methods.

Furthermore, we have limited the assignment of each method to exactly one analyzer because duplicated methods would increase the laboratory's operational and installation costs. However, frequent methods ISE and LIH are duplicated to avoid unnecessary sample transportation in the system. This idea could be further expanded to other frequently occurring methods to minimize the number of necessary transports while maintaining high system throughput and low sample TAT. However, with duplicated methods, the workload of the analyzers is no longer determined only by the method assignment but also by the rules determining which analyzer performs the methods.

8 References

[1] Joren Marynissen and Erik Demeulemeester. Literature review on multi-appointment scheduling problems in hospitals. European Journal of Operational Research, 272(2):407-419, 2019.

[2] Amir Ahmadi-Javid, Zahra Jalali, and Kenneth J. Klassen. Outpatient appointment systems in healthcare: A review of optimization studies. European Journal of Operational Research, 258(1):3-34, 2017.

[3] Suleyman Sevinc, Ulus Ali Sanli, Erdem Goker. Algorithms for scheduling of chemotherapy plans. Computers in Biology and Medicine, 43(12):2103-2109, 2013

[4] Laquanda T Leaven. Improving Hospital Laboratory Performance: Implications for Healthcare Managers. Hospital Topics, 93(2):19-26, 2015.

[5] Eline R. Tsai, Andrei N. Tintu, Derya Demirtas, Richard J. Boucherie, Robert de Jonge and Yolanda B. de Rijke. A critical review of laboratory performance Indicators. Critical Reviews in Clinical Laboratory Sciences. 56(7):458-471, 2019

[6] Onyenekwu CP, Hudson CL, Zemlin AE, et al. The impact of repeat-testing of common chemistry analytes at critical concentrations. Clinical Chemistry and Laboratory Medicine. 52(12):1739–1745, 2014.

[7] Tzu-I Chien, Jin-Ying Lu, Jau-Tsuen Kao, Ya-Chih Cheng, and Ya-Fen Lee. Evaluation and improvement strategy of analytical turn-around time in the stat laboratory. Journal of the Formosan Medical Association, 106(7):558 – 564, 2007.

[8] Yang T, Wang TK, Li VC, Su CL. The optimization of total laboratory automation by simulation of a pull-strategy. Journal of Medical Systems. 39(1):162, 2015.

[9] Wenhua Li and Xing Chai. The medical laboratory scheduling for weighted flow-time. Journal of Combinatorial Optimization, Nov 2017.

[10] Williams, Edward, Lote, Ravi, Ülgen and Onur. (2009). Simulation Of Medical Laboratory Operations To Achieve Optimal Resource Allocation. 10.7148/2009-0249-0255.

[11] James C Boyd and John Savory. Genetic Algorithm for Scheduling of Laboratory Personnel. Clinical Chemistry, Volume 47(1):118–123, January 2001

[12] Stephen S. Raab, Justin Swain, Natasha Smith, Dana M. Grzybicki. Quality and patient safety in the diagnosis of breast cancer. Clinical Biochemistry. 46(13-14):1180-1186, 2013

[13] Leeftink AG, Boucherie RJ, Hans EW, et al. Predicting turn-around time reductions of the diagnostic track in the histopathology laboratory using mathematical modeling. Journal of Clinical Pathology. 69:793-800, 2016.

[14] Prevedig Medical. https://www.prevedig.cz/. 2021

[15] Darrell Whitley. A genetic algorithm tutorial. Statistics and Computing. 4:65–85, 1994.

[16] Eiben A.E., Smith J.E. Introduction to Evolutionary Computing. Natural Computing Series. Springer, Berlin, Heidelberg. 2015

[17] Karel Gavenčiak. Optimization of samples processing in medical laboratories: Problem analysis based on data. Master’s thesis, Czech Technical University in Prague, Prague, Czech Republic, 7 2019.

[18] Adan, Ivo, and Jacques Resing. Queueing theory. 2002.

[19] Lehoczky, John P. "Real-time queueing theory." In 17th IEEE Real-Time Systems Symposium, pp. 186-195. IEEE, 1996.

[20] Czech Instutite of Informatics, Robotics and Cybernetics. https://www.ciirc.cvut.cz/. 2021

[21] Gurobi. Gurobi Optimization, LLC. https://www.gurobi.com/. 2021

[22] Pandas 1.0.5. The Pandas Development Team. https://doi.org/10.5281/zenodo.3509134. 2021

[23] JsonCpp. https://github.com/open-source-parsers/jsoncpp. 2019