Koninklijke Landmacht

A MILP Model to Optimize Staffing and Work Schedules with Reusable Resources at SIVO

Coenraad Kerbert

University of Twente

Faculty of Behavioural Management and Social Sciences

Industrial Engineering & Management

Supervisor SIVO:

Major S.A.P. de Waal

Supervisors University of Twente

First supervisor: **Dr. A. Asadi**

Second supervisor. **Dr. P.B. Rogetzer**

November 2023 – Enschede, Netherlands

Preface

Dear Reader,

In front of you, you find the thesis that I worked on as a closing of the Bachelor of Industrial Engineering & Management at the University of Twente. During this period, I had the honour to be part of SIVO, which is one of the training institutes of the Dutch Ministry of Defence Force.

At SIVO I was considered as one of the team and with great interest from my colleagues I was able to conduct a helpful investigation of a problem they have been struggling to solve for a long time. I must first thank Major Sabine de Waal who provided me with a spot in her team, her willingness to provide me with all the necessary information and contacts and made sure I was a real part of the team that was under her lead.

Secondly, I would like to thank my first supervisor Dr. Amin Asadi and my second supervisor Dr. Patricia Rogetzer for their continuous support. Even though the struggles faced throughout the process and the sometimes difficult situations, they have been able to support me and showed their willingness to make sure we got the best out of this thesis.

I would also like to put a special thank you to ir—Cornelis Ten Napel, who was my study coordinator for the past years. At many moments I have requested his help, and he has always shown his willingness and dedication to support me throughout.
Finally, I would not be able to have finished this thesis without the support and care of my family, girlfriend and housemates who have all helped to their greatest extent.

Hopefully, you enjoy reading this thesis.

Coenraad Kerbert

Enschede, November 2023

Management Summary

This research is conducted at SIVO (school initial shaping petty officers) which is an educational department of the Dutch Ministry of Defence Force. Within this department, there is the Transport Group which is responsible for supporting the educational activities outside of the base in Ermelo. The Transport Group is dealing with specific peak days throughout the week and is about to receive a new type of vehicle that is highly impacting the current schedule. A schedule that is already not running efficiently. The focus of this research is to find an optimal schedule for the peak hours and provide insights into how much staff and trucks are needed to solve the logistical issues. In order to do so, we will be guided by the following research question:

'How do different levels of availability of staff and reusable resources impact the makespan on peak days within the Transport Group of SIVO?'

During the research into the problem, it became clear how SIVO is struggling with the planning concerning the Transport Group. The interviews showed that the communication between the departments has not always been optimal. The Transport Group is often understaffed at peak days but overstaffed at other times during the week resulting in a lack of motivation among workers at the Transport Group. Lastly, the Transport Group seems to have a difficult situation coming up with the workload when a new type of truck is deployed which increases the processing time of multiple jobs.

To provide SIVO insights into how the situation can be handled, a Mixed Integer Linear Programming Model (MILP) is designed. The model aims to minimize the makespan of days when a peak occurs. We conduct experiments to study the impact of changing the number of staff and resources. Within the model, certain requirements are considered and some relaxations are implemented to prevent the model from becoming too tight. For example, an assumption is that all routes have a constant driving time. Lastly, the model considers the number of available reusable resources, which are the different types of trucks available to carry out the tasks.

The results of the model have been analysed using a sensitivity analysis for different numbers of staff. Also, the availability of reusable resources, trucks, has been tested to see what results are when their availability changes. The results showed that decreasing the number of staff can drastically change the processing times, but also brings a greater uncertainty to what the makespan of the schedule will be. Regarding the resources, 'Resource 0' which is responsible for the transportation of the students seemed to have little impact on the makespan unless only three units were available. Similarly, 'Resource 4' which transports the food for the platoons, was also hardly impacted by a smaller number of available trucks. 'Resource 2', needed for the transportation of the gear, on the other side seemed to have a large impact on the total processing times.

Ultimately SIVO was advised to find a solution for the off-peak days and to not necessarily hire more staff, as the MILP models proved that the tasks can still be executed with the current level of staff, despite the changing conditions after the deployment of the new type of truck.

Table of Contents

Abbreviations

Term	Meaning
CAV	Cavalerie
IG	Instruction Group
LOG-PEL	Logistical Platoon
Matbevo	Material supply
MILP	Mixed Integer Linear Programming
OL	Operation Logistics
OPG	Deputy Group Commander Training
PAS dagen	Partial Labour Participation Seniors
… -Pel	Platoon
SI	Schooting Instructors
SIVO	School Initial Shaping Petty Officers
Staf	Management
Verk	Reconnaissance
WLS	Swap Loading System

1 Introduction to the company

The Dutch Ministry of Defence Force is responsible for protecting all that the Dutch nation cherishes. They fight for a world of freedom and security. SIVO is a sub-department within this ministry. SIVO is the Dutch abbreviation for School Initiële Vorming Onderofficieren (School Initial Shaping Petty Officers). At SIVO petty officers are trained on their basic skills for their upcoming job role. SIVO trains a total of 500 soldiers spread among twelve different platoons ranging from 30 to 50 soldiers per platoon. All platoons follow training for 42 weeks during which they complete the education and become an official petty officer. The education consists of theoretical training and practical (field) training. The aim of becoming a petty officer is to develop further in the fields of leadership- and mentorship. It is worth mentioning that not everyone who starts this training also completes it (De Waal, 2022a).

In this thesis, we take a closer at the logistical department of SIVO in Ermelo. Every military base in The Netherlands has its logistical department that is responsible for a set list of tasks that apply to the nature of the military base. In general, the logistical department consists of three sub-departments that are all under the same chain of command from the upper hierarchy. This department is red-circled in Figure 1. The three different departments are responsible for the maintenance of trucks and cars, Matbevo (Material Supply) needed for soldiers to be able to operate and finally the Transport Group. The nature of the logistical department, and especially the Transport Group that we are focusing on, is to make sure that the educational platoons are supported throughout their training. This can be during exercises in the field, when they need supplies on the shooting range or the necessary food. This could mean bringing the petty officers to the site, dropping off the camp and its necessities, the water cart, and the fuel tank. Next to that they are responsible for the transport of food necessities in the field and support the 'line service' to and from the external transport facilities. This is the place where all packages and orders are collected and the place from which they are distributed among the base.

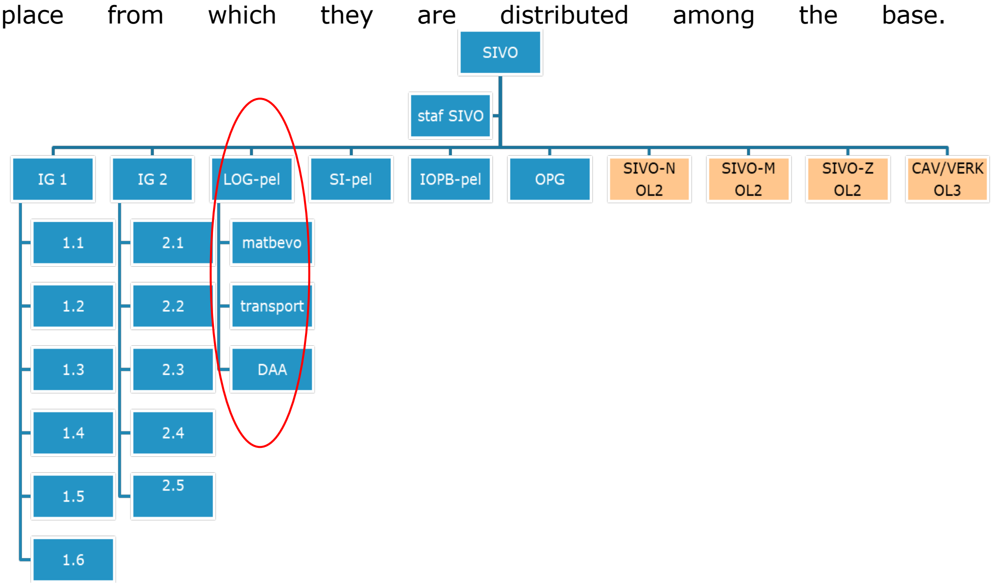

Figure 1: Organizational structure SIVO Base Ermelo (De Waal, 2022b)

2 Introduction to the problem

In the upcoming chapter, we introduce the problem at SIVO and the causes of the management problem. First, the core problem is identified, then the research question is presented and lastly, the problem-solving approach is outlined.

2.1 Management Problem

In this section, we will introduce the management problem at SIVO. Over the past few years, SIVO has been able to develop a greater interest among the public and because of that, their operations have been growing. This meant a requirement for more educational platoons, from nine to twelve (De Waal, 2022a). However, the Transport Group that supports these platoons logistically has remained in size and has become unable to provide its services to its fullest during peak moments. Those peak moments take place on Mondays when the platoons must be brought into the field, and on Thursdays when the platoons must be brought back to the military base in Ermelo. At these moments, a shortage of staff occurs regularly. As a result of this shortage, the Transport Group has a hard time completing all their tasks. This puts pressure on the staff which do not improve the working conditions.

A second aspect of the investigation is that the Transport Group of SIVO must cope with a new type of truck that makes use of a different loading system. The Dutch Ministry of Defence Force has contracted Scania to update the entire fleet with brand-new trucks (Ministerie van Defensie, 2023). One of the important trucks the Transport Group is currently making use of, a flatbed truck (WLS) that can load the exercise equipment in ten minutes (Figure 2), is being replaced by a new type of Scania that needs to be loaded with a crane as seen on the right in Figure 2. As a result, each crate of gear must be loaded individually, raising the additional loading time up to an hour. This puts substantially more pressure on the schedule of the Transport Group. As an example, loading the truck and driving it to one of the training locations has increased from an hour to almost three hours. The Transport Group is afraid they may not be able to cope with the upcoming changes considering the current staffing issues (De Wit, 2022).

Management problem: SIVO faces shortages of staff on peak days and is unable to cope with all tasks.

Figure 2: The old WLS vs. the new loading truck (Defensie, n.d.; Redactie, 2020)

2.2 Core Problem Identification

Multiple factors influence the management problem that is described in Section 2.1. These different elements are shown in the problem cluster in Figure 3. One by one we will address the problems.

2.2.1 Problem Cluster

Within the cluster of Figure 3, different issues lead to the same problem, a shortage of staff at peak moments. As indicated in Section 2.1, the Transport Group has two days on which there is a peak in the number of tasks to be performed by the Transport Group. However, on the remaining three days, there is sometimes too little work. The combination of a sometimes low intrinsic motivation of staff to perform additional work and workers who have become used to a small workload results in the fact that workers choose to take up leave on days when there is a peak in the number of tasks (Driver 1, 2022). Therefore, as a result, workers from the planning department are forced to drive at times when too few drivers are present.

Staff taking up leave, within the organization called 'PAS dagen' (Partial Labour Participation Seniors) is a combinatorial issue with the issues mentioned in the previous paragraph. The PAS Dagen is a crucial right staff have and enables the workers to perform the work they do on a full-time basis while considering their age. However, it leads to a similar issue that Pas Dagen are requested and approved on days when there are a considerable number of tasks to be done.

De Wit (2022) pointed out another crucial problem they are dealing with within the Transport Group. The available trucks require a special driving license provided by the Dutch Ministry of Defence Force. For the type of trucks that have been around for years, there is no issue. However, the new type of Scania that is delivered requires an additional license for the drivers. The problem lies in the duration of the process of requesting and planning the driving lessons. Until the driver's license has been provided to the driver, they are not allowed to drive the new Scania which makes it difficult in the planning to assign the right drivers to the trucks. Especially in the future, this will lead to problems when more and more Scania's of the new type are rolled out, but the driving licenses remain behind.

Another difficulty within the planning of all tasks is the limitation on working times. The drivers that are employed are limited to eight working hours per day, but the span in which tasks need to happen is between 06:00 and 18:00. This requires additional planning which could cause some problems if the availability of workers at a peak day is low.

A problem that occurs occasionally, is the sudden request for an additional platoon to go into the field and therefore request for more work from the Transport Group. It has happened that the planning department from SIVO was already aware of this change, but it was not communicated to the planning of the Transport Group.

Lastly, the new type of trucks as mentioned in Section 2.1 provides a large upcoming problem to the Transport Group. Due to the lengthened loading times of certain trucks, staff will be on duty for one task for a longer period. Time in which they cannot perform a different task. Considering the current shortages of staff this could cause more issues in the near future.

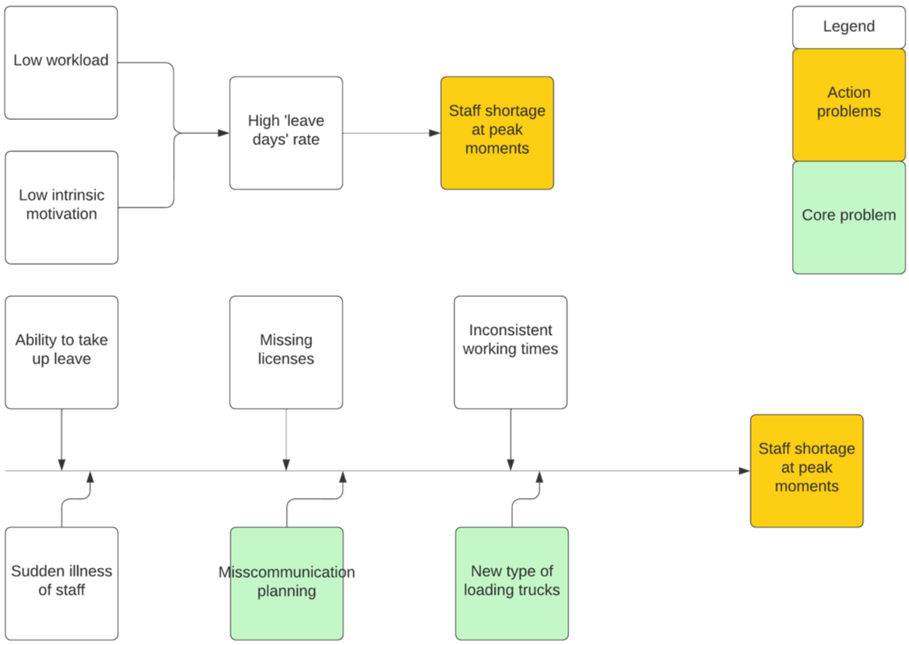

Figure 3: Problem Clusters SIVO

2.2.2 *Conclusion*

Considering the issues that have been discussed in Section 2.2.1 a core problem is identified. In Figure 3 two problems have been marked green as being the core problem. They have been chosen as the core problem as they both directly influence the operations of SIVO while they are also possible to directly be influenced and have an impact on the situation.

The focus of this thesis will, however, mainly lie on the change that will take place at the Transport Group due to the new type of trucks that will be delivered in the upcoming period. SIVO will need to find a way to deal with this new type of loading and its lengthened processing time per task.

2.3 Research Questions & Problem-Solving Approach

Building upon the core problem identified in Section 2.2, we now continue with the research question and the problem-solving approach.

2.3.1 *Research Question*

The research that will be answered throughout this thesis is as follows:

'How do different levels of availability of staff and reusable resources impact the makespan on peak days within the Transport Group of SIVO?'

2.3.2 *Problem-Solving Approach*

The main aim of the research is to work out how we can more efficiently deal with the different tasks of the Transport Group at the SIVO in Ermelo such that the working conditions are improved, and the peak moments are better managed. To do so we conduct a parallel machine scheduling strategy. Machine scheduling is a decision-making process that 'deals with the allocation of resources to tasks over given time periods with the goal to optimize one or multiple objectives' (Pinedo, 2022).

To gain some structure in our research we make use of a research model introduced by Stillman (2015). This structure is visually represented in Figure 4.

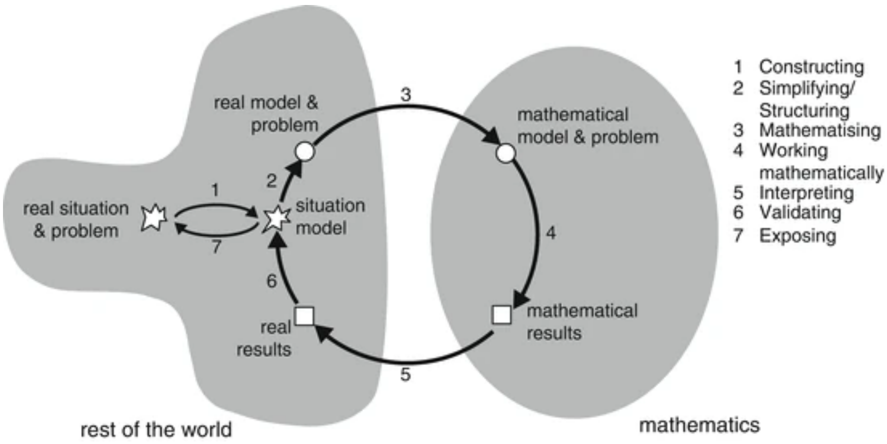

Figure 4: Modelling Cycle (Stillman et al., 2015)

Phase 1: Constructing

Within the constructing phase, we are positioned in the so-called 'real world'. Within this phase, interviews are conducted with the employees and identify the action problems present at the SIVO. Throughout this phase, we try to get an understanding of the problems and what their effects are but also look at how potential solutions can have a certain impact on a very generic scale. To solve the problem, we map out the core problem and find a potential type of model that could introduce a solution to the action problem. In a first attempt, an initial model on paper is constructed where insights are created into what the potential parameters and variables should be.

Phase 2: Simplifying / structuring

Now that the 'brainstorming' phase of the model has been rounded off a more critical eye towards the model is required. Which of the introduced elements on paper are needed, and which are considered unnecessary? We do this to make the model as efficient as possible and prevent errors throughout the process.

Phase 3: Mathematising

Within this phase, we construct the actual model and move from the paper to the computer model. In our case, we will be making use of Python. It is also at this point that we need to sort out data that is correct, useable, and readable by the program. When all elements have been loaded into the programme, we set the objective function and keep refining it until a workable result is presented. In our case, we will also be performing different experiments to gain insights into different scenarios.

Phase 4: Interpreting

The model presented in Python will give us insights into the different scenarios with different results for each scenario. Within this phase we look at how within these different scenarios the schedules come forward and if they are realistically applicable to the real-life world.

Phase 5: Validating

Within this phase we look at how valid certain results are, and which are not useable. Based on that we filter the useable data and visualise the new situation. The validation of the model is done at the very start, where a small test data sample is created and used to run the model. From this dataset, the solution can be calculated by hand and to validate the model it is checked whether the model provides a similar answer. Through this, the data set is constantly expanded until it captures the size of the actual dataset.

Phase 6: Exposing

Within this last phase, we provide the solutions and recommendations to the SIVO. From here on we have provided them with the necessary tools for them to be supported in their decision making.

3 Literature research

This section focuses on the supporting literature regarding the modelling of SIVO's logistical issue. The main aim of this section is to provide an answer to the following knowledge question:

'What methods and options are provided through literature to solve capacity and scheduling issues in the transportation industry?'

To provide an answer to the knowledge question above it is shown more insight through a sub-question about modelling the logistical issue. The knowledge question may provide a large set of information that is too broad in this sense. The sub-question, therefore, narrows down and puts a better focus on the research.

'How can mathematical solution methods support the optimization of the use of available resources in the transportation industry?'

3.1 Machine Scheduling

This section introduces machine scheduling and provides insights into how it applies to the point of interest of this thesis. Next, this section shows different methods of machine scheduling and their application.

First, a general introduction to scheduling is presented. Scheduling is a method used in the decision-making process for dealing with the allocation of resources over different tasks and a given period to optimize a set of objective(s). Scheduling can take different forms and types. In scheduling, there can be both deterministic and stochastic models, and the number and types of machines to be scheduled can vary depending on the specific problem (Pinedo, 2022).

Throughout this section, we will provide different machine scheduling methods such as single- and parallel machine scheduling, flow shop scheduling and job shop scheduling.

3.1.1 Single-machine Scheduling

In this section, we investigate single-machine scheduling. Single-machine scheduling is also said to be one of the building blocks of machine scheduling (Abedinnia, Glock, & Schneider, 2017). As the name suggests, it is a scheduling model with a single machine required to process certain jobs. Within single-machine scheduling, we have a set of N jobs that are processed on a single machine. Once a machine has finished a job, it becomes available again to process another job as it can often only process one job at a time. Figure 5 is an example of such a single-machine schedule. It clearly shows there is only one machine and there are six jobs to be processed with each their own processing time and deadline. In the table in Figure 5 the deadlines, due dates, end dates and lateness are shown and the *'Maximum Lateness'* is 13. From Figure 5 it can be concluded that the maximum lateness among all jobs is 13-time units. We will further explore this heuristic in Section 3.2.3. Depending on the model this can be done in various ways. The single-machine scheduling model can become more complex when different elements such as release times and deadlines are added. These are parameters that can be added to the model to *a)* make the model more sophisticated and *b)* provide an optimal solution to the real-world situation (Weerdt, Baart, & He, 2020).

Job	**1**	**2**	**3**	**4**	**5**	**6**
Processing time	6	3	9	7	8	2
Due Date	3	7	12	10	20	29
End Date	6	9	25	16	33	35
Lateness	3	2	13	6	13	6

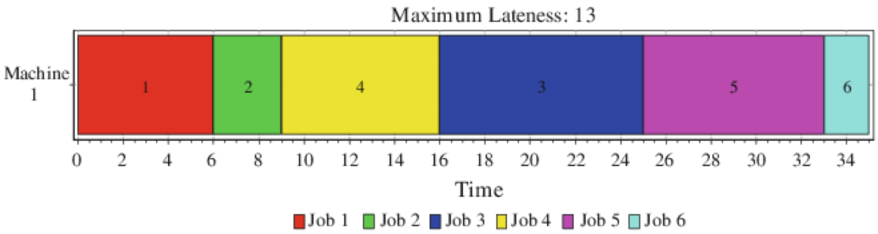

Figure 5: Table of job details and Gantt chart of a single machine schedule (Framinan et al., 2014)

3.1.2 *Parallel Machine Scheduling*

Parallel machine scheduling is based on two different types: identical and unrelated machines. Identical parallel machine scheduling is often used in for example the scheduling of processors of computers or the scheduling of a factory with identical machines (Xu & Nagi, 2013). Unrelated parallel machine scheduling implies that the processing on one machine does not impact the processes on other machines. Parallel machine scheduling is different from single-machine scheduling when dividing the load of all the jobs across machines is of interest. Therefore, the main objective is often, just as in many other scheduling techniques, to minimize the makespan (Pinedo, 2022). Within parallel machine scheduling all jobs must be executed but only need to be processed by one machine. This is visible in Figure 6, there are a total of four machines and eleven jobs. As shown, the schedules display the optimal solution such that each job is scheduled exactly once. Parallel machine scheduling also provides room to consider resources, especially reusable resources. Adding (reusable) resources creates a higher level of complexity, as not only machine availability needs to be considered but also that of the (reusable) resources. This means that this type of schedule can be used for more advanced systems that are larger in scale and require more machines, such as a production line where a tangible product is produced or a scheduling problem (Fanjul-Peyro, 2020).

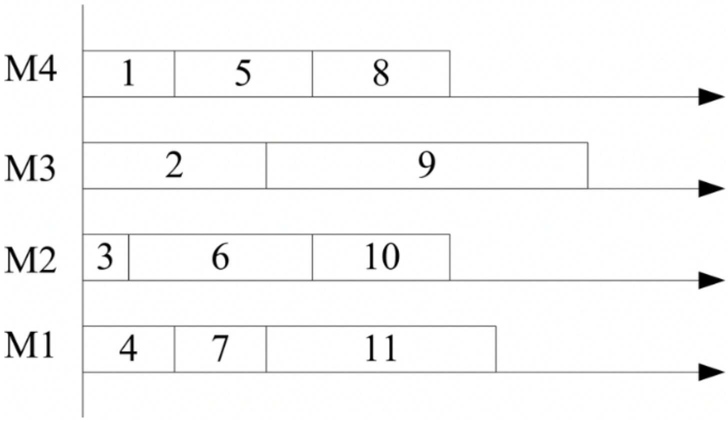

Figure 6: Gantt Chat of a parallel machine scheduling model (Xu & Nagi, 2013)

3.1.3 *Flow Shop Scheduling*

Next, we will dive into the matter of flow shop scheduling. Flow shop scheduling is a machine scheduling method often used in the production industry where each machine has its task, and each product needs to pass each machine. Within a flow shop scheduling problem, there are *m* machines (M_1, M_2, … M_m) available to process *n* independent jobs J_1, J_2, …, J_n. We have already seen similar notations in previous scheduling problems. What makes a flow shop special is that each job separately consists of *Q* operations Q_{ij} (i = 1,2, …, *m*, j = 1,2, …*n*). Each job needs to be processed on the dedicated machine in a pre-set order (Laribi, Yalaoui, Belkaid, & Sari, 2016). Figure 7 displays an example of a flow shop schedule. In total, four cars need to undergo degreasing and painting. These two tasks are done by different machines, and the cars first need to go through the degreasing process before they can be painted. A flow shop scheduling model has some assumptions that are required:

- Preemption requirements are not allowed.
- Jobs are independent and immediately available.
- A machine can only process one job at a time.
- Order of jobs is identical (Taillard, 1990)

Also, for a flow shop, it is possible to extend the model's complexity with additional constraints subject to the situation at hand. The goal of a flow shop scheduling method is generally to minimize the makespan or minimize the non-reusable resources as we have seen in the investigation of Labiri (2016). An example where such models occur often is standardized operations in factories, for example, the process of assembling a toy.

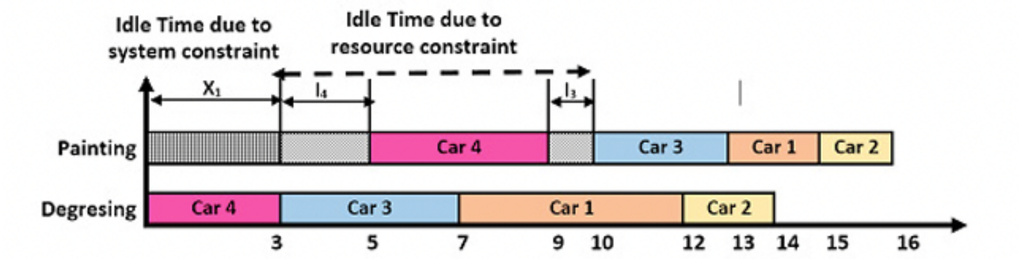

Figure 7: Schedule of a flow shop displayed in a Gantt Chart (Laribi et al., 2016)

3.1.4 *Job Shop Scheduling*

In this section, we discuss job shop scheduling. The method of job shop scheduling is like that of the flow shop problems. Whereas flow shop scheduling follows an identical path for each job, job shop systems do not have a fixed path for every job. This difference is seen in Figure 8 where certain jobs are scheduled over time *t.* Unlike Figure 7 the order of machines at which the products need to go through is not identical. It could for example be that a certain job needs to recirculate, which means that it needs to come back to the same machine for it to finish its processing (Pinedo, 2022).

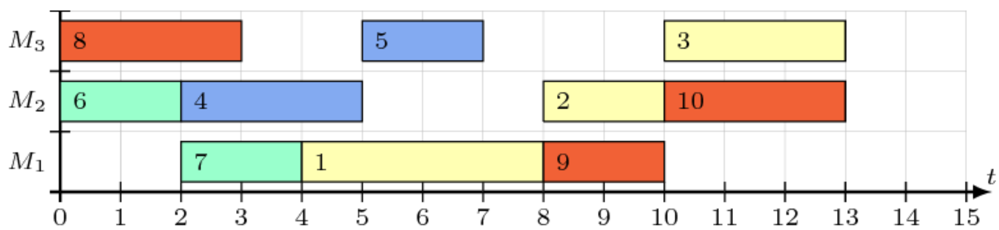

Figure 8: Gantt chart of a job shop scheduling model (Bürgy & Bülbül, 2017)

3.1.5 *Conclusion Machine Scheduling*

Within the previous sections, different techniques of machine scheduling have been presented, for now, it can be evaluated what plausible options could be. The schedules include a set of jobs that need to be completed and, depending on the choice of the scheduling method, the jobs will go through different machines where different tasks are carried out. The methods can vary from a simple production line where only two different types of machines are used, to a larger problem with multiple constraints. What they have in common is that the main objective is most often to minimize the makespan to find the most efficient schedule.

When this knowledge is applied to the objectives of this thesis then a conclusion about the application can be made. At SIVO, the goal is to optimize the work schedules and get insights into how many staff and resources are required to be able to fulfil all tasks. The tasks that are to be carried out are a set of tasks that each have their own processing time which is considered as fixed, and the sequence of the jobs is not primarily of importance. To execute the tasks a set of reusable resources are present, trucks in this case, and staff is needed to operate the trucks. Flow shop scheduling does not apply to the problem as for this type of solution method a fixed sequence of tasks is set up. In our case, there is no pre-set sequence of jobs. Job shop scheduling assumes that each machine has its own dedicated task, however, our staff members are identical and can therefore execute a range of different tasks, just one job at a time. That leaves us with two options, single and parallel machine scheduling. However, a single machine is insufficient for the situation as we need to consider multiple machines that work at the same time as there are multiple staff present. Therefore, the decision is to make use of a parallel machine scheduling with reusable resources. An example of a similar problem presented by Fanjul-Peyro (2020) and (ZHANG, 2021), comparing the problem to that of this thesis justifies the decision for unrelated parallel machine scheduling.

3.2 Choice of Solution Method

Now that it has become clear what kind of scheduling we can apply to our problem at SIVO, it is also important to identify the solution method to the problem.

'How can mathematical solution methods support the optimization of the use of available resources in the transporting industry?'

Within this section, multiple mathematical techniques for parallel machine scheduling are put forward and lastly, we will conclude which option is the best.

Throughout this section, the focus lies on exact and approximate solution methods. When options are considered to find an optimal solution to scheduling problems, the decision can be made to choose either of the two options. Both have their pros and cons and their purpose. All these elements will be discussed in the upcoming sections.

3.2.1 *(Mixed) Integer Linear Programming*

Within this section, the option is presented to make use of a (mixed) integer linear programming ((M)ILP) method. MILPs have a rich history in the operations research field and have proven to efficiently provide solutions. These methods present an objective function that consists of decision variables that are subject to constraints. The objective function is most often a linear equation to minimize or maximize. The results will provide an exact or near-optimal solution to the problem and give a good insight into the effects of the variables on the solution. Within these types of solution methods, the decision variables can only take on integer values and so too can the constraints (Swennenhuis, 2018).

The downside to making use of this type of solution method is due to its nature of finding the exact or near-optimal solution, it may become a very lengthy and timeconsuming option when the size of the problem increases. This can be due to the number of tasks to be scheduled, or the model being too tight due to the number or nature of the constraints. It therefore may show an infeasible result. To prevent this, the size and complexity of the model should be checked. If the model becomes too difficult to solve optimally, the decision should be made to accept a near-optimal solution (Pinedo, 2022).

3.2.2 *Constraint Programming*

Secondly, the methodology of constraint programming is discussed. Whereas MILPs already have a rich history in the field of operations research, constraint programming has become more common in the fields of computer science and artificial intelligence. Constraint programming aims to clarify relationships between different variables, unlike MILP these can also be non-linear. Constraint programming therefore tries to find a value to satisfy all constraints and an objective function is not necessarily needed. It can therefore be used for a wider range of problems and not necessarily just optimization.

The advantage of constraint programming is its performance for more complex problems, in contrast to MILP which could find it difficult to find a solution. However, the downside to its performance is the length it can take to come up with the result. Another downside can be the formulation of the model, which could become more complex than a MILP (Pinedo, 2022). Amodeo (2023) provides a constraint programming model for a machine scheduling problem where it is seen that the nature of the solution is like the (M)ILP, such as the objective function and the constraints. The main difference seen lies in the formulation and the actual nature of the method used to ultimately find the solution.

3.2.3 *Priority Rules*

Priority rules are a set of rules that determine the sequence of jobs that must be processed (Swamidass et al., 2004). One of the most common approximate solution methods is the use of priority rules. Examples of priority rules are first come, first serve (FCFS), earliest due date (EDD) or longest processing times (LPT).

The process of priority rule scheduling is easy to implement for the user. Within this process, one or multiple priority rules are chosen. Based on these priorities the jobs can be scheduled accordingly. This process is easy to follow for the planner and many different alternatives can be correct. What is most important is to have the correct data hand subject to the priority rule of choice (Pinedo, 2022).

The biggest upside of this process is the low amount of effort needed to find a proper schedule when no further constraints apply. There are many options and due to its simplicity, no real expertise is required. Therefore, it also takes little time and can be more cost-effective than large and expensive computer programs.

As a downside, the larger the problem gets the more difficult it becomes to find a suitable schedule. This might be due to the number of jobs that are to be scheduled or the number of constraints that become applicable. Also, schedules are often seeking optimality and efficiency, but making use of priority rules may prevent exact solutions resulting in a near-optimal solution that is not as effective as the optimal result.

3.2.4 *Simulated Annealing*

In this section, we discuss the process of simulated annealing. The book of Pinedo (2022) describes the process of simulated annealing. The general idea is to start with an initial schedule that based on a chosen heuristic is made. This could for example be a longest processing time (LPT) heuristic, while still considering the release times of jobs. Then the simulated annealing process starts with finding the next best alternative. If S_k is the current schedule in the k^{th} iteration and $G(S_k)$ is the value of the corresponding objective function, then we accept the next schedule only when $G(S_k) > G(S_{k-1})$. When a solution is not necessarily better than the current solution, an acceptance function is used that provides the chance that the solution is accepted anyway.

$$P_{cur,nb} = e^{\min\{f(\sigma_{cur}-\sigma_{nb}),0\}/c} \quad (1)$$

Equation (1) provides an exponential function where σ_{cur} is the value of the objective function of the current solution, σ_{nb} is the new value of the objective function and c is the 'temperature' at this iteration within the algorithm. The term temperature is an analogy by which the result's value is meant. The starting and final temperatures are determined using a special algorithm. The idea behind the temperature is that it is initiated with an extremely high value, and when the algorithm approaches a more refined schedule the temperature approaches zero, where it becomes less likely to accept worse solutions than the current solution (Rajaee, Aminduost, & Asadpour, 2018). Rajaee et al. (2018) apply such an algorithm to parallel machine scheduling and provide a clear application to their problem and therefore how it applies to the situation at SIVO. Even though the objective of minimizing tardiness is different to our objective of minimizing the makespan, the algorithm can still be applied as the objective of the model is not leading to whether simulated annealing could potentially be used.

However, there are also downsides to the use of simulated annealing. The risk of simulated annealing is that a solution can be chosen which is worse than the optimal solution. Therefore, simulated annealing is not an algorithm that provides an exact solution to the problem, but an approximate one that should be workable in many situations. This also provides an upside to making use of this heuristic, as it can handle large-scale problems efficiently whereas exact problems like the ones we mentioned previously may have more trouble finding an optimal solution when the size of the problem increases.

3.3 Conclusion

Within the previous sections, different elements regarding solution methods for parallel machine scheduling have been identified and elaborated on. All methods in one way or another will provide a solution to the problem that is workable for SIVO. To do so, we need to be aware of the elements present in the problem and the assignment set by SIVO. SIVO is looking to schedule its tasks more efficiently among the Transport Group and wants insights into what the level of staffing and resources should be to do so. Within the problem we must therefore look at the staffing levels, the presence of the available resources and the days at which a peak amount of work is requested from the Transport Group. We want to provide SIVO with an optimal schedule that prevents the ad hoc way of scheduling they are currently performing and therefore we want the solution provided to be as close to optimality as the situation allows.

The priority rule scheduling and simulated annealing do not apply to our problem since these methods are not exact enough and do not provide new insights into efficiency. Next to that, SIVO also wants more insights into what an optimal level of staffing is for their Transport Group. Priority rule scheduling and simulated annealing could potentially provide a schedule that is more efficient but the computational times for a priority rule scheduling or simulated annealing are approximately as lengthy as for a solution method that could find an exact solution considering the size of our problem.

This leaves us with two options that are relatively like each other, Mixed Integer Linear Programming, and constraint programming. The downside of constraint programming lies in the non-linear constraints whereas the MILP can more efficiently find the solution to the optimality problem we are facing at SIVO. This is due to the presence of having to schedule the resources for the jobs, that carry an integer value. Also, MILPs can carry larger problems than constraint programming. Therefore, the choice for the solution method of MILP is made.

4 Model

This chapter further develops the knowledge gained from Chapter 3 where a selection was made on the mathematical model regarding the scheduling techniques needed for the problem at SIVO.

In Chapter 2 of this report, we have identified the issue SIVO is facing and the causes that lead up to this problem. Part of the problem is that there is currently no systematic way of scheduling the different jobs that apply to the Transport Group of SIVO. Therefore, within this chapter, a systematic approach and solution to the scheduling of these sets of jobs is developed.

To recap this model's aim, a parallel machine scheduling model with reusable resources is chosen to obtain an improved workable schedule and to provide SIVO insights on the number of machines required and the efficient use of reusable resources. The optimization is targeting the Transport Group of SIVO. The optimizations are not route-specific, but it is the optimization of all tasks that must be conducted on a peak day. It is important to point out that in this case the staff of SIVO is treated as a machine, and that the reusable resources are several types of trucks each specified for their purpose.

4.1 Assumptions

Within the model, certain assumptions were made to prevent it from being overcomplicated or to leave out factors that are too uncertain to consider.

- The routes driven have a constant driving time, and uncertainty due to for example traffic jams is not considered.
- 'Resource 1' has been loaded and ready before it needs to drive from the base to its location. The unloading at the location is considered in its processing time.
- The platoons are always of similar size such that exactly three units for 'Resource 0' are needed.
- In the scheduling of driving food, only the jobs regarding breakfast and lunch were considered.
- Resources are always available for use, a breakdown for example is not considered.
- All jobs can be performed at any time and there are no set deadlines for certain tasks.

4.2 Parameters

For the model, it is important to set parameters that provide information and boundaries to ultimately come to the solution. The parameters include elements such as the number of machines and the number of jobs to be scheduled but also provide information about the jobs themselves what the release date of a job is and how many reusable resources are available.

Table 1: The parameters of the MILP model

Notation	Description	Value
$\boldsymbol{m}$	Set of machines	$i \in \{1, \dots, M\}$
$\boldsymbol{n}$	Set of jobs	$j \in \{1, \dots, N\}$
$\boldsymbol{p_j}$	Processing time of job j	
$\boldsymbol{r}$	Number of resource types	r = 1,2,3,4
$\boldsymbol{l_r}$	Number of resources type r	$l_1 = 9$, $l_2 = 3$, $l_3 = 3$, $l_4 = 4$
$\boldsymbol{c_j}$	Completion time job j	
$\boldsymbol{w_j}$	Total operating time machine j	
$\boldsymbol{t}$	Time periods in stages	$t = 1, \dots, T$
M	Big M	

4.3 Decision Variables

Next, we identify the decision variables. In this case, we have two binary decision variables, so they can only take on a value of 0 or 1. These two decision variables are crucial to constructing the matrices when the calculations are performed. The first decision variable is three-dimensional, which takes on a value of 1 when a job *j* starts on machine *i* at time t, and 0 otherwise. The second decision variable concerns the assignment of the reusable resource *r* to corresponding job *j.* The last decision variable is continuous and represents the starting time of job *j* on machine *i*.

$$x_{ij}^{t} \begin{cases} 1, when\ job\ j\ starts\ on\ machine\ i\ at\ time\ t \\ 0, elsewhere \end{cases} \quad (2)$$

$$y_{jr}^{t} = \begin{cases} 1, when\ resource\ type\ r\ is\ used\ for\ job\ j\ at\ time\ t \\ 0, elsewhere \end{cases} \quad (3)$$

$$s_{ij} = starting\ time\ of\ job\ j\ on\ machine\ i \quad (4)$$

4.4 Objective Function

The objective function regarding the problem we are modelling is to minimize the total makespan. That is the total time taken from the first task to start until the last task is finished. Minimizing the total makespan shows how the time can be used most effectively. Equation (5) shows this objective function.

$$\min C_{max} \quad (5)$$

4.5 Constraints

$$\sum_i \sum_t x_{ij}^t = 1,\ \forall j \quad (6)$$

$$s_{ij} = \sum_t t\ \cdot x_{ij}^t \quad (7)$$

$$s_{ij'} \geq s_{ij} + p_j\ \forall i, \forall j \neq j' \quad (8)$$

$$\sum_j \sum_t p_j \cdot x_{ij}^t \leq C_{max}\ ,\ \forall i \quad (9)$$

$$\sum_{j=1}^n y_{jr}^t \leq l_r\ \forall r, t \quad (10)$$

$$\sum_j \sum_t \quad p_j \cdot x_{ij}^t \leq 24, \forall i \quad (11)$$

$$y_{jr}^t \leq 1 - \left(\frac{t-c_j}{M}\right)\ \ \forall j, r, t \quad (12)$$

$$M = \sum p_j\ \forall j \quad (13)$$

$$x_{ij}^t \in \{0, 1\},\ \forall i, j, t \quad (14)$$

$$y_{jr}^t \in \{0, 1\}, \forall j, r, t \quad (15)$$

$$s_{ij} \geq 0\ \forall i, j \quad (16)$$

Equation (6) ensures that all jobs are started exactly once and are therefore assigned to a machine such that no tasks remain incomplete at the end. Equation (7) defines the continuous variable s_{ij} and provides the link with the binary variable x_{ij}^t. Equation (8) is put in place to ensure that each staff member only processes one job at a time, the machines by themselves cannot run parallel tasks. Equation (9) makes sure that the total processing time is smaller than the outcome of our objective function to minimize the makespan. Equation (10) prevents more reusable resources at time *t* are used than there are available in total. Equation (11) calculates the total time a staff member is occupied which must be smaller than 6, or 24 time units of 15 minutes each, hours per day as that is the set time in which jobs must be finished. Equation (12) releases the used resource for a job when it is completed. Equation (13) defines the value of Big M. Equations (14) and (15) assign the binary values. Equation (16) defines the continuous variable for the starting time of each job.

4.6 Model Notation

To start the development of the model, it is important to know the details of the different jobs, their machines and how the model notation is used. Some brief details are mentioned in the introduction of this chapter, and we will now share more details about the elements behind the model.

4.6.1 Machines

Machines are an important part of the current model. Whereas machines are often associated with large bulky objects inside a factory that produce a certain good, in this case, machines are the staff members of the Transport Group. One of the aims of this investigation is to find the optimal number of staff to be able to undertake all needed operations. Considering the staff as a machine means that all staff can be seen independently and as all staff inside the Transport Group are assigned similar tasks it simplifies the model rather than doing a workforce scheduling model and minimizing staff as SIVO is interested in the effects of different staffing levels and resources. Lastly, as the staff can all do the same tasks, only the time they are occupied when performing a task is of importance as they cannot be scheduled for any other task during that time.

4.6.2 *Resources*

For this MILP model, a parallel machine scheduling model is used in which the use of reusable resources is considered. The models of Olteanu (2022) and Fleszar (2018) have been taken as a basis for this model.

The reusable resources that are of concern are the trucks that are used to perform the jobs. Four different types of trucks are used to perform different types of jobs. In Table 2, the current availability of the different types of resources is shown. As an example, of 'Resource 0' there are nine units in total available to be scheduled. They can be run parallel at the same time and be used for a new job when the previous job is finished.

Table 2: The different types of resources and their amount

Type of resource	**Amount available**
0	9
1	3
2	3
3	4

The nature of each resource is different. 'Resource 0' is used to transport all petty officers to any location that is needed. Each resource can carry a total of 16 people. 'Resource 1' is a WLS truck that is being replaced by a new model that requires a different way of loading, as was discussed first in Section 2.1. 'Resource 2' is the type of resource that the Transport Group makes use of to transport the electricity generator and a water tank. These are individually placed on their trailer, and both require an individual resource to be pulled. The last resource, 'Resource 3' is a pickup truck that is used to drive the food to the practice locations or to be somewhere quickly when needed.

4.6.3 *Jobs*

For the reusable resources to be assigned, more details about the different jobs are required. Four different locations can be visited during the jobs.

Route 0: work at the base

Route 1: Base – Sparrendaal – Base

Route 2: Base – Beekhuizerzand – Base

Route 3: Base – Stroe – Base

Route 4: Base – ASK – Base

In Figure 9, the different locations are displayed. Due to the nature of the jobs, no routes can be combined. Therefore, we do not consider certain routing heuristics.

Figure 9: A map with the different locations (Google, n.d.)

In Table 3 the details from the different jobs are displayed. For every day there can be any combination of jobs needed to be performed by the Transport Group. This list displays all jobs but are thus not the fixed list for each day.

Table 3: Details from jobs that could be applicable on a peak day.

Job	Route	Duration	Type of Resource required	# of resource required	Description
1	1	90	1	1	Driving practice gear
2	1	30	2	2	Driving watertank and generator
3	1	30	0	3	Driving students
4	1	30	3	1	Driving breakfast
5	1	15	3	1	Driving lunch / diner
6	2	120	1	1	Driving practice gear
7	2	30	2	2	Driving watertank and generator
8	2	30	0	3	Driving students
9	2	75	3	1	Driving breakfast
10	2	45	3	1	Driving lunch / diner
11	3	180	1	1	Driving practice gear
12	3	120	2	2	Driving watertank and generator
13	3	120	0	3	Driving students
14	3	120	3	1	Driving breakfast
15	3	90	3	1	Driving lunch / diner
16	4	180	1	1	Driving practice gear
17	4	120	2	2	Driving watertank and generator
18	4	120	0	3	Driving students
19	0	60	1	1	Unloading resource

4.7 Conclusion

Throughout this chapter, the MILP for our case at SIVO has been developed. The different elements such as jobs and resources have been clarified. Next to that, the parameters and decision variables have been set up, and lastly, the constraints applicable to our case have been shared and explained in Section 4.4.

5 Results and Analysis

In this chapter, the results of the developed model are analysed. Firstly, a discussion on the performance of the model is presented, secondly, the results of the sensitivity analysis of the different number of machines and reusable resources are shown.

5.1 Structure of the runs

To get a proper interpretation of the results it is important to be aware of the structure of the experiments. In Table 4 the different experiments are indicated, the order inside the table is also the order in which the experiments were conducted. The table also presents the increment levels between the lower and upper bounds of the experiments.

Table 4: Details of different experiments on staff and reusable resources

Experiment	Lower Bound	Upper Bound	Increment Level
Staff	6	15	1
'Resource 0'	3	9	3
'Resource 2'	1	3	1
'Resource 3'	1	4	1

This sequence was decided according to the aim of this research. It is most important to SIVO to see what levels of staffing have a certain effect on the total makespan of all tasks on peak days. From Chapter 2 we know that these peak days are on Monday and Thursday, these are the days that have been analysed when such a peak occurred. Within the machine experiments, we initially started with fifteen machines, as this would act as a base for the model and a point to start with. Next, the number of machines was reduced to a maximum of six machines. A total of six staff members was chosen as for each route a minimum of six drivers are needed and on peak days it is given that there are at least two routes to be driven so six workers is a bare minimum. We then continued with the experiments for the resources but continued with the optimal number of machines that would not exceed 360 minutes, and 24 timestamps as indicated by equation (11) in Section 4.5. This was decided as more than 360 minutes would mean the processing time would become too high and limit the platoons in their preparations at the location where they have been dropped off.

Within the experiments for the resources, we started with 'Resource 0' (the resource used to transport the students) and reduced these for each experiment by three units. So, from nine to six to three. We decreased by three as for each route exactly three of these resources are required, as is indicated in the assumptions in Section 4.1. For the other resources we kept the same strategy as for the number of staff and 'Resource 0' but only decreased the units by one unit.

5.2 Performance of the Model

Within this section, the performance of the model is elaborated on. The model was executed in Python using the MIP package. The MIP package enables Python to run mixed-integer linear programs and provide a solution to the presented problem (MIP, n.d.) Within Python, the CBC solver was used.

Regarding the computational time of the model, the performance was satisfactory for SIVO to analyse the results day by day, with computational times of one to three minutes, up to a maximum of eight minutes. The experiments were run per day so total jobs ranged from nine to a maximum of forty jobs. It is known that the MIP package and using the CBC-solver is not the most powerful tool for such problems but due to the size of the problems solutions were produced within minutes.

5.3 Results

Within this section, the results of the model are presented. Experiments were conducted for the first twenty-six weeks of 2023 where a total of eight extremely busy days were identified. A day was considered busy when three or more platoons had to be sent out to the field. If there were two platoons or less to be sent out it was considered an ordinary day. For the current experiments, all busy days during the first twenty-six weeks were considered.

There are different elements to the results, firstly experiments were executed among the number of staff for the given set of jobs. Once an ideal number of staff had been found, experiments were done regarding the number of resources available and the effects if their numbers were changed.

Within the analysis of the staff and the different reusable resources, a few elements will be touched upon. There are some common elements such as the mean, standard deviation, and standard error. Additionally, we will also investigate the F-statistic and the p-value. First, the sum of squares between groups and within groups tells us about the variance in the data between the groups or within the groups. Between groups would be the variance between the different numbers of staff. Within groups would be the variance between all the data of only ten staff members, for example. the F-statistic will tell us the ratio of the variability between the different groups. A high ratio of 4.0, for example, would tell us that the variability between the groups is 4.0 times higher than what would be expected by chance (Glenn, n.d.). Lastly, the p-value is a statistical measure on which the researcher can base the likelihood that the outcomes are the result of chance. Often, a value of 0.05 is taken for reference to accept or decline the null hypothesis. In our case, the null hypothesis (H_0) is rejected when the p-value < 0.05. This means the results are unlikely to be due to chance alone and there is a systematic difference. When a p-value > 0.05, then we cannot conclude that not enough evidence is available to reject the null hypothesis and we can therefore not say that the groups are different based on the data. (Beers, 2023).

5.3.1 Staffing Experiments

Within this section, an analysis is done on how different number of staff have an impact on the makespan. After conducting a total of 65 runs on the number of machines, a trend can be observed. In Table 5 average results from all runs together are displayed.

Table 5: results from experimenting on the number of machines and total makespan

Number of staff	Mean total makespan (min)	Max.	Min.	σ	Std. error
15	260.63	360	180	66.57	23.54
13	279.38	360	180	70.78	25.03
12	282.86	345	180	78.15	29.54
11	282.86	345	180	78.15	29.54
10	304.29	450	180	104.18	32.80
9	334.29	510	210	106.32	40.18
8	381.43	570	225	121.51	45.93
7	450.00	730	285	129.61	48.99
6	531.43	735	345	151.21	57.15

Table 6: Analysis of variance in number of staff

Source	Degrees of freedom	Sum of Squares	Mean Square	F-Stat	p-Value
Between Groups	8	498459.57	62307.45	6.03	0
Within Groups	56	578965.15	10338.66		
Total:	64	1077424.72			

Using the information from Table 5, the following line graph in Figure 10 can be constructed.

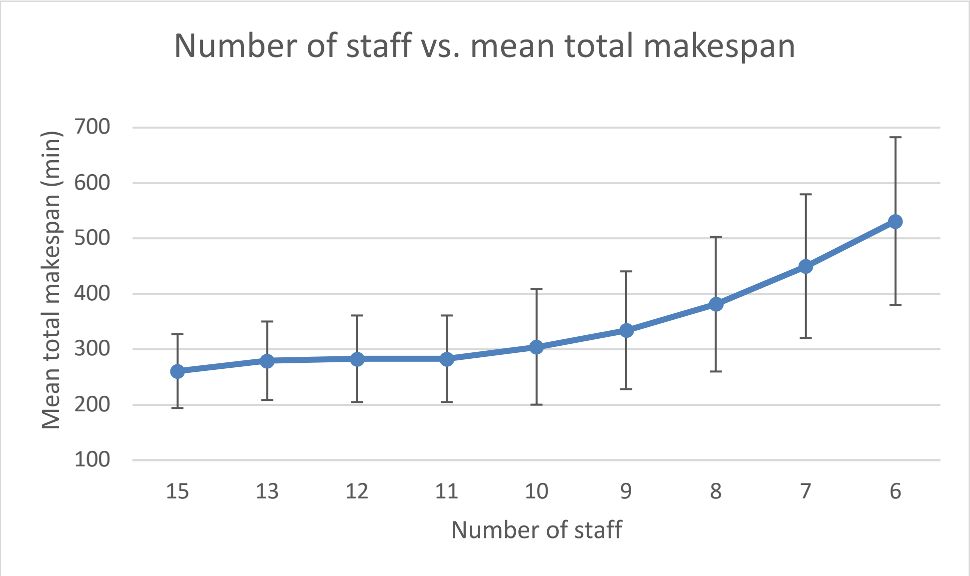

Figure 10: Number of machines vs. average total makespan

Figure 10 provides a good indication of what the effect is of decreasing the deployment of staff on the mean total makespan. In Figure 10 we see an inverse relationship between the number of staff and the mean total makespan. However, we can learn more from this relationship if we also look at Table 5. Table 5 shows the dataset on which Figure 10 is constructed. From this data, we can read that the increase in the mean total makespan among all conducted experiments only slowly increased when decreasing staff from 15 to 10 members with incremental steps of one. As we decrease to nine staff members, we obtain a substantially bigger increase in the mean total makespan. Table 5 also provides us with information about the standard deviation and standard error. From 15 to 11 staff members, they are both stable. For 15 and 11 staff members the standard deviation is 66.57 and 78.15, respectively. But, when only 10 staff members are deployed the standard deviation increases 104.18. This standard deviation increases to 151.21 for only six staff members.

Table 6 provides information on the conducted experiments and the effect of deploying fewer staff members on the mean total makespan. Table 6 is an analysis of the variance of the number of staff. We mainly look at the F-statistic and the p-value. The meaning behind these values is explained in Section 5.3. The value of the F-statistic is 6.03, meaning that the variability between the different numbers of available staff is 6.03 times higher than is expected due to chance. The p-value has taken on a value of 0. Zero is below our threshold of 0.05 and therefore there is a 0.0% chance the results are produced by chance and therefore the difference between the numbers of staff is unlikely to be equal.

To conclude, both Table 5 and Table 6 provide us with insights that there is a significant difference when the number of present staff is altered. There is a certain boundary around 10 and 9 staff members where the mean total makespan and the standard deviation considerably increase. As a result, this means it not only takes longer for the staff members to finish all jobs but there is also a greater uncertainty in how long it takes as indicated by the standard deviation.

5.3.2 Experiments on Resources

As a second step in the process, a sensitivity analysis of the number of available resources has been conducted. Previously we looked at the number of staff, and throughout this process, an ideal number of staff per experiment was chosen as a base to perform this analysis on the resources. For example, in week 13 on Thursday there are a total of 24 jobs to be scheduled. When performing the experiments on the number of staff we get a total processing time of 345 minutes when eleven to fifteen staff members are available and 450 minutes for ten staff members. 345 minutes is below the 6 hours (360 minutes) mark that was set for all jobs to be finished. Therefore, eleven machines were chosen to conduct the experiments on the reusable resources. The detailed results can be found in Table 7.

Table 7: Result from the experiments on week 13 on Thursday

#staff	# 'Resource 0'	# 'Resource 1'	# 'Resource 2'	# 'Resource 3'	Total Makespan (min)
15	9	3	3	4	345
13	9	3	3	4	345
12	9	3	3	4	345
11	9	3	3	4	345
10	9	3	3	4	450
9	9	3	3	4	510
8	9	3	3	4	570
7	9	3	3	4	630
6	9	3	3	4	735
11	6	3	3	4	345
11	3	3	3	4	465
11	6	3	2	4	405
11	6	3	3	2	345
11	6	3	3	1	390

Resource 0

The first sensitivity analysis on the reusable resources was on 'Resource 0. For 'Resource 0' only experiments were conducted for the availability of nine, six and three trucks, this is due to the assumption we made in Section 4.1.

From Table 8 we can first conclude there is little difference in the mean total makespan between the three different levels of availability of 'Resource 0'. 9 and 6 units of 'Resource 0' have a similar mean of 327.86 and 3 units increases a little to a mean of 345. Furthermore, we can see that the standard deviation when only three units are available is increased to 60 while 9 and 6 units only have a standard deviation of 29.27. Although it is a small difference, it is a noticeable difference in what the effect is of altering the availability of 'Resource 0'.

Based on the experiments performed we can also draw conclusions from the F-stat and the p-value shown in Table 9. First, the F-stat is much lower than what we have seen in Section 5.3.1 where the analysis of the different numbers of staff was conducted and had a value of 6.03. The F-stat on 'Resource 0' is 0.39, meaning the variance between the three different levels of availability is only 0.39 times larger than what is expected due to chance. Second, the p-value of 0.68 is larger than the set boundary of 0.05 and we therefore do not have enough evidence to reject the null hypothesis and can therefore not say the groups are substantially different than based on the presented data.

Concluding for 'Resource 0' we see there is little difference in having fewer units available than the current nine units that are available to SIVO. Only when three units are available there might be a bit more uncertainty in the mean total makespan of the jobs. However, the F-stat and the p-value tell us it is unlikely the groups are very different to each other and there is therefore little to no effect when decreasing the available units.

Table 8: Results sensitivity analysis for 'Resource 0'

# 'Resource 0'	Mean total makespan (min)	Max.	Min.	σ	Std. error
9	327.86	345	285	29.27	11.07
6	327.86	345	285	29.27	11.07
3	345	465	285	60.00	22.68

Table 9: Analysis of Variance on number of available units of 'Resource 0'

Source	Degrees of Freedom	Sum of Squares	Mean Square	F-Stat	p-Value
Between Groups	2	1371.44	685.72	0.39	0.68
Within Groups	18	31885.71	1771.43		
Total:	20	33257.15			

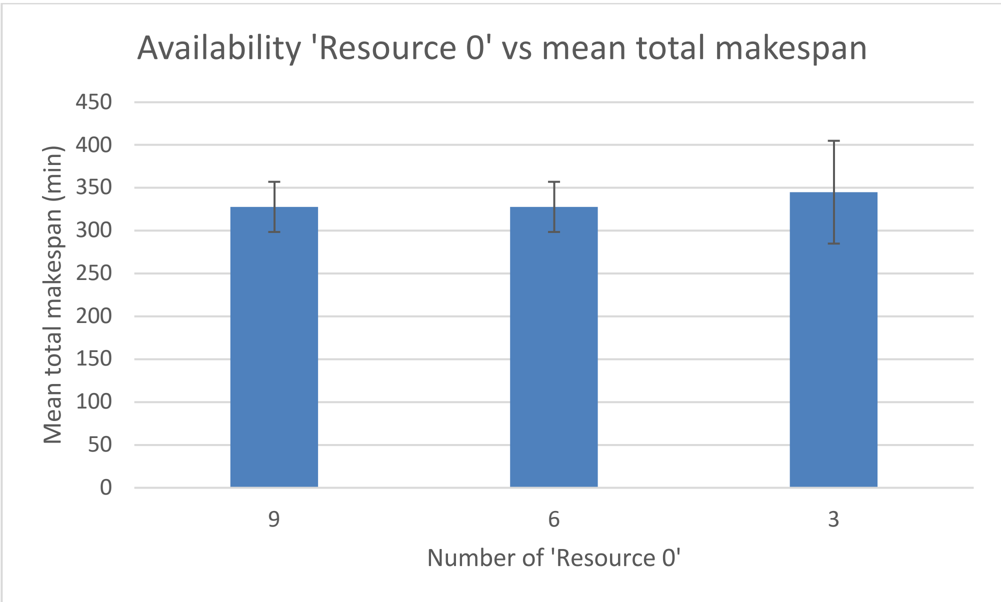

Figure 11: Number of 'Resource 0' vs mean total makespan

Resource 1

As mentioned in Section 5.2, due to the restrictions of the model and the assumption that was made, no further experiments were conducted for 'Resource 1'.

Resource 2

In Table 10 the average results of the sensitivity analysis of 'Resource 2' can be found, followed by figure 12 converting the information visually.

Table 10: Results sensitivity analysis for 'Resource 2'

# 'Resource 2'	Mean total makespan (min)	Max.	Min.	σ	Std. error
1	450	600	300	112.25	45.83
2	336	405	285	43.30	16.34
3	321.43	345	285	29.82	11.27

Table 11: Analysis of Variance on number of available units of 'Resource 2'

Source	Degrees of Freedom	Sum of Squares	Mean Square	F-Stat	p-Value
Between Groups	2	65134.27	32567.13	6.96	0.0062
Within Groups	17	79585.71	4681.51		
Total:	19	144719.98			

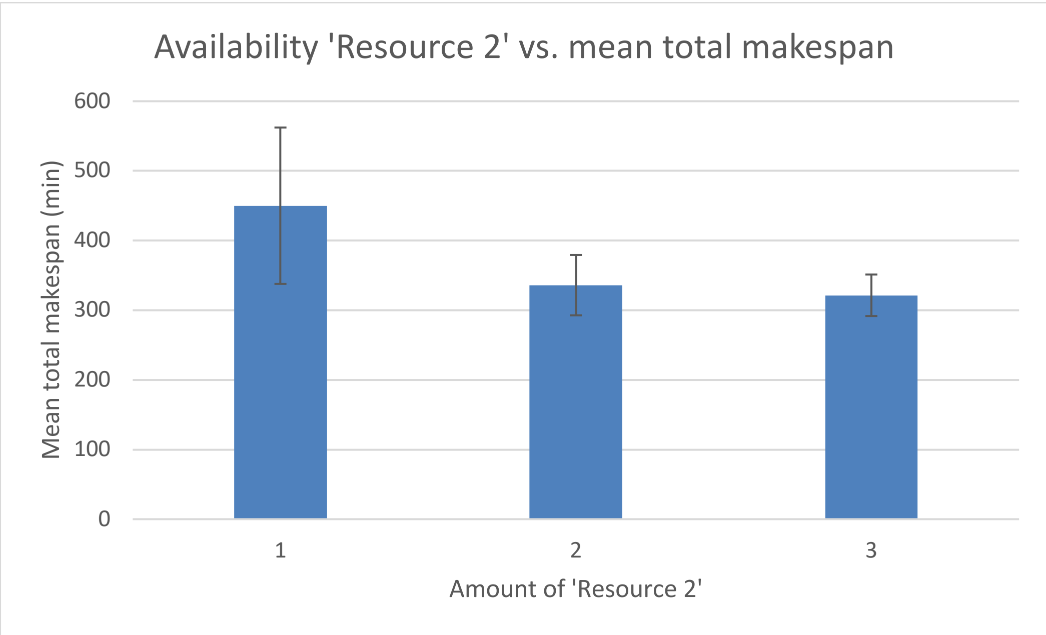

Figure 12: The availability of 'Resource 2' vs. average total makespan

From the information on 'Resource 2' in Table 10 and Figure 12 a similar conclusion can be drawn as for the experimentation on the number of staff. As fewer of 'Resource 2' becomes available, the longer the average total makespan. Not only does the processing time increase, but also the range of processing times, which changes from 60 to 300 minutes, and therefore the standard deviation increases from 29.82 to 112.25. These results provide a good insight into the necessity of this resource and that when one of the resources becomes unavailable it puts pressure on the schedule. This is also evident by the data shown in Table 11 where we have a high value on the F-statistic of 6.96 and a p-value of 0.0062 of which we can confirm that there is a significant effect of the number of available units of 'Resource 2' and the mean total makespan which is unlikely to be due to chance.

For 'Resource 2' we can conclude that there is a high variability when the amount of the resource is changed. It is important for SIVO to be aware of these effects and to consider taking precautions in the availability of this resource.

Resource 3

Lastly, the sensitivity analysis was done for 'Resource 3'. In Tables 12 and 13 and Figure 13, it is clearly shown that there is little to no effect on the average total makespan when the number of 'Resource 3' is changed. In contrast to the previous analysis on the number of machines and 'Resource 2' we obtain a much lower value for the F-stat of 0.06, meaning that the variability between groups is on 0.06 larger than what is expected due to random chance. Also, the p-value is above the 0.05 bound and therefore it can be concluded there is no significance in increasing or decreasing the number of available units of 'Resource 3'.

Table 12: Results sensitivity analysis of 'Resource 3'

# 'Resource 3'	Mean total makespan (min)	Max.	Min.	σ	Std. error
1	327.50	390	285	39.59	16.16
2	321.43	345	285	29.82	11.27
3	321.43	345	285	29.82	11.27
4	321.43	345	285	29.82	11.27

Table 13: Analysis of Variance on number of available units of 'Resource 3'

Source	Degrees of Freedom	Sum of Squares	Mean Square	F-Stat	p-Value
Between Groups	3	172.02	57.34	0.06	0.98
Within Groups	23	23844.66	1036.72		
Total:	26	24016.69			

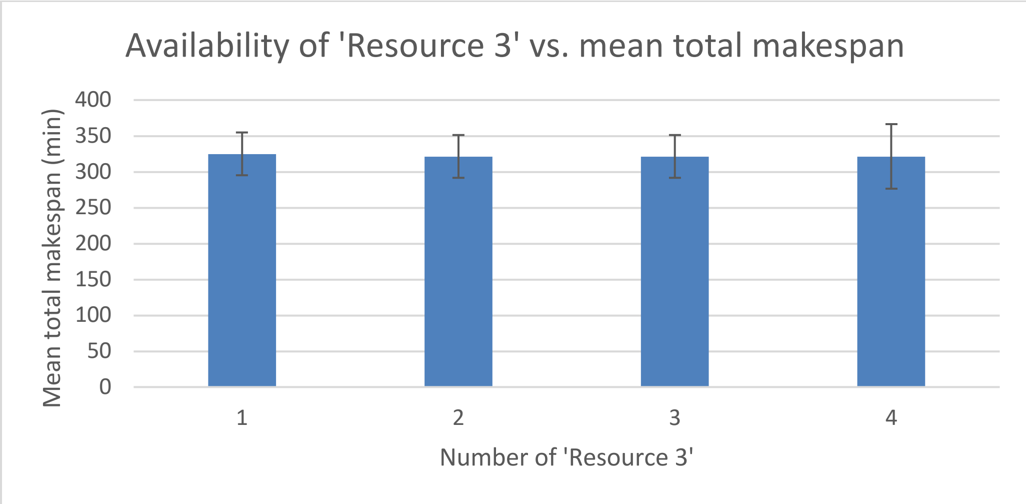

Figure 13: Availability of 'Resource 3' vs. average total makespan

As a conclusion for 'Resource 3' we can confirm that having four, three or two units available for the regular transport routes on peak days is sufficient. Having only one unit can cause troubles as the variability increases and therefore a greater uncertainty in the makespan will exist, having only one unit available caused some results to be infeasible, and SIVO must prevent this from happening in real-life situations.

5.3.3 *Computed schedules*

For all the experiments computed, the model also created Gantt Charts displaying the order of the jobs in a visual manner. From this visual, it could easily be interpreted what the suggested sequence of jobs would be, for each machine individually. The charts can be viewed in Figures 14 to 17. From these Gantt Charts, it is most noticeable that they show signs of the priority rule LPT. Although it is impossible to say that the model made such a 'choice', it is noticeable within the schedules.

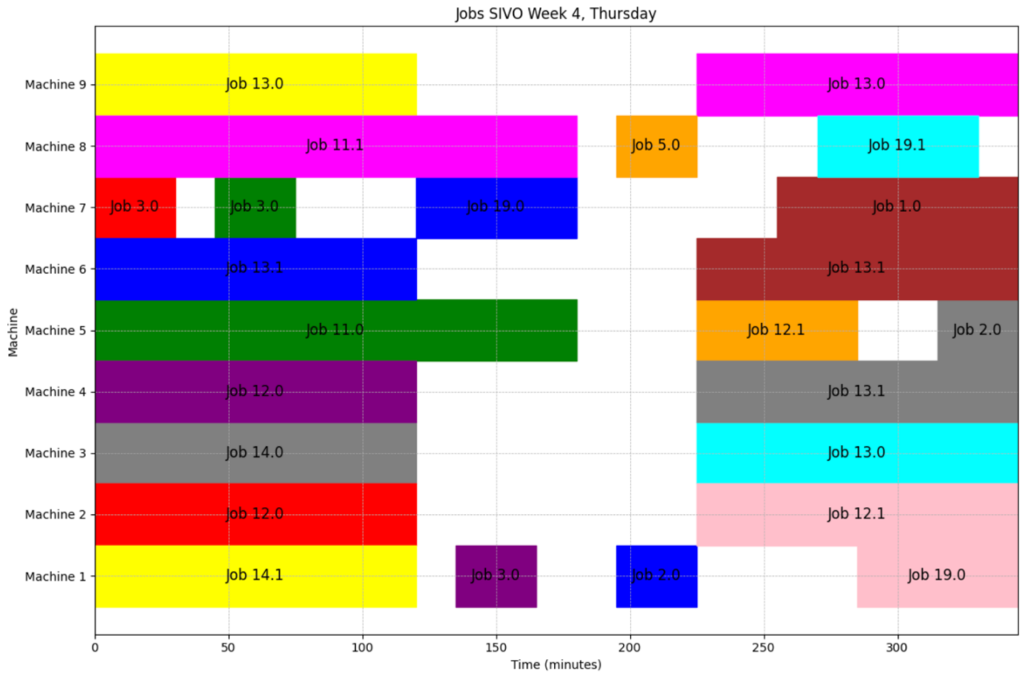

Figure 14: Schedule of week 4, Thursday. 9 machines, 6 units of 'Resource 0', 3 units of 'Resource 1', 2 units of 'Resource 2', 2 units 'Resource 3'.

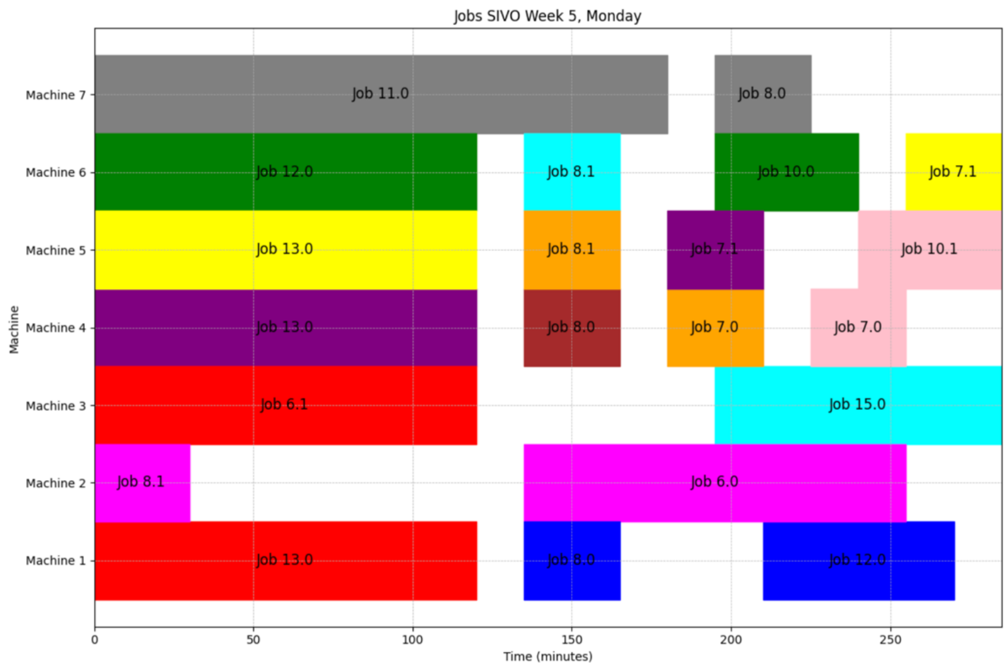

Figure 15: Schedule of week 5 Monday, 7 machines, 6 units of 'Resource 0', 3 units of 'Resource 1', 2 units of 'Resource 2', 2 units of 'Resource 3'.

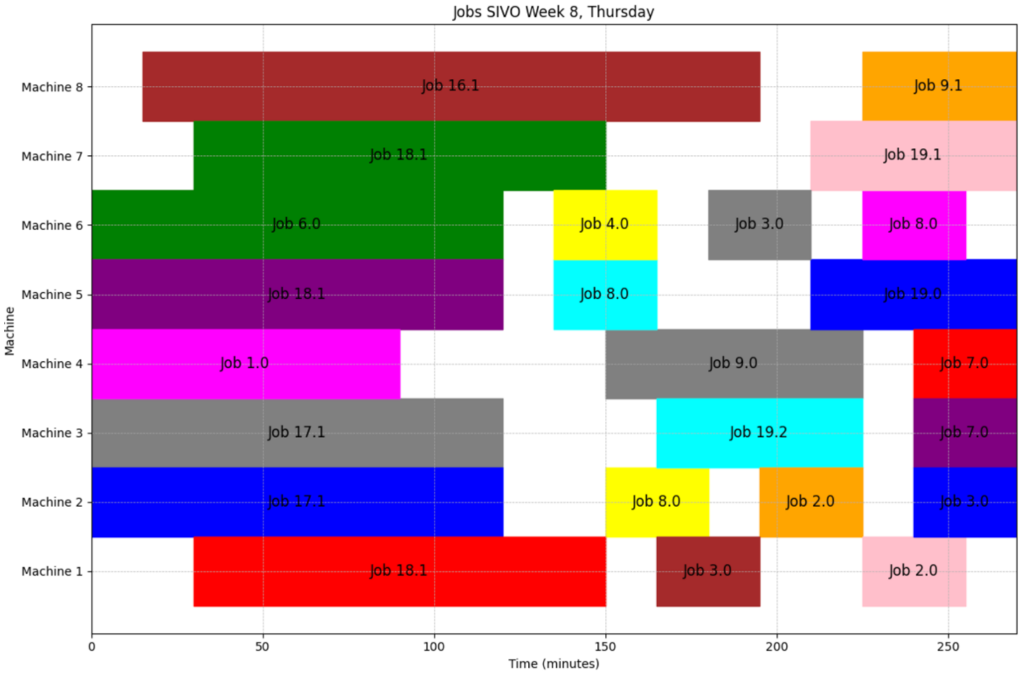

Figure 16: Schedule of week 8 Thursday, 8 machines, 3 units of 'Resource 0', 3 units of 'Resource 1', 3 units of 'Resource 2', 4 units of 'Resource 3'.

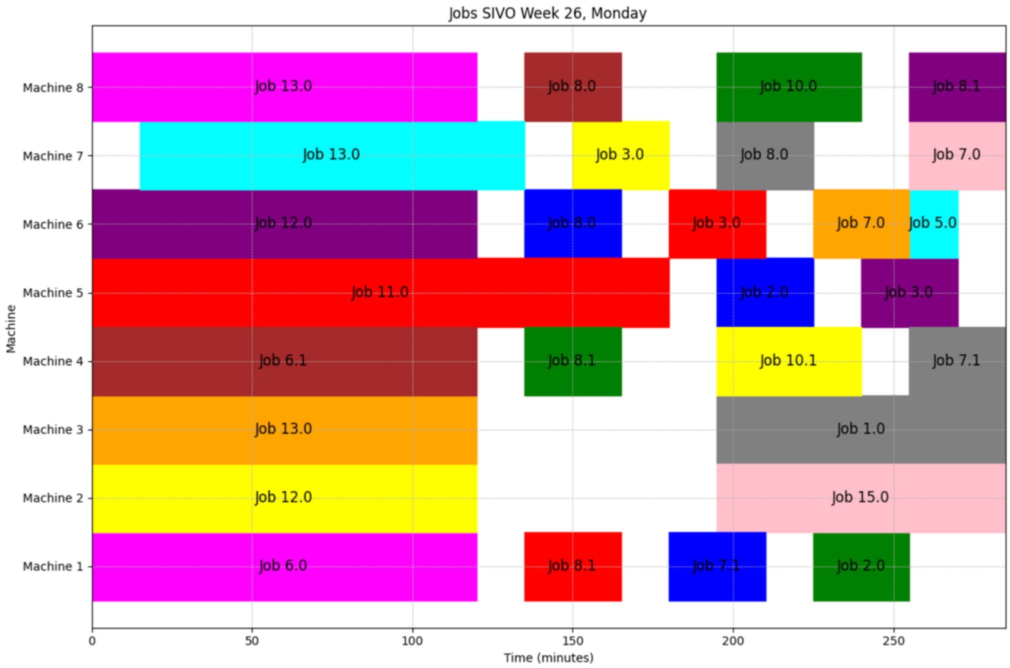

Figure 17: Schedule of week 26 Monday, 8 machines, 6 units of 'Resource 0', 3 units of 'Resource 1', 2 units of 'Resource 2', 3 units of 'Resource 3

6 Conclusion, Recommendation and Further Research

Throughout the following chapter, we will provide a summary of what has been conducted throughout this thesis and then provide a conclusion and recommendation to ultimately discuss the scientific contribution of this thesis and provide some suggestions for further research.

6.1 Summary

This thesis has been conducted at SIVO which is part of the Dutch Ministry of Defence Force and is responsible for the first basic training of petty officers. At SIVO a special team called the Transport Group is responsible for the support of the platoons when they go away from the base for field training. These trainings always take place at one of the four locations they have in their possession. To support these platoons, the Transport Group makes use of resources, trucks, to be able to transport the students and to provide the necessities to set up a camp.

The Transport Group is facing issues with their staffing on peak days. Most often platoons are required to go away for training on Monday and will then arrive back on Thursday. If multiple platoons go into the field, it can be extremely busy for the Transport Group. The Transport Group consists of eleven full-time workers who possess the right to take leave for one or two days per week. For each platoon, a total of six drives are needed to carry out the tasks. Next to that, the other days are off-peak days and often too little work is at hand to keep the present employees busy, resulting in low motivation and a high absence rate. Lastly, SIVO will receive a new type of resource that replaces the current 'Resource 1'. This resource is needed to load the practice gear and drop it off, originally this was always done with a quick loading method, but the new resource requires a different type of loading, and this method takes at least an hour as opposed to the ten minutes it originally took.

To provide SIVO with some support regarding this issue, a MILP model has been created. The goal of the model was to first provide SIVO insights into what the effect is of different numbers of staff and the number of present resources.

6.2 Conclusion

From the proposed model and the evaluation of results we can draw a couple of conclusions, we will discuss these throughout this section.

The model as presented in Chapter 4 provides SIVO with information on how the upcoming situation with the new type of 'Resource 1' can best be dealt with and how different levels of the remaining reusable resources and staffing will have their effect on the makespan on the to be completed tasks during peak days.

What we have seen in the results of the model is when the number of staff or available resources decreases, the range of the total average makespan becomes considerably larger. For example, if there are twelve staff members present than on average the total makespan is around 283 minutes. However, if some staff call in sick or make use of their PAS-day and only eight staff members are present to perform the set of jobs then it would require an average of around 382 minutes. Next to that, the level of uncertainty in the total makespan increased from 70.78 to 121.51. This could mean that in practice eight staff members could need an additional two hours, on top of the already busy schedule, to finish all tasks.

Considering the analysis of the resources we can also draw a few conclusions. Regarding 'Resource 0', used for the transport of students, we witnessed that there is little to no change in the mean total makespan when nine or six units are present was witnessed as the mean total makespan was 327.86 minutes for both. For only three units we saw that the mean total makespan increased to 345 minutes. Next to that, for nine and six units the uncertainty in the mean total makespan was lower, 29.27, than when three units of 'Resource 0' were present, 60.00. For 'Resource 1', no analysis has been conducted as the assumption was made that the trucks are ready and loaded as soon as they are scheduled. Changing the availability would require loading in between jobs and the model does not consider this. 'Resource 2', however, does have a considerable impact on the schedule. This resource is used to drive water tanks and electricity generators to the campsite. When three units are available, as by default, there was a total average makespan of 321.43 minutes opposed to 450 minutes when only one unit was available. Also, the variability in the schedule increased from 29.82 for three units and 112.25 for only one unit. Lastly, we have analysed the impact on the number of available units of 'Resource 3', which is used to drive breakfast, lunch, and dinner to the sites. From this resource, there are normally four units available. Computing the mean total makespan for four, three, and two units resulted in equal figures, namely 321.43 for the makespan and 29.82 for the standard deviation. For the availability of only one resource, the makespan increased a little up to 327.50 but the standard deviation increased to 39.59. Therefore, also for 'Resource 3' we can conclude that the availability of the resources is extensive and that the Transport Group could cope with fewer units.

Furthermore, we can conclude that SIVO has been able to get a greater insight into their actual situation. From the management problem to the problem identification, SIVO gained more information about the actual situation and what the effect is if the situation changes. Through this research, SIVO can now make a considerable judgement on this situation and decide on what further steps to take. Therefore, in the future, the Transport Group should be able to run more smoothly and more efficiently with peak moments that are better coped with. This not only creates a more pleasant working atmosphere if the pressure lowers, but it also becomes more sustainable in the long run. Less staff is likely to become sick due to the stress and only the necessary number of trucks can be made use of. Also, whenever the conitions at SIVO change, they can reconsider the situation and use the proposed model in order to recalculate their needs.

Lastly, we can conclude on the appliance of this model within the entire Ministry of Defence Force. Every unit within the ministry has similar trucks to their availability and has its own Transport Group. Applying this model to multiple departments could lead to an overall more efficient use of the available trucks and staff.

6.2.1 *Recommendation*

For this section, we provide a recommendation to SIVO on how to deal with their issues considering the initial research conducted at the Transport Group department and the MILP model that has been presented in Chapter 4 and evaluated in Chapter 5.

To start off, there are some managerial aspects to the problem that could potentially solve the problem of the current peak moments. In Chapter 1 we have introduced the problem and multiple issues were brought to the surface that are a part of the current problem. One of them pointed out there is a lack of communication between the planning department of SIVO and that of the Transport Group as at times suddenly more platoons were requested to go into the field than was planned. It is recommended to improve this communication to prevent sudden jobs needing to be carried out while not all staff is present. Also, in Chapter 1 we pointed out that staff are taking leave on peak days which puts more pressure on the workload. Therefore, it is recommended that SIVO reconsiders its position on staff taking leave on the peak days; Monday and Thursday. Since during the off-peak days not a substantial amount of work is needed to be carried out, we advise SIVO to investigate what additional work could be assigned to the staff present, this will potentially also solve the problem of low motivation.

Next, there are also some recommendations that can be made based on the model presented in this thesis. Currently, there are eleven staff members hired at the Transport Group, we advise SIVO to not hire additional staff to solve the problem on peak days, as this will increase the problems on off-peak days. In Section 5.3.1 we have seen that eleven staff members is a stable amount as the range of the average total makespan lies between 180 and 345 minutes. Moving to ten or nine staff members is possible, but on extremely busy days this can extensively increase the makespan to a maximum of 510 minutes. Next, we advise SIVO to consider the availability of resources. As was pointed out in Section 5.3.2, the amount available units of 'Resource 2' can put considerable pressure on the total makespan of the schedule, to create more ease within the department and additional units could support SIVO in their goals. Additionally, the sensitivity analysis proved that for both 'Resource 0' and 'Resource 3' a considerably large amount is available whereas the analysis showed not as many are needed. Whereas for 'Resource 0' the mean total makespan remained the same for nine and six units and for 'Resource 0' having only one unit available would potentially impact the makespan.

6.3 Scientific and Practical Contribution

The presented MILP model using reusable resources provides a significant scientific advancement to SIVO. Whereas SIVO at first made use of an ad hoc way of scheduling, they now have real insights into what their schedules could look like in optimal situations. Additionally, SIVO has finally gained insights into their required level of staffing and what the effects are if less staff is present on one of their peak days. Similar knowledge has been presented about the reusable resources that are applicable to the Transport Group and what the effects are if a different number of resources are available. Having been able to simulate all the different scenarios on real-life peak days through Python has provided SIVO with facts they can now implement in their future policies, which is a step in the right direction away from 'guessing' where the problems lie and what the effects of all the different elements within the Transport Group are.

6.4 Suggestions for further research

Within this thesis, there are elements that have not yet been fully explored or that require more attention to strengthen the research even further.

To start off, in Section 4.1 assumptions have been made that relax the current model. When conducting further research, it would be interesting to omit those assumptions and incorporate them into the current model, making it more sophisticated and even more realistic. It is especially interesting to conduct research on the decrease in the sizes of the platoons as 42 weeks of the training progress and students quit the training. If platoons lose a significant number of students over time, maybe only one unit of 'Resource 0' would be needed for certain platoons, resulting in fewer drivers needed on that day. Secondly, we have stumbled upon work-related issues where at times a lack of motivation exists due to the off-peak days. It would be interesting to dedicate more thorough research to these problems and investigate if the staff can have more say in what works better for them. To start off it would be of interest to see what portion of the workweek is dedicated to driving tasks and what portion of the week is 'empty' where no tasks are dedicated. Lastly, the MILP model could be strengthened if uncertainty levels of for example the driving times and staff being suddenly sick are considered. This would require the change from deterministic to stochastic but provides a more real-life result. This would of course not only apply to this single case, but these adaptations to the model could also then be applicable to other similar cases at different companies.

References

Abedinnia, H., Glock, C. H., & Schneider, M. D. (2017). Machine scheduling in production: A content analysis. *Applied Mathematical Modelling*, *50*, 279–299. https://doi.org/10.1016/j.apm.2017.05.016

Aslam, M. U., Nasr, M. M., & Alharkan, I. (2014). Scheduling of jobs on identical parallel machines to minimize total flow time subject to optimal makespan. *Advanced Materials Research*, *903*, 390–395. https://doi.org/10.4028/www.scientific.net/amr.903.390

Beers, B. (2023, March 28). *P-value: What it is, how to calculate it, and why it matters*. Investopedia. https://www.investopedia.com/terms/p/p-value.asp#:~:text=A%20p%2Dvalue%20is%20a,significance%20of%20the%20observed%20difference.

Bürgy, R., & Bülbül, K. (2018). The job shop scheduling problem with convex costs. *European Journal of Operational Research*, *268*(1), 82–100. https://doi.org/10.1016/j.ejor.2018.01.027

Clement, R. (2015). *Mixed Integer Linear Programming Models for Machine Scheduling* (thesis). University of Newcastle , Newcastle (Australia).

De Waal, S. (2022a) Expert Interview Head of Logistics Department, SIVO, 10-10-2022, Ermelo.

De Waal, S. (2022b) Welkom bij de SIVO [slide 4].

de Weerdt, M., Baart, R., & He, L. (2020). Single-machine scheduling with release times, deadlines, setup times, and rejection. *European Journal of Operational Research*, *291*(2), 629–639. https://doi.org/10.1016/j.ejor.2020.09.042

De Wit, W., (2022) Expert Interview Planner at Transport Group, SIVO, 20-10-2022, Ermelo.

Defensie (n.d.) *Een Scania Wissellaadsysteem met een flatrack.* Scania-wissellaadsysteem. Retrieved October 18, 2023, from https://www.defensie.nl/onderwerpen/materieel/voertuigen/scania-wissellaadsysteem

Driver 1, (2022) Interview on the working conditions with one of the drivers of the Transport Group, 19-10-2022, Ermelo.

Fanjul-Peyro, L. (2020). Models and an exact method for the unrelated parallel machine scheduling problem with setups and resources. *Expert Systems* *with Applications: X*, *5*, 100022. https://doi.org/10.1016/j.eswax.2020.100022

Fleszar, K., & Hindi, K. S. (2018a). Algorithms for the unrelated parallel machine scheduling problem with a resource constraint. *European Journal of Operational Research*, *271*(3), 839–848. https://doi.org/10.1016/j.ejor.2018.05.056

Fleszar, K., & Hindi, K. S. (2018b). Algorithms for the unrelated parallel machine scheduling problem with a resource constraint. *European Journal of Operational Research*, *271*(3), 839–848. https://doi.org/10.1016/j.ejor.2018.05.056

Framinan, J. M., Leisten, R., & Ruiz, R. (2014). Overview of Scheduling Methods. In *Manufacturing Scheduling Systems; An Integrated View on Models, Methods and Tools* (pp. 153–190). essay, Springer-Verlag .

Gacias, B., Artigues, C., & Lopez, P. (2010). Parallel Machine scheduling with precedence constraints and setup times. *Computers &; Operations Research*, *37*(12), 2141–2151. https://doi.org/10.1016/j.cor.2010.03.003

Glenn, S. (n.d.). *F statistic / F value: Simple definition and interpretation*. Statistics How To. https://www.statisticshowto.com/probability-and-statistics/f-statistic-value-test/

Google, (n.d.). [Google Maps overview Veluwe area between ‘t Hard and Kootwijk in The Netherlands]. Retrieved June 2 2023.

Laribi, I., Yalaoui, F., Belkaid, F., & Sari, Z. (2016). Heuristics for solving flow shop scheduling problem under resources constraints. *IFAC-PapersOnLine*, *49*(12), 1478–1483. https://doi.org/10.1016/j.ifacol.2016.07.780

Ministerie van Defensie. (2023, August 24). *Scania gryphus-transportvoertuig*. Materieel | Defensie.nl. https://www.defensie.nl/onderwerpen/materieel/voertuigen/scania-gryphus-vrachtwagen

MIP. Python. (n.d.). https://www.python-mip.com/

Olteanu, A.-L., Sevaux, M., & Ziaee, M. (2022). Unrelated Parallel Machine scheduling with job and machine acceptance and Renewable Resource Allocation. *Algorithms*, *15*(11), 433. https://doi.org/10.3390/a15110433

Oujana, S., Amodeo, L., Yalaoui, F., & Brodart, D. (2023). Mixed-integer linear programming, constraint programming and a novel dedicated heuristic for production scheduling in a packaging plant. *Applied Sciences*, *13*(10), 6003. https://doi.org/10.3390/app13106003

PINEDO, M. L. (2022). *Scheduling: Theory, algorithms, and systems*. SPRINGER INTERNATIONAL PU.

Rajaee, A. A., Amindoust, A., & Asadpour, M. (2018). International Conference on Computers &Industrial Engineering (CIE48). Auckland. Retrieved June 16, 2023, from https://www.researchgate.net/publication/330400233_APPLYING_SIMULATED_ANNEALING_ALGORITHM_FOR_PARALLEL_MACHINE_TARDINESS_PROBLEM_SUBJECT_TO_JOB_SPLITTING.

Redactie. (2020). *Eerste Scania Gryphus 8X8 aan Defensie Breda uitgeleverd*. rtvfocuszwolle.nl. Retrieved October 18, 2023, from https://www.rtvfocuszwolle.nl/eerste-scania-gryphus-8x8-aan-defensie-breda-uitgeleverd/amp/ .

Stillman, G., Blum, W., & Biembengut, M. S. (2015). Chapter 13: Problem Solving Methods for Mathematical Modelling. In *Mathematical Modelling in Education Research and Practice Cultural, social and cognitive influences* (pp. 173–183). essay, Springer International Publishing.

Swamidass, P. M., Kuroda, M., Monden, Y., & Morito, S. (2004). Priority Scheduling Rules. In *Encyclopedia of Production and Manufacturing Management* (pp. 527–528). essay, Asakura Shoten.

Swennenhuis, C., Iersel, L. van, Janssen, T., & Aardal, K. (2018). *Parallel Machine Scheduling: with Partition Constraints* (thesis).

Taillard, E. (1990). Some efficient heuristic methods for the flow shop sequencing problem. *European Journal of Operational Research*, *47*(1), 65–74. https://doi.org/10.1016/0377-2217(90)90090-x

Xu, J., & Nagi, R. (2013). Identical parallel machine scheduling to minimise makespan and total weighted completion time: A column generation approach. *International Journal of Production Research*, *51*(23–24), 7091–7104. https://doi.org/10.1080/00207543.2013.825379

Appendix A: The Python codes for the model

```
[ ]: from mip import Model, xsum, minimize, BINARY, OptimizationStatus, INTEGER, GUROBI, CONTINUOUS
     import matplotlib.patches as patches
     import matplotlib.pyplot as plt
     import numpy as np
     import pandas as pd
     from datetime import datetime
     import openpyxl
```

```
[ ]: # Function to convert time in HH:MM:SS format to minutes
     def convert_to_minutes(time_str):
         h, m, s = map(int, time_str.split(':'))
         real_minutes = round(h * 60 + m + s / 60)
         return real_minutes // 15  # Divide by 15 to convert to timesteps of 15 minutes
```

```
[ ]: #add decision variables and create the model
     model = Model("MyModel", sense=minimize, solver_name='CBC')

     x_ijt = [[[model.add_var(var_type=BINARY) for t in range(T)] for j in range(n)] for i in range(m)] # 1 if job j starts on machine i at time t, 0 otherwise
     y_jrt = [[[model.add_var(var_type=BINARY) for t in range(T)] for r in range(k)] for j in range(n)] # 1 if resource type r is used for job j at time t, 0 otherwise
     s_ij = [[model.add_var(var_type=CONTINUOUS) for j in range(n)] for i in range(m)] # Starting time of job j on machine i
     c_j = [model.add_var(name=f"c_{j}", var_type=INTEGER, lb=0, ub=T) for j in range(n)] #completion time of a job
```

```
[ ]: C_max = model.add_var(var_type=INTEGER) # maximum completion time (makespan)
     model.objective = C_max
```

```
[ ]: #Adding the constraints
```

```
[ ]: # Constraint to ensure makespan is greater than or equal to completion time of all jobs
     for i in range(m):
         for j in range(n):
             model += C_max >= s_ij[i][j] + p_j[j]
```

```
[ ]: #relationship between xijt and sij, extracting the starting time of job j
     for i in range(m):
         for j in range(n):
             model += s_ij[i][j] == xsum(t * x_ijt[i][j][t] for t in range(T))
```

```
[ ]: # Each job j starts on exactly one machine at exactly one time stage
     for j in range(n):
         model += xsum(x_ijt[i][j][t] for i in range(m) for t in range(T)) == 1
```

```
[ ]: # no 2 jobs can be processed at the same time on the same machine
     for i in range(m):
         for t in range(T):
             model += xsum(x_ijt[i][j][t_prime] for j in range(n) for t_prime in range(max(0, t-p_j[j]+1), min(T, t+p_j[j]))) <= 1
```

```
[ ]: # Resource usage constraint
     for r in range(k):  # where k is the number of resource types
         for t in range(T):  # T is the number of time periods
             model += xsum(y_jrt[j][r][t] for j in range(n)) <= l_r[r]
```

```
[ ]: # Linking job starting times to resource usage
     for j in range(n):
         for r in range(k):
             if r == r_j[j]:
                 for t in range(T - p_j[j] + 1):
                     for tau in range(p_j[j]):
                         model += y_jrt[j][r][t + tau] >= xsum(x_ijt[i][j][t] for i in range(m))
```

```
[ ]: #assignment of resource type
     for j in range(n):
         for r in range(k):
             for t in range(T):
                 if r != r_j[j]:
                     model += y_jrt[j][r][t] == 0
```

```
[ ]: #release constraint
     for j in range(n):
         r = r_j[j]  # Considering that each job is associated with one resource
         for t in range(T):
             # If a job starts at time `t_prime`, it would use the resource until `t_prime + p_j[j] - 1`
             # set y_jrt to 0 for times `t` where `t > t_prime + p_j[j] - 1`
             model += y_jrt[j][r][t] <= xsum(x_ijt[i][j][t_prime] for i in range(m) for t_prime in range(max(0, t - p_j[j] + 1), t+1) if t <= t_prime + p_j[j] - 1)
```

```
[ ]: #constraint to first conduct tasks if present (breakfast driving)
     priority_job_ids = [5, 11, 17]
     priority_indices = [job_ids.index(job_id) for job_id in priority_job_ids if job_id in job_ids]

     for i in range(m):  # loop over machines
         for j in range(n):  # loop over all jobs
             if j not in priority_indices:  # for all non-priority jobs
                 for priority_index in priority_indices:  # loop over priority jobs
                     # the start time of priority jobs should be less than or equal to the start time of non-priority jobs
                     model += s_ij[i][priority_index] <= s_ij[i][j]
```

[ ]:
```
# Maximum time difference allowed between the start times
D = 30

# Determine the specific job IDs for parallel execution
specific_job_ids = [4]
# Find the corresponding indices for those job IDs in data
specific_job_indices = [index for index, job_id in enumerate(job_ids) if job_id in specific_job_ids]

# Define the constraint using those indices
for idx1 in specific_job_indices:
    for idx2 in specific_job_indices:
        if idx1 < idx2:  # Compare each pair
            for i in range(m):
                for j in range(m):
                    model += s_ij[j][idx2] - s_ij[i][idx1] <= D  # Ensure the start time difference is less than or equal to D
                    model += s_ij[i][idx1] - s_ij[j][idx2] <= D  # This constraint ensures the order can be the other way around too
```

[ ]:
```
# Maximum time difference allowed between the start times
D = 30

# Determine the specific job IDs for parallel execution
specific_job_ids = [16]
# Find the corresponding indices for those job IDs in your data
specific_job_indices = [index for index, job_id in enumerate(job_ids) if job_id in specific_job_ids]

# Define the constraint using those indices
for idx1 in specific_job_indices:
    for idx2 in specific_job_indices:
        if idx1 < idx2:  # Compare each pair
            for i in range(m):
                for j in range(m):  # Need another loop as the jobs can be on different machines
                    model += s_ij[j][idx2] - s_ij[i][idx1] <= D  # Ensure the start time difference is less than or equal to D
                    model += s_ij[i][idx1] - s_ij[j][idx2] <= D  # This constraint ensures the order can be the other way around too
```

[ ]:
```
# Maximum time difference allowed between the start times
D = 30

# Determine the specific job IDs for parallel execution
specific_job_ids = [22]

# Find the corresponding indices for those job IDs in your data
specific_job_indices = [index for index, job_id in enumerate(job_ids) if job_id in specific_job_ids]

# Define the constraint using those indices
for idx1 in specific_job_indices:
    for idx2 in specific_job_indices:
        if idx1 < idx2:  # Compare each pair only once
            for i in range(m):
                for j in range(m):  # Need another loop as the jobs can be on different machines
                    model += s_ij[j][idx2] - s_ij[i][idx1] <= D  # Ensure the start time difference is less than or equal to D
                    model += s_ij[i][idx1] - s_ij[j][idx2] <= D  # This constraint ensures the order can be the other way around too
```

[ ]:
```
# Maximum time difference allowed between the start times
D = 30

# Determine the specific job IDs for parallel execution
specific_job_ids = [10]

# Find the corresponding indices for those job IDs in your data
specific_job_indices = [index for index, job_id in enumerate(job_ids) if job_id in specific_job_ids]

# Define the constraint using those indices
for idx1 in specific_job_indices:
    for idx2 in specific_job_indices:
        if idx1 < idx2:  # Compare each pair only once
            for i in range(m):
                for j in range(m):  # Need another loop as the jobs can be on different machines
                    model += s_ij[j][idx2] - s_ij[i][idx1] <= D  # Ensure the start time difference is less than or equal to D
                    model += s_ij[i][idx1] - s_ij[j][idx2] <= D  # This constraint ensures the order can be the other way around too
```

[ ]:
```
# Objective function to minimize makespan
model.optimize()

for i in range(m):
    for j in range(n):
        print(f"s_{i}{j} (immediately after optimization) = {s_ij[i][j].x}")

if model.status == OptimizationStatus.OPTIMAL:
    print('Optimal solution found!')

    # Display machine assignments, start times, and finish times
    for j in range(n):
        for i in range(m):
            for t in range(T):
                if x_ijt[i][j][t].x > 0.5:  # assuming binary values, so a value > 0.5 implies it's 1
                    start_time_real = t * 15   # converting to real-world minutes
                    finish_time_real = start_time_real + p_j[j] * 15  # converting to real-world minutes
                    print(f"Job {job_ids[j]} starts on machine {i+1} at minute {start_time_real} and finishes at minute {finish_time_real}. Resource used: {data.loc[j, '
                    break


    # Display resource usage for each job
    for j in range(n):
        for r in range(k):
            for t in range(T):
                if y_jrt[j][r][t].x > 0.5:
                    print(f"Job {job_ids[j]} utilizes resource type {r} at minute {t*15} to {t*15 + 15}")  # Convert to minutes


elif model.status == OptimizationStatus.FEASIBLE:
    print('Feasible solution found, but not necessarily optimal.')
else:
    print('No solution found.')
```

[ ]:

```python
def plot_simple_gantt_chart(s_ij, x_ijt, p_j, job_ids, job_colors=None):
    # Define default colors if none are provided
    if job_colors is None or len(job_colors) < len(job_ids):
        default_job_colors = ["gray", "blue", "green", "red", "purple", "yellow", "cyan", "magenta", "orange", "pink", "brown"]
        job_colors = default_job_colors * (len(job_ids) // len(default_job_colors)) + default_job_colors[:len(job_ids) % len(default_job_colors)]

    plt.figure(figsize=(12, 8))

    earliest_start = 0  # Explicitly set to 0 to include jobs starting at time 0
    latest_end = max([s_ij[i][j].x * 15 + p_j[j] * 15 for i in range(len(s_ij)) for j in range(len(job_ids))])

    for j, job_id in enumerate(job_ids):
        for i in range(len(s_ij)):
            # Check if job 'j' is scheduled on machine 'i'
            if any(var.x > 0.5 for var in x_ijt[i][j]):
                start_time = s_ij[i][j].x * 15  # Convert to minutes
                end_time = start_time + p_j[j] * 15  # Convert to minutes

                plt.fill_between([start_time, end_time], i-0.5, i+0.5, color=job_colors[j], label=f"Job {job_id}")
                plt.text((start_time + end_time) / 2, i, f"Job {job_id}", va='center', ha='center', color='white', fontsize=8)

    # Set the x-axis limits and labels
    plt.xlim(earliest_start, latest_end)
    plt.xlabel("Time (minutes)")
    plt.ylabel("Machine")
    plt.yticks(list(range(len(s_ij))), [f"Machine {i+1}" for i in range(len(s_ij))])
    plt.title("Jobs SIVO Week 8, Thursday")
    plt.tight_layout()
    plt.grid(True, which='both', linestyle='--', linewidth=0.5)
    plt.show()

# Call the function with appropriate parameters
plot_simple_gantt_chart(s_ij, x_ijt, p_j, job_ids)
```