CHALMERS | UNIVERSITY OF GOTHENBURG

MASTER'S THESIS

Mathematical modelling of the scheduling of a production line at SKF

MAHSA FAIZRAHNEMOON

Department of Mathematical Sciences
Division of Mathematics
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2012

Thesis for the Degree of Master of Science

Mathematical modelling of the scheduling of a production line at SKF

Mahsa Faizrahnemoon

Department of Mathematical Sciences
Division of Mathematics
Chalmers University of Technology and University of Gothenburg
SE – 412 96 Gothenburg, Sweden
Gothenburg, September 2012

Matematiska vetenskaper
Göteborg 2012

Mathematical modelling of the scheduling of a production line at SKF

Mahsa Faizrahnemoon

September 4, 2012

Abstract

The main purpose of this thesis project is to find the required sizes of the buffers in one of the future roller production channels in SKFs factory in Gothenburg. An integer linear programming model for finding the best schedule for the channel is developed. The model minimizes sum of the lead times for the batches of rollers in the channel. Thereby, the total time that the rollers are kept in the buffers is minimized, which in turn minimizes the average demand for the volumes of the buffers. Since the mathematical model is time-indexed, it represents an approximation of the real scheduling problem. Therefore, post-processing is used to improve the solution obtained. We study a case from the channel at SKF and present results in the form of optimal production schedules and number of pallets of rollers in the buffers during the planning period. A comparison is made between optimal schedules obtained from the time-indexed model with different time step intervals.

Keywords : Production Scheduling - Integer Linear Programming - Heuristic - Optimization Modelling

To my parents and my little sister

Acknowledgements

This master thesis was performed in collaboration with SKF Sverige AB. Here, I would like to thank my examiner Ann-Brith Strömberg for her help and her contribution to extend my knowledge. I also wish to thank my supervisor at SKF, Anders Leire, for being very supportive and friendly during this time. I would like to thank my friends at the optimization division at the Department of Mathematical Sciences and my co-workers at SKF for their help and for answering my questions.
Finally, I would like to express my deepest appreciation to my family, who have always supported me strongly even from long distance, for their patience, guidance, and sympathy.

Contents

1 Introduction

1.1 Background

In order to extend a future roller production channel in SKF's factory in Gothenburg, Sweden, a number of the current machines will be replaced by new production cells. The existing machines are old and might be dangerous for the workers, while the new cells are not only safer but also more efficient. Moreover, the new cells produce rollers of higher quality in comparison with the current machines. The main buffer of the current channel will be removed and replaced by several new buffers, which should be built at locations between the machines.

1.2 Aims and limitations

The aim of this thesis project is to find sufficiently large sizes of the buffers in the future production channel. The buffers should be as small as possible since the area between the machines is limited. In case that the required area for a buffer is more than the available area at that location, a two-store buffer should be built, and an elevator will be used to access the basement.

The leadtime of a batch is the time duration for processing the batch in the channel. The sum of the leadtimes of all the batches should be as small as possible. In other words, when a batch of rollers enters the first machine in the channel, it should leave the last machine of the channel as soon as possible.

It is assumed, in the beginning of each planning period, that all batches of raw material are available, and that all machines have already been reset and are ready to start processing the appropriate batches. It is also asumed that all machines are always intact, and never broken during the scheduling time. In fact, no delay for repairing the machines is taken into account in this thesis. These cases are, however, not the fact in a real planning situation. Therefore, the mathematical models developed have to be adjusted to become operational. In practice it is possible to include these limitations in the model.

1.3 Methodology

In this project, time is considered to be discrete since the mathematical models for scheduling problems based on discrete time steps typically require fewer variables and constraints, as compared to continuous time models. This leads to the computing time required being considerably shorter for the discrete time models. The length of the time steps is defined by the user, and the data must be adapted to the length of the time steps. The data of one year is scattered over the months of the year, and the methodology is run for each month separately. The goal of this thesis is acheived in four methodological steps. The first step is a heuristic which yields a feasible schedule and provides a suitable number of time steps to be used in the mathematical model. In the second step, an ILP model with discrete time steps, being an approximation of the real problem (with continuous time), is solved to find an optimal schedule according to the length of the time steps. The accuracy of the optimality depends on the length of the time steps. This means that by choosing shorter time steps, the result becomes more accurate and realistic. A squeezing of the production schedule by ignoring the discretization is executed in the third step. This results in a possibly tighter schedule than the one found in step two. Finally, the schedules of the buffers over time are obtained from the squeezed production schedules of the machines and the number of pallets in the buffers during the plannning period are calculated.

1.4 Outline

Section 2 describes the current and the future production channels at SKF. Iluustrations of and information about the different lines and machines in the channel are provided. Section 3 presents the mathematical tools used to model the production channel: linear programming (LP), integer linear programming (ILP), and mathematical modelling of scheduling problems. Section 4 presents the heuristic algorithm, which is used to find a feasible solution, the integer linear programming (ILP) model used to find an optimal solution to the discretized problem, and the squeezing procedure that possibly improves the optimal schedule from the ILP model possibly more compact. In Section 5 a small scale example, similar to the channel at SKF, is presented and solved. Since the number of machines and batches in the channel at SKF is huge, this little example is helpful in understanding the problem and its properties. Section 6 presents the schedules obtained for a number of planning cases for the future channel at SKF. The last sections include the conclusions of the research and suggestions for future work.

2 Definition and description of the problem

The production channel under consideration has the role of producing different types of rollers with different shapes, sizes, and materials. These rollers will finally be installed in the appropriate bearings, also produced by SKF.

The first process on the raw rollers, which are imported, is *pressing*. There are three different pressing machines in the channel under consideration; see Figure 1. Each pressing machine can perform the pressing process on special types of rollers according to the size and material. Therefore, each type of roller passes its appropriate pressing machine.

After pressing, the *heat treatment* process should be performed. As shown in Figure 1, there are three machines in the the production flow for the heat treatment process. Two heat treatment machines are available in the RQ building, each of which can perform the heat treatment process on special types of rollers. There is also an external heat treatment machine, which is located outside the factory. As much as possible of the heat treatment is processed by the two internal machines, but the capacity of these machines is limited. Therefore, some batches have to be sent to the external heat treatment which requires a much longer processing time because of the transportation. It should also be mentioned that because of the limited capacity of the heat treatment machines, a so called big batch of rollers — containing more than 100,000 rollers — should be split into two batches for the heat treatment process.

The following processes are the *rough grinding* and the *end forming*. Several types of machines are included in the current channel performing these processes. Channels of rough grinding and end forming are shown in Figure 1. Since the end forming machines of the current production channel are slow, a large RQ (unfinished roller) buffer is currently needed between the rough grinding and the end forming machines. Batches of RQ rollers are kept in the RQ buffer before the end forming process. It constitutes the main buffer in the channel and it is occupying a huge area.

The final process is *fine grinding* which includes several channels, each of which includes several machines. The channels of fine grinding are shown in Figure 1. A description of the details of the machines in each fine grinding channel is not necessary here. All machines in a certain channel of fine grinding perform their respective processes on the same types of rollers. Therefore, if one specific batch enters the first machine in a channel of fine grinding, it leaves from the last machine in the same channel. In other words, batches are not transferred between different channels of fine grinding.

In order to expand the production capacity in the factory, the plan is to install four new production cells replacing the current machines in the channels of rough grindig and end forming. These new cells are faster, and occupy less space in the building. The new cells can perform the whole processes of rough grinding and end forming in one stage and they are capable of processing all types of rollers of different sizes and materials. It means that after the heat treatment process on a batch of any type of roller, as soon as one of the new cells is empty, the batch enters the empty cell. As previously mentioned, the large RQ buffer is currently needed between the end forming and rough grinding since the end forming machines are old and slow. By replacing the old machines by new production cells, the RQ buffer is no longer necessary. Therefore, another plan for expanding the factory is to fill the area of the large RQ buffer by new machines.

In the future flow, after replacinging the old channels of rough grinding and end forming by new cells, a large area in the building will be available for new buffers. Instead of the RQ buffer, several small buffers will be built in the area between the machines and the new cells: one buffer before each heat treatment machine, four buffers before the new cells (one buffer for each cell), and seven buffers before the fine grinding channels. Since the fine grinding includes seven lines of machines (R4, R5, R7, R8, R9, R11, and R12), one buffer should be placed before each line of fine grinding. Figure 2 indicates the future production channel with the new cells and the future buffers.

One objective of this project is to find the required sizes of these buffers. The buffers should be as small as possible because the available area is limited. On the other hand, the buffers should be large enough to comprise enough space for keeping the unfinished rollers. If the required area for a certain buffer is more than the available space, another buffer will be built in the basement and an elevator should be installed for providing access to this buffer. The lead time defined as the time required to finish the processing of all the rollers in all the machines, should be as short as possible. Minimizing the leadtime and keeping small buffers constitute the main goals of the project. These goals are related to each other: If the production line is working very fast, the leadtimes are short and the batches of rollers are kept in the buffers for short time which leads to smaller required sizes of the buffers.

3 Mathematical theory

Optimization is a field within applied mathematics and deals with using mathematical models and methods to find the best possible alternative in decision making situations. In fact optimization is the science of making the best possible decision. The expression "best" means that an objective should be defined, and "possible" indicates that there are some restriction for the decision ([7]).

In general, an optimization problem can be formulated as to

$$\begin{aligned} \text{minimize} \quad & f(x), \\ \text{subject to} \quad & x \epsilon X, \end{aligned} \tag{1}$$

in which $f : R^n \mapsto R$ is the objective function and $x \epsilon R^n$ are the decision variables. The set $X \subset R^n$

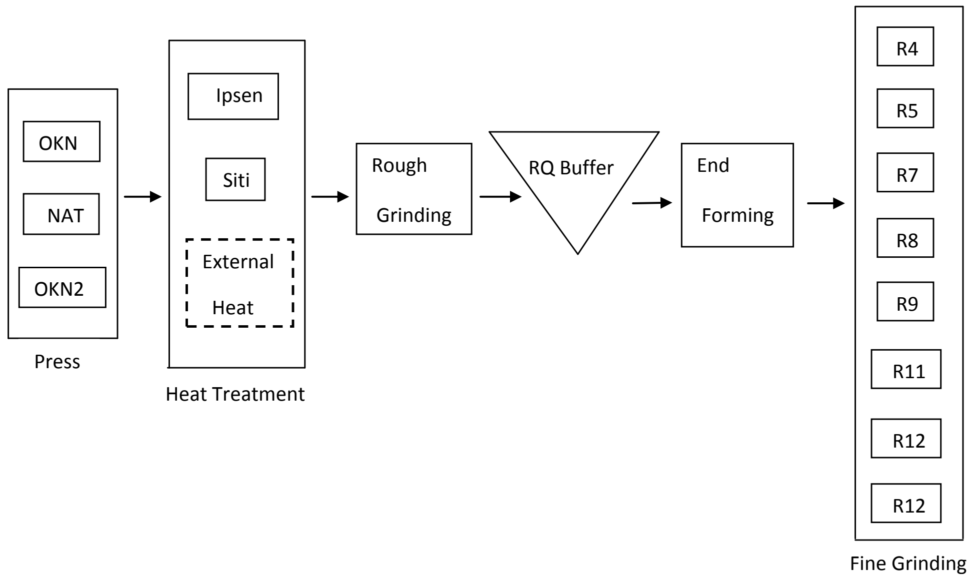


Figure 1: The structure of the current production flow. OKN, OKN2, and NAT are pressing machines. Ipsen and Siti are the two heat treatment machines available in the building. The External Heat is located outside the factory. The Rough Grinding, and the End Forming are two channels. The Fine Grinding process includes several channels. The RQ buffer is indicated by a triangle

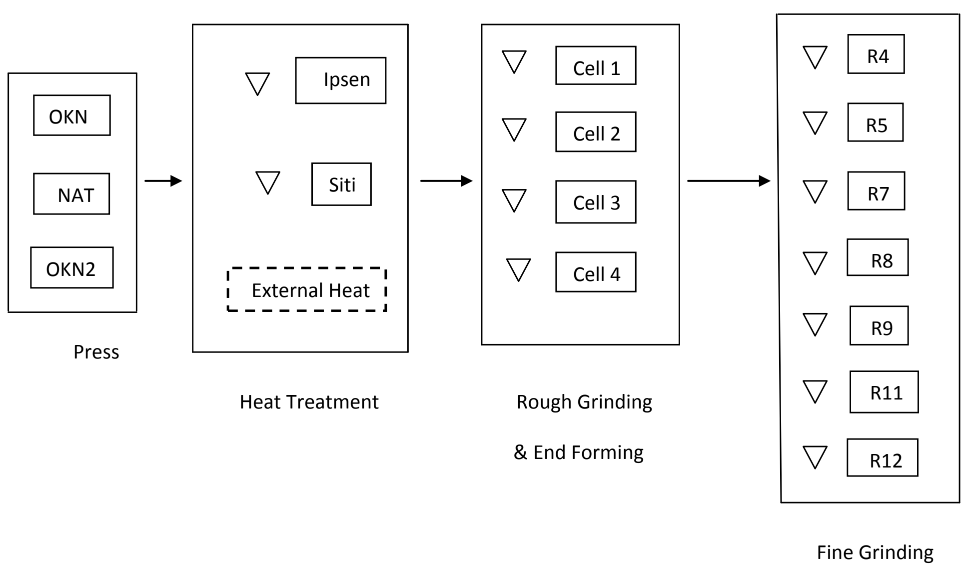


Figure 2: The structure of the future production flow. The four new cells perform the Rough Grinding and theEnd Forming. The future buffers located between the machines are indicated by triangles.

is the set of feasible solutions of the problem, and it usually is expressed by constraints ([7]).

3.1 Linear programming

In linear programming (LP) problems the objective function and all the constraints of problem (1) are linear functions. An LP problem can be written as to

$$\text{minimize} \quad z = \sum_{j=1}^{n} c_j x_j, \tag{2a}$$

$$\text{subject to} \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i, \qquad i = 1, ..., m, \tag{2b}$$

$$x_j \geq 0, \qquad j = 1, ..., n, \tag{2c}$$

where c_j denotes the coefficient for the variables, x_j, in the objective function, a_{ij} is the coefficient of the variable x_j in constraint i, and b_i is the right hand side coefficient for constraint i ([7]).

A set of all variables $x_1, ..., x_n$ satisfying all the constraints (2b) and (2c) is called a feasible point or a feasible vector. The set of all such points constitutes the feasible region or the feasible space. A linear programming problem can be stated as follows: Among all feasible vectors, find one that minimizes (or maximizes) the objective function, the optimal solution ([1]).

An LP problem can be written with both equality and inequality constraints, and the variables might be restricted with respect to sign. The variables can be non-positive, non-negative, or even non-restricted (free variables ([7]).

A linear program is said to be in standard from if all restrictions are equalities and all variables are nonnegative. The simplex method, which is created by G. B. Dantzig to solve linear programming problems, is designed to be applied only after the problem is put in the standard form ([1]).

To represent an optimization problem as a linear program, several assumptions, such as proportionality, additivity, divisibility, and determinism, are needed. The proportionality means that the contribution of the given variable, x_j, to cost is $c_j x_j$ and to the ith constraint is $a_{ij} x_j$. The assumption of additivity ensures that there are no substitution or interaction effects among the activities. The divisibility assumption ensures that the decision variables can be divided into any fractional levels so that non-integer values for the decision variables are allowed. The deterministic assumption means that all coefficients of the linear programming problem should be known deterministically. Any probablistic or stochastic elements are assumed to be approximated by some deterministic equivalents ([1]).

The simplex method checks the border of the feasible area to find the optimal solution of the linear programming problem. It starts by an extreme point, which means that it cannot be a convex combination of two other feasible points, and searches for the next extreme point such the value of the objective function is more desirable. It checks all the adjacent extreme points and choses one of them as the next extreme point. This method is continued until the current extreme point has the most desirable objective value compared to the adjacent extreme points.

3.2 Integer linear programming

Some applications can only be modeled using integer or binary variables. An integer linear programming (ILP) problem is an optimization problem in which at least one variable is restricted to integer or binary values. If all variables are integer, the problem is a pure integer programming problem, if some of the variables are integer and the other variables are continuous, we have a mixed integer linear programming (MILP) problem . Different reasons may force us to define integer or binary variables such as modeling of Yes/No decisions, modeling of fixed costs, or when only a discrete set of values is allowed.

In a Linear Programming problem, all variables may take fractional values, the feasible area is convex and the objective function is continuous. The feasible area of an integer programming problem consists of a discrete set of points and is as such a non-convex set.

It is always possible to find the optimal solution of an LP problem in an extreme point of the feasible area, while it is unlikely to find the optimal solution of an ILP Problem in a corner point or along an edge of the feasible area of the LP Relaxation in which the integrality is relaxed. Sometimes, it even happens that the optimal solution of the ILP Problem is nearly far away from the optimal solution of the LP Relaxation. It is of course not possible to round the optimal values of the LP Relaxation to get an optimal ILP solution since these rounded values might not even be feasible.

One may think that it is easier to solve ILP problems than LP problems, since LP problems have continuous variables, and they have infinite number of feasible solutions, while ILP problems have a finite number of feasible solutions. For small ILP problems it might be possible to enumerate all solutions, check feasibility, and simply choose the best. However, it is not a reasonable approach and it does not work for big ILP problems.

In general, IP problems are much harder to solve than LP problems. ILP problems are not convex because of their discrete feasible region. This fact makes them more difficult to solve. LP problems are easier to solve because we know that the optimal solution always appears in an extreme point of the feasible region. This property is used in the Simplex method to solve LP problems.

3.3 Scheduling

One of the shop scheduling problems that finds the optimal sequences of a given jobs on a given set of machines is job shop scheduling; see [3]. The first ILP formulations of job shop scheduling were formulated by Manne ([8]), Wagner ([14]), and Bowman ([2]). They have modeled the dimension of time in three different ways.

Manne ([8]) considered the problem of sequencing jobs with presedence constraints, that indicate which operation must precede another operation, on a single machine. In Manne's ILP model, the decision variables represent by y_{jq} equal 1, if job j precedes job q, and 0 otherwise. Fattahi et al. ([4]), Özgüven et al. ([17]), Roslöf et al. ([9]), and Zhu et al. ([16]) used this type of variables in their modelling approaches.

In Wagner's research ([14]) only the ordering of the jobs on each machine is considered. In his model, a decision variable equals 1 if the corresponding job is scheduled in a specific resource at a specific order-position, and 0 otherwise. Stafford et al. ([11]) have compared models using Manne's and Wagner's approaches to minimize the makespan, i.e., to minimize the time at which all jobs are completed.

Bowman, in [2], has divided the planning period into an integer number of time periods with equal length. The decision variables used in this approach equal 1 if the corresponding job is processed by a specific resource during a specific time period, and 0 otherwise. Kedad-Sidhoum et al. ([6]), have used this type of variables in their model.

Another way of modelling time is considered by Sousa et al. ([10]), Wolsey ([15]), and Thörnblad ([12]). Decision variables equal 1 if the corresponding job starts at a specific discrete time step, and 0 otherwise. Using this kind of binary variables in ILP problems leads to very large models with a lot of variables and constraints. On the other hand, models using this kind of variables typically yeild better lower bounds (for the objective value) than other MILP models of scheduling problems; see [13] and [12]. A time-indexed model using the same kind of variables is developed in Section 4.2 to schedule the production line under consideration at SKF.

Minimizing the makespan is the objective which is the most frequently used in scheduling problems; see [5]. Other common objectives are related to the job's earliness, tardiness, or completion times, which is used in the time-indexed model in Section 4.2. Inventory holding costs associated with the jobs can also be an objective for scheduling problems.

4 Modelling

When solving scheduling problems, the solution time may increase exponentially with the number of variables and constraints. By assuming that time is discrete, and not continuous, the required number of variables and constraints is typically smaller than when modelling using continuous time variables. A mathematical model in which time is considered to be discrete is developed which produces an approximate solution. By choosing shorter time steps (and making the discretization approximation closer to a continuous time model), the result will be closer to an optimal one and also more realistic, but the solution time will be longer.

4.1 A mathematical model for distributing the batches over the year

In order to be able to solve the scheduling model in acceptable time, the number of variables and constraints should be decreased. Instead of running the scheduling model for one year data, the model can be run for the months of the year separately. The number of batches produced in a month is much smaller than the number of batches produced in one year. Therefore, the number of constraints and variables of the scheduling model will be smaller than the model for one year data. The mathematical model represented in this section developes the appropriate input data for each month of the year. The result of this model will be used as data for scheduling model to be run for the months. This preallocation of the batches to specific months may, however, remove or exclude the optimal solution from the feasible set. Hence the resulting final schedule may be suboptimal.

4.1.1 Definitions

Several types of rollers are produced in the production channel. In Table 1 the set of all types of rollers is indicated by $\mathcal{R}$, and the set of months of the year by $\mathcal{Y}$.

Different types of rollers are produced with different frequencies according to the demand of the

Notation	Definition
$\mathcal{Y}$	Set of the months of the year
$\mathcal{R}$	Set of roller types

Table 1: Definitions of the sets

Notation	Definition
f_i	the frequency of producing roller type i
	(i.e. the number of batches to be produced during the year)
d_i	the time distance (number of months)
	between the production occasions of roller type i

Table 2: Definitions of the parameters for indices $i\epsilon\mathcal{R}$

customers during the year. Frequencies of rollers are indicated by f_i; see Table 2. The appropriate months for production of each type of rollers is regulated with respect to its frequency during one year. The time distances between the production of a type of roller are almost equal in the production channel. Therefore, d_i, the time distance between the production occasions of type i, is considered to equal $\lceil 12/f_i \rceil$ i.e. d_i must be an integer.

4.1.2 Model

The following model determines which types of rollers should be produced in each month of the year such that the number of batches produced in each of the months are as equal as possible.

The binary variables x_{ij} determine weather roller type i, $i\epsilon\mathcal{R}$, should be produced in month j, $j\epsilon\mathcal{M}$, or not; see Table 3. The non-negative variables s_j^+ and s_j^- are slack variables to relax the equality constraint (3d); see Table 3.

$$\text{minimize} \quad \sum_{j\epsilon\mathcal{Y}} (s_j^+ + s_j^-), \tag{3a}$$

$$\text{subject to} \quad \sum_{j\epsilon\mathcal{Y}} x_{ij} = f_i, \qquad i\epsilon\mathcal{R}, \tag{3b}$$

$$\sum_{j=1+ad_i}^{(1+a)d_i} x_{ij} \geq 1, \qquad i\epsilon\mathcal{R} \quad a\epsilon 0, ..., 12/d_i - 1, \tag{3c}$$

$$\sum_{i\epsilon\mathcal{R}} x_{ij} + s_j^+ - s_j^- = \sum_{i\epsilon\mathcal{R}} f_i/12, \qquad j\epsilon\mathcal{Y}, \tag{3d}$$

$$s_j^+ \geq 0, \qquad j\epsilon\mathcal{Y}, \tag{3e}$$

$$s_j^- \geq 0, \qquad j\epsilon\mathcal{Y}, \tag{3f}$$

Notation	Definition
x_{ij}	$\begin{cases} 1 & \text{if type } i \text{ is produced in month } j \\ 0 & \text{otherwise} \end{cases}$
s_j^+	How much product can be produced more than $\sum_{i\epsilon\mathcal{R}} f_i/12$ in month j
s_j^-	How much product can be produced less than $\sum_{i\epsilon\mathcal{R}} f_i/12$ in month j

Table 3: Definitions of the variables for indices $i\epsilon\mathcal{R}, j\epsilon\mathcal{Y}$

$$x_{ij}\epsilon\{1,0\}, \qquad\qquad i\epsilon\mathcal{R} \quad j\epsilon\mathcal{Y}. \tag{3g}$$

The constraints (3b) ensure that rollers of type i is produced f_i times during one year and the constarints (3c) are formulated to keep a minimum distance between the productions for each type of roller. The distance between the frequencies of type i in one year, d_i, is obtained from the formula $d_i = \lceil 12/f_i \rceil$.

The slack variables in the constraints (3d) relax the equality requirement, and the average difference from equality is minimized by the formulation of the objective function (3a).

4.2 The scheduling problem

We next formulate an ILP model to construct an optimal schedule for the future production channel.

4.2.1 Definitions required for the model

All batches of rollers produced during one year must be included in the scheduling model. Different types of rollers are produced with different frequencies according to the demand of the customers. The set of all the batches is represented by $\mathcal{B}$, which is indicated in Table 4. All the machines of the channel are considered to be in set $\mathcal{M}$. The external heat and each of the new cells are in our model considered as machines. Each channel of fine grinding is also considered to be a machine in the set $\mathcal{M}$.

As mentioned before, time is considered to be discrete. The set of time steps is finite and indicated by $\mathcal{T}$. The number of time steps, T, should be large enough, otherwise the model will be infeasible and it will be impossible to schedule all of the batches. On the other hand, if the value of T is too large, the computation time can become too long. A heuristic has been formulated to develope a feasible solution of the model which leads to a suitable value for the parameter T. By decreasing the length of the time steps, T will be increased and the solution time increases rapidly.

Each machine of the channel, except for the new cells, can process only special types of rollers with respect to size and material. Therefore, when a batch of rollers enters the channel, it is assigned a specific route among a subset of the machines. The parameters λ_{bm} indicate which batch passes which machines in the channel. The new cells can process all types of rollers. Therefore, according to the definition of λ_{bm} in Table 5, $\lambda_{bm} = 1$ for all batches of rollers when m denotes one of the new cells.

Notation	Definition
$\mathcal{B}$	Set of batches of rollers
$\mathcal{M}$	Set of machines in the channel
$\mathcal{T}$	Set of time steps= $\{1, 2, ..., T\}$

Table 4: Definitions of the sets

Notation	Definition
λ_{bm}	$\begin{cases} 1 & \text{if batch } b \text{ is allowed to enter machine } m \\ 0 & \text{otherwise} \end{cases}$
A_{mn}	$\begin{cases} 1 & \text{if it is possible for a batch to enter machine } n \text{ directly after machine } m \\ 0 & \text{otherwise} \end{cases}$
t_{mn}	transportation time from machine m to machine n
P_{bm}	processing time of batch b in machine m
r_m	resetting time of machine m
F_m	the time that machine m is available

Table 5: Definitions of the parameters for indices $b\epsilon\mathcal{B}, m\epsilon\mathcal{M}, n\epsilon\mathcal{M}$

The notations for the different machines are indicated in Table 7. Our ILP model chooses one of the new cells for each batch.

The parameters A_{mn} specify the possible routes and directions for the batches between the machines. A batch cannot enter a machine after leaving another machine in the same process or in the next process. The order of the processes should be satisfied according to A_{mn}.

The parameters t_{mn} denotes the transportation time between machines m and n. The values of all parameters t_{mn} are considered to be zero in the data because all the machines are in the same building, except the external heat. For the external heat the transportatiom time and the resetting time, indicated by r_m in Table 5, of the external heat are considered to be zero. The processing time, indicatded by P_{bm} in Table 5, of each batch in the external heat includes the time from a batch is being sent to the external heat until it arrives again in the RK building. In other words, the heat treatment process in the external heat, the resetting time, and the transportation time are included in P_{bm} for m representing the external heat.

The parameters F_m indicate the time when machine m is available. Since the model should be run for all the months, the value of the parameters F_m should be changed for each month. In the first month it is assumed that all the machines are available in the first time step. For the next month, it should be checked when the machines are available. When machine m has finished the process of the last batch of the previous month, and is reset, it is available for the batches of a current month.

The variables used in the model determine the time steps in which the batches enter the machines. Table 6 indicates the definition of the variables.

Notation	Definition	
x_{bmu}	$\begin{cases}1 \\ 0\end{cases}$	if batch b enters machine m in the beginning of time step u otherwise

Table 6: Definitions of the variables for indices $u\epsilon\mathcal{T}, b\epsilon\mathcal{B}, m\epsilon\mathcal{M}$

Machine	OKN	NAT	OKN2	Ipsen	Siti	Ex Heat	cell 1	cell 2	cell3	cell4
Notation	p_1	p_2	p_3	h_1	h_2	h_3	c_1	c_2	c_3	c_4

Table 7: The notations for the different machines. The notations of the fine grinding channels are the same as those indicated in Figures 1 and 2. The set M is defined as $M = \{p_1, p_2, p_3, h_1, h_2, h_3, c_1, c_2, c_3, c_4, R_4, R_5, R_7, R_8, R_9, R_{11}, R_{12}\}$.

4.2.2 The scheduling model without a big batch

Given the parameters, sets, and variables defined in Section 4.2.1, the problem to schedule the future production channel is formulated as to

$$\text{minimize} \qquad \sum_{u\epsilon\mathcal{T}} \sum_{b\epsilon\mathcal{B}} \sum_{m\epsilon\mathcal{M}} (u + P_{bm}) x_{bmu}, \tag{4a}$$

$$\text{subject to} \qquad \sum_{u\epsilon\mathcal{T}} x_{bmu} = \lambda_{bm}, \qquad b\epsilon\mathcal{B}, \quad m\epsilon\mathcal{M}\backslash\{c_1, c_2, c_3, c_4\}, \tag{4b}$$

$$\sum_{j=1}^{4} \sum_{u\epsilon\mathcal{T}} x_{b,c_j,u} = 1, \qquad b\epsilon\mathcal{B}, \tag{4c}$$

$$x_{bmu} + A_{mn} \sum_{t=1}^{\lfloor u+P_{bm}+t_{mn} \rfloor} x_{bnt} \leq 1, \qquad b\epsilon\mathcal{B}, \quad m\epsilon\mathcal{M}, \quad n\epsilon\mathcal{M}, \quad u\epsilon\mathcal{T}, \tag{4d}$$

$$x_{bmu} + \sum_{t=u}^{\lfloor u+P_{bm}+r_m \rfloor} \sum_{\beta\epsilon\mathcal{B}\backslash\{b\}} x_{\beta mt} \leq 1, \qquad b\epsilon\mathcal{B}, \quad m\epsilon\mathcal{M}\backslash\{h_3\}, \quad u\epsilon\mathcal{T}, \tag{4e}$$

$$\sum_{t=1}^{\lfloor F_m \rfloor} \sum_{b\epsilon\mathcal{B}} x_{bmt} = 0, \qquad m\epsilon\mathcal{M}, \tag{4f}$$

$$x_{bmu}\epsilon\{1, 0\}, \qquad m\epsilon\mathcal{M}, \quad u\epsilon\mathcal{T}. \tag{4g}$$

If batch b enters machine m at time step u, which means that $x_{bmu} = 1$, Figure 3 indicates that the batch leaves machine m at time $u + P_{bm}$, which may not be the beginning of a time step. Therefore, the objective function of the model ((4a)) tries to finish the processes of all the machines on all batches as soon as possible. In other words, the model strives to keep the batches of rollers in the buffers for as short time as possible which leads to small buffer sizes. Therefore, the model is minimizing the sizes of the buffers indirectly as well.

It should be mentioned that minimizing the processes of the machines in fine grinding channel ,$R_4, R_5, R_7, R_8, R_9, R_{11}, R_{12}$, on the batches may not be usefull because the main goal is having tight schedule which leads to small buffers. The schedule of the batches on the machines is the result required not the optimal value, so according to the objective function The batches should be processed on all machines as soon as possible.

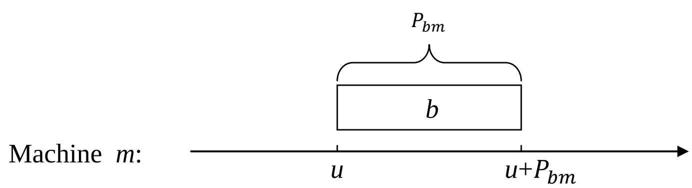


Figure 3: Batch b enters machine m at the time step u, and leaves it at time $u + P_{bm}$

The constraints (4b) ensure that each batch passes each machine, except the new cells, exactly once (if it is allowed to); if $\lambda_{bm} = 1$, batch b passes machine m exactly once; if $\lambda_{bm} = 0$, batch b never passes machine m. It should be noticed that all new cells of rough grinding can accept all kinds of batches, and $\lambda_{bm} = 1$ for all batches in these machines. The constraints (4c) make sure that each batch passes exactly one of the new cells.

The constraints (4d) ensure that each batch cannot be in more than one machine at a certain time. In other words, if batch b enters machine m at time step u ($x_{bmu} = 1$), it cannot enter the next machine, n, before time $u + P_{bm} + t_{mn}$ (not before leaving m, and being transported to the next machine, n), which means that $x_{bnt} = 0$ for $t \epsilon \{1, ..., \lfloor u + P_{bm} + t_{mn} \rfloor\}$ (since time steps are denoted by integer values, and the value of $u + P_{bm} + t_{mn}$ is not necessarily an integer, we need the integer part of $u + P_{bm} + t_{mn}$). It should be noticed that x_{bnt} might be equal to one, for a time step greater than $\lfloor u + P_{bm} + t_{mn} \rfloor$. In Figure 4, $x_{bnt} = 1$ for $t = \lfloor u + P_{bm} + t_{mn} \rfloor + 1$

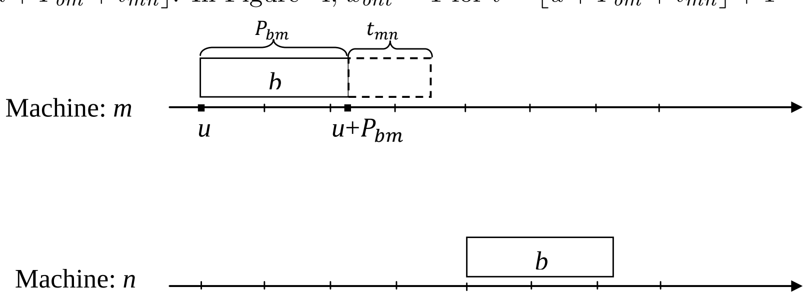


Figure 4: A batch can not go to the next machine before leaving the current machine and being transported to the next machine

The constraints (4e) are formulated to make sure that each machine does not process more than one batch at a certain time, except the external heat. Moreover, when one batch leaves a machine, the next batch cannot enter the machine before resetting the machine. In other words, when batch b enters machine m at time step u, which means $x_{bmu} = 1$, no other batch can enter machine m as long as batch b is in machine m or machine m is resetting. Therefore, ($x_{\beta mt} = 0, \beta \epsilon \mathcal{B} \backslash \{b\}$ for

$t \epsilon \{u, ..., \lfloor u + P_{bm} + r_m \rfloor\}$. It is feasible to have $x_{\beta mt} = 0$ for $t > \lfloor u + P_{bm} + r_m \rfloor$. In Figure 5, the next batch enters machine m at time step $\lfloor u + P_{bm} + r_m \rfloor + 1$.

According to the constraints (4f), no batch can enter machine m as long as this machine is occupied by the batches of the previous month. The model should be run for all months of the year, so the result of each month affects the schedule for the next month. These constraints are formulated to consider this property.

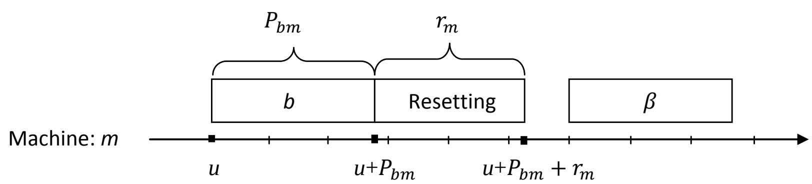


Figure 5: The next batch cannot enter the machine as long as the current batch is in the machine or the machine is resetting

4.2.3 The scheduling model including a big batch

As explained previously, certain big batches, containing more than 100,000 rollers, should be split into two batches for the heat treatment. First, half of the batch enters the heat treatment machine (Ipsen or Siti), then the machine is reset; after that the second half of the batch enters the machine.

In order to model this complication, we divide the big batch into two batches, and instead of one specific big batch b_i, we consider two batches b_i^a and b_i^b. The corresponding two batches should be kept together in all machines except for the heat treatment machines, this complication is modelled by the constraints:

$$x_{b_i^a mu} = x_{b_i^b mu}, \qquad u \epsilon \mathcal{T} \quad m \epsilon \mathcal{M} \backslash \{h_1, h_2\}. \tag{5}$$

When we add the constraints (5) to the model (4), it will become infeasible because of the constraints (4d) which ensure that each machine cannot have more than one batch at a certain time. If it holds that $x_{b_i^a mu} = 1$, according to the third constraint $x_{b_i^a,m,u} = 1$ is not possible, while the constraints (5) is forcing $x_{b_i^a mu}$ to equal one. We conclude that we need make some new changes in the model to make it feasible. Instead of the constraints (4d), we formulate the following constraints:

$$x_{bmu} + \sum_{t=u}^{\lfloor u+P_{bm}+r_m \rfloor} \sum_{\beta \epsilon \mathcal{B} \backslash \{b_i^a, b_i^b, b\}} x_{\beta mt} \leq 1, \qquad b \epsilon \mathcal{B} \backslash \{b_i^a, b_i^b\}, \quad m \epsilon \mathcal{M} \backslash \{h_3\}, \quad u \epsilon \mathcal{T}, \tag{6a}$$

$$x_{b_i^a mu} + \sum_{t=u}^{\lfloor u+P_{b_i^a m}+r_m \rfloor} \sum_{\beta \epsilon \mathcal{B} \backslash \{b_i^a, b_i^b\}} x_{\beta mt} \leq 1, \qquad m \epsilon M \backslash \{h_1, h_2, h_3\}, \quad u \epsilon \mathcal{T}, \tag{6b}$$

$$x_{b_i^a m u} + \sum_{t=u}^{\lfloor u+P_{b_i^a m}+r_m \rfloor} x_{b_i^b m t} \leq 1, \qquad u \epsilon \mathcal{T}, \quad m \epsilon \{h_1, h_2\}, \tag{6c}$$

$$x_{b_i^b m u} + \sum_{t=u}^{\lfloor u+P_{b_i^b m}+r_m \rfloor} x_{b_i^a m t} \leq 1, \qquad u \epsilon \mathcal{T}, \quad m \epsilon \{h_1, h_2\}. \tag{6d}$$

In the constraints (6a), the batches b_i^a and b_i^b are separated from the rest of the batches. The constraints (6b) are formulated for the batches b_i^a and b_i^b, when they are not in any of the heat treatment machines, and the constraints (6c) and (6d) concern the batches b_i^a and b_i^b when these are being processed in the heat treatment machines. By separating b_i^a and b_i^b from the rest of the batches in the constraints (6a), we overcome the infeasibility of the model which occured due to the addition of the constraints (5).

Since the indices b_i^a and b_i^b are not included in the set of batches in the constraints (6a), it is necessary to include certain constraints for b_i^a and b_i^b as well. The constraints (6b) have been formulated to ensure that as long as batch b_i^a is processed by a machine, which is not a heat treatment machine, no other batch can enter that machine except the batch b_i^b. Therefore, according to the constraints (5), the batches b_i^a and b_i^b enter the allowed machines, except for the heat treatment machines, at the same time.

The constraints (6c) and (6d) make sure that when one half of b_i, which is a big batch, is in one of the heat treatment machines, the other half of b_i cannot enter the heat treatment machine as long as the first half is still in the machine or the machine is resetting. It should be mentioned that none of the constraints (6c) or (6d) are redundant. By removing constraints (6d), the solution would not be acceptable because the model would try to finish the processing of all the batches as soon as possible. The batch b_i^b enters the heat treatment machine before the batch b_i^a, and the batch b_i^a enters the machine while it has not finished the processing of the batch b_i^b or is resetting.

4.3 Heuristic

When solving the scheduling model, the values of all parameters need to be known. The set of time steps $\mathcal{T} = \{1, 2, \ldots, T\}$ is finite, but the value of T that is required is not known. It is desirable that the value of T is small to ensure a short computation time, but if T is not large enough, then there may exist no feasible solution to the problem. Therefore, a heuristic has been developed to find a feasible schedule which leads to a suitable value for T; see Algorithm 1.

In the feasible schedule resulting from running the heuristic all the constraints of the scheduling model should be satisfied. When one batch leaves a machine, the machine should be reset before processing the next batch in that machine. Moreover, each batch enters the next allowed machine, after leaving the current one. Since the new cells are allowed to process all types of rollers, and each batch should pass exactly one of the new cells, the heuristic chooses one of the new cells for each batch by random before starting the scheduling. In other words, Algorithm 1 changes the values of the parameter λ_{bm} for $m \epsilon \{c_1, c_2, c_3, c_4\}$ before the algorithm is run. For the scheduling model, $\lambda_{bm} = 1$ for all batches if m represents one of the new cells, and the model chooses one of the new cells for each batch such that the result is optimal, but in the heuristic $\lambda_{bm} = 1$ for only one of the new cells for each batch.

4.3.1 A heuristic for the problem without considering a big batch

In Algorithm 1, a number of matrices and variables are defined for the batches and the machines. In order to find out when one batch can enter the next machine, it is needed to know which was the previous machine, when the batch left this previous machine, and when the batch is ready to enter the next machine. To find a feasible schedule for a machine, the required information is which was the previous batch in this machine, when the previous batch left this machine, and when the machine is available for the next batch (when the resetting of the machine is finished).

After the initialization, in which all matrices are assigned initial values, the algorithm loops over all the batches. The next step is another loop over all the machines. The last batch of each machine, which is the previous batch, should be found. The value of λ_{bm} should be checked to make sure whether the batch is allowed to pass the machine or not. Then the process of scheduling starts. If the batch already left the previous machine, and the machines are available, it enters the machine in the next time step. If the batch is ready to enter the machine, but the machine is occupied by another batch or it is being reset, the batch can enter the machine in the first time step that the machine is available after resetting. Then the values of the matrices indicating when the batch is ready to enter the next machine, and the machine is ready to accept the next batch should be updated for scheduling the rest of the batches and the rest of the machines. After the loop over the machines, when the schedule of each batch is completed, the algorithm saves the time that the batch is produced in the variable TP. After the loop over the batches, the next integer value of the maximum elemet of the vector of TP is the suitable value of T for the scheduling model.

4.3.2 A Heuristic for the problem considering a big batch

When there is a big batch, denoted by b_i, we know that it should be split into two batches for the heat treatment process. It has been assumed that in the set of batches, $\mathcal{B}$, instead of one big batch, b_i, there are two batches, b_i^a and b_i^b. These two batches are processed together in all the machines, except in the heat treatment machine. Therefore, the heuristic has a specific different algorithm for these batches. In Algorithm 2, the matrices, variables, and the initialization are defined equivalently as Algorithm 1. The scheduling for the other batches is also the same. In Algorithm 2, b_i^a and b_i^b enter the loop separately and respectively. The algorithm used for scheduling all the batches is performed for b_i^a as well. If $b = b_i^b$, the set of the machines, $\mathcal{M}$, is divided in to three sets of machines: before the heat treatment, the heat treatment, and after the heat treatment machines. If the machine is before the heat treatment, the batch b_i^b starts and finishes in the same machines and exactly when b_i^a starts and finishes, respectively. If m is a heat treatment machine, the batch b_i^b possesses the same scheduling algorithm as the other batches. The previous batch is b_i^a because both the batch b_i^a and the batch b_i^b should pass the heat treatment machine. If m is a machine after the heat treatment in the flow, b_i^b follows the same algorithm as all the batches to be scheduled. When all the batches are sheduled in all machines, the schedule of the batch b_i^a should be changed in the machines after the heat treatment, because the batches b_i^a and b_i^b should be processed together in the machines before and after the heat treatment. So far, they have been together in the machines before the heat treatment. Therefore, another loop over the machines after the heat treatment fixes the batch b_i^a to the batch b_i^b in the machines.

4.4 Improving the schedule by squeezing

The value of T chosen for the scheduling model is the number of the time steps at which the last batch is finished in its last machine. By solving the model, we get an optimal schedule. Since the model is

Algorithm 1, Heuristic without a big batch

$\text{URPM}_{bm} \leftarrow 1, \text{Ustart}_{bm} \leftarrow 0, \text{Ufinish}_{bm} \leftarrow 0, \text{URNM}_{bm} \leftarrow 1 * fr(m)$
  for $b\epsilon\mathcal{B}$ **do**
    for $m\epsilon\mathcal{M}$ **do**
      find the previous batch
      if $\lambda_{bm} = 1$ **then**
        $\text{URPM}_{bm} \leftarrow \text{Ufinish}_{\text{PrevM},b} + t_{\text{PrevM},m}$
        $\text{Ustart}_{bm} \leftarrow max(\text{URNB}_{\text{PrevB},m}, \text{URPM}_{bm})$
        $\text{Ustart}_{bm} \leftarrow ceil(\text{Ustart}_{bm})$
        $\text{Ufinish}_{bm} \leftarrow \text{Ustart}_{bm} + P_{bm}$
        $\text{URNB}_{bm} \leftarrow \text{Ufinish}_{bm} + r_m$
        $\text{PrevM} \leftarrow m$
      end if
    end for
      $\text{TP}_b = max(\text{URNB})$
  end for
    $T = \lceil max\text{TP}_b \rceil$

time-indexed, the solution found is not necessarily optimal in the "exact" model (which we actually haven't formulated). Moreover, since we model an approximation of the real problem, a solution to the "exact" model may not be the most appropriate anyway.

When one batch is ready to enter the next machine of its route in the channel, there are three possibilities; the next machine is available, the next machine is occupied by another batch, or the next machine is being reset. If a batch is ready to enter an available machine at time $t + \epsilon$, where $0 < \epsilon < 1$ and $t\epsilon\mathcal{T}$, it enters the machine at time step $t + 1$ (to have feasible schedule according to the time indexed model). Then the machine is empty in the time interval $[t+\epsilon, t+1]$. If the next machine is not available, the batch cannot enter the machine (is then in the buffer). If the next machine is available at time $u+\delta$, where $u+\delta > t+\epsilon$, $0 < \delta < 1$, and $u\epsilon\mathcal{T}$, the batch enters the next machine at time step $u + 1$. In the feasible solution the machine is empty during the time interval $[u + \delta, u + 1]$. These cases are not realistic since in prcatice the machines never stop working.

Therefore, in the next step, we "squeeze" the optimal solution of the model as much as possible. After squeezing, one batch enters the next machine in its route as soon as the next machine is available. For instance, when a batch is ready to enter the machine at time $t + \epsilon$, where $0 < \epsilon < 1$ and $t\epsilon\mathcal{T}$, if the machine is available, the batch enters the machine at $t + \epsilon$. If the machine is occupied or being reset, the batch enters the machine as soon as the machine is available.

The squeezing algorithm (Algorithm 3) takes as input the optimal schedule obtained from the solution of the scheduling model. In the loops over the batches and the machines, the time at which each batch enters each machine is calculated. In a real schedule, each batch can enter a machine after leaving the previous machine, and a machine can process a new batch if it is available. Therefore, the maximum of these times is considered to be the time when a batch can enter the machine.

5 An illustrating example

In this section, a little flow which is similar to the production channel at SKF is develped. In this example, ten machines (denoted $m_i, i = 1, ..., 10$) are available, and three batches (denoted $b_i, i = 1, ..., 3$) are to be processed. Five processes are to be performed on each type of batch. All

Algorithm 2, Heuristic algorithm with a big batch

$\text{URPM}_{bm} \leftarrow 1, \text{Ustart}_{bm} \leftarrow 0, \text{Ufinish}_{bm} \leftarrow 0, \text{URNM}_{bm} \leftarrow 1 * fr(m)$
  for $b \epsilon \mathcal{B}$ **do**
    if $b \neq b_i^b$ **then**
    find the previous machine
      for $m \epsilon \mathcal{M}$ **do**
          if $\lambda_{bm} = 1$ **then**
            $\text{URPM}_{bm} \leftarrow \text{Ufinish}_{\text{PrevM},b} + t_{\text{PrevM},m}$
            $\text{Ustart}_{bm} \leftarrow max(\text{URNB}_{PrevB,m}, \text{URPM}_{bm})$
            $\text{Ustart}_{bm} \leftarrow ceil(\text{Ustart}_{bm})$
            $\text{Ufinish}_{bm} \leftarrow \text{Ustart}_{bm} + P_{bm}$
            $\text{URNB}_{bm} \leftarrow \text{Ufinish}_{bm} + r_m$
            $\text{PrevM} \leftarrow m$
          end if
      end for
    else if $b = b_i^b$ **then**
    find the previous machine
      for $m \epsilon \mathcal{M}$ **do**
        if *m is before the heat treatment* **then**
          $\text{Ustart}_{bm} = \text{Ustart}_{b_i^a,m}$
          $\text{Ufinish}_{bm} = \text{Ufinish}_{b_i^a,m}$
        else if *m is a heat treatment machines* **then**
          if $\lambda_{bm} = 1$ **then**
            $\text{PrevB} \leftarrow b_i^a$
            $\text{URPM}_{bm} \leftarrow \text{Ufinish}_{\text{PrevM},b} + t_{\text{PrevM},m}$
            $\text{Ustart}_{bm} \leftarrow max(\text{URNB}_{\text{PrevB},m}, \text{URPM}_{bm})$
            $\text{Ustart}_{bm} \leftarrow ceil(\text{Ustart}_{bm})$
            $\text{Ufinish}_{bm} \leftarrow \text{Ustart}_{bm} + P_{bm}$
            $\text{URNB}_{bm} \leftarrow \text{Ufinish}_{bm} + r_m$
            $\text{PrevM} \leftarrow m$
          end if
        else if *m is after the heat treatment* **then**
          if $\lambda_{bm} = 1$ **then**
            $\text{URPM}_{bm} \leftarrow \text{Ufinish}_{\text{PrevM},b} + t_{\text{PrevM},m}$
            $\text{Ustart}_{bm} \leftarrow max(\text{URNB}_{\text{PrevB},m}, \text{URPM}_{bm})$
            $\text{Ustart}_{bm} \leftarrow ceil(\text{Ustart}_{bm})$
            $\text{Ufinish}_{bm} \leftarrow \text{Ustart}_{bm} + P_{bm}$
            $\text{URNB}_{bm} \leftarrow \text{Ufinish}_{bm} + r_m$
            $\text{PrevM} \leftarrow m$
          end if
        end if
      end for
        $\text{TP}_b = max(\text{URNB})$
  end for
      for *m after the heat treatment* **do**
        $\text{Ustart}_{b_i^a,m} = \text{Ustart}_{b_i^b,m}$
        $\text{Ufinish}_{b_i^a,m} = \text{Ufinish}_{b_i^b,m}$
      end for
    $T = \lceil max \text{TP}_b \rceil$

Algorithm 3, The squeezing algorithm

```
for b∈B do
    for m∈M do
      if m is not external heat
        Ustart_bm = max{Ufinish_b,PrevM, URNB_PrevB,m}
      else
        Ustart_bm = Ufinish_b,PrevM
      end if
    end for
end for
```

processes are to be performed on all batches exactly once. Figure 6 shows the set of machines.

The plan is to build eight buffers, one buffer before each of the machines m_3, m_4,..., m_{10}. The required sizes of these buffers are the entities seeked. These buffers should be as small as possible while they should have enough space to store the unfinished batches. Therefore, we are trying to minimize the leadtime in order to keep the batches in the buffers as short time as possible which means minimizing the sizes of the buffers indirectly.

Each pair of machines in Figure 6 perform the same process on different types of batches, so each batch should pass a specific machine of each process. In this example m_7 and m_8 have the role of new cells at the SKF channel which means that they do the same process on all kinds of batches. Each of these machines can process one batch at a certain point in time, but when a big batch is ready to enter the machine m_3 or m_4, it should be divided into two batches. In fact, m_3 and m_4 have the role of heat treatment machines in this example.

Figure 7 illustrates the routes of the three batches through the set of machines. Each batch is assigned a specified machine for each process except the fourth process. When one batch is finished by the third process, both m_7 and m_8 can perform the fourth process. Therefore, the mathematical model chooses one of m_7 and m_8 for each batch.

Each machine can process only one batch at a certain time, after which it must be reset. Table 12 indicates the resetting times of all the machines. After the resetting, the next batch may enter the machine. In this example, b_1 is a big batch. When this batch is ready to enter m_4, it should be split into two batches since the capacity of m_4 is limited, and it cannot process the whole batch b_1 at the same time. Therefore, when b_1 enters m_4, it is split into two batches. The first half of b_1 enters m_4, and when the processing is finished, m_4 is reset and the second half of b_1 enters m_4. To handle this exception, we split b_1 into b_1^a and b_1^b. We assume that instead of one batch (b_1), we have two batches (b_1^a and b_1^b). These two batches should be together in all the machines except m_4. The capacity of m_3 is also limited, and if a big batch should enter m_3 it should be split into two batches. In this example, b_2, which is not a big batch, is passing m_3, so it is not needed to divide it. When batch b_3 enters m_4, it is neither necessary to divide it since b_3 is not a big batch.

The information in Figure 7 about the routes of the batches in the machines is also indicated in Table 8, which shows the values of λ_{bm} for all the batches in all the machines. If $\lambda_{bm} = 1$, batch b may pass machine m during its route in the channel. Since m_7 and m_8 can perform their processes on all batches, $\lambda_{bm} = 1$ for all batches in these machines. Table 9 indicates the processing times of each machine on each batch. If $\lambda_{bm} = 0$ for a batch in a machine, then $P_{bm} = 0$ too. In Tables 8 and 9, there are four batches in which b_1^a and b_1^b have equivalent values of λ_{bm} and P_{bm} respectively. In practice, the batches b_1^a and b_1^b are together in all the machines, except m_3 and m_4

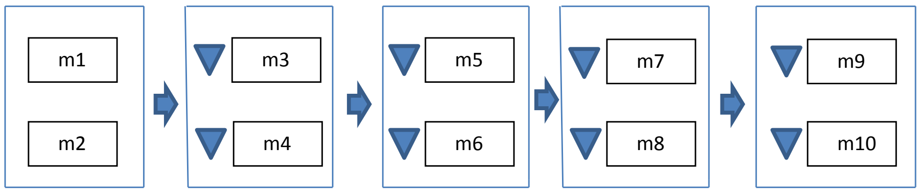


Figure 6: Ten machines are available, each of which processes special types of batches. Each pair of machines perform the corresponding process.

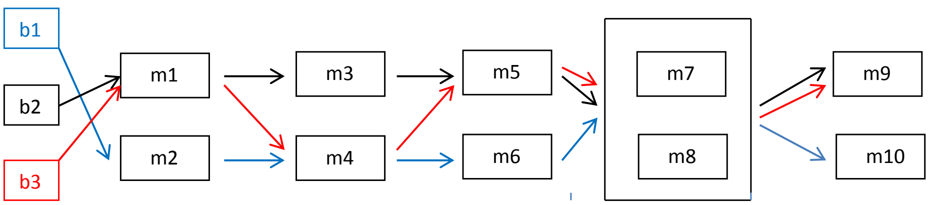


Figure 7: The routes of the three batches through the ten machines.

5.1 Instruction

As explained in Section 4, we perform a four step procedure to calculate the volumes needed for the buffers.

1. The heuristic produces a feasible solution in order to find a suitable value of T to use in the model. The solution developed by the heuristic is not necessarily a good solution because it assigns the batches on appropriate machines, only such that the resulting schedule is feasible. The heuristic chooses one of the machines m_7 and m_8 by random to produce a feasible solution.
2. The ILP problem (7) is solved to find an optimal schedule. The model minimizes the lead time. The solution of the model is the optimal schedule according to the specified time step intervals. The schedule is, however, an approximation of the case that time is continuous.
3. The optimal schedule is modified by squeezing the batches in the machines as much as possible.
4. The schedules for the buffers are obtained from the squeezed optimal schedule of the channel. Having the schedules of the buffers and the number of boxes in each batch, as indicated in Table 11, leads to the graphs of the buffers showing the number of boxes in the buffer at each point in time.

The ILP model for scheduling the channel of this example is to

$$\text{minimize} \quad \sum_{u \epsilon \mathcal{T}} \sum_{b \epsilon \mathcal{B}} \sum_{m \epsilon \mathcal{M}} (u + P_{bm}) x_{bmu}, \tag{7a}$$

$$\text{subject to} \quad \sum_{u\epsilon\mathcal{T}} x_{bmu} = \lambda_{bm}, \qquad b\epsilon\mathcal{B}, \quad m\epsilon\mathcal{M}\backslash\{m_7, m_8\}, \tag{7b}$$

$$\sum_{u\epsilon\mathcal{T}}(x_{b,m_7,u} + x_{b,m_8,u}) = 1, \qquad b\epsilon\mathcal{B}, \tag{7c}$$

$$x_{bmu}+A_{mn}\sum_{t=1}^{\lfloor u+P_{bm}+t_{mn}\rfloor} x_{bnt} \leq 1, \qquad b\epsilon\mathcal{B}, \quad m\epsilon\mathcal{M}, \quad n\epsilon\mathcal{M}, \quad u\epsilon\mathcal{T}, \tag{7d}$$

$$x_{b_1^a,m,u} = x_{b_1^b,m,u}, \qquad u\epsilon\mathcal{T}, \quad m\epsilon\mathcal{M}\backslash\{m_3, m_4\}, \tag{7e}$$

$$x_{bmu} + \sum_{t=u}^{\lfloor u+P_{bm}+t_{mn}\rfloor} \sum_{\beta\epsilon\mathcal{B}\backslash\{b_1^a,b_1^b,b\}} x_{\beta mt} \leq 1, \qquad b\epsilon\mathcal{B}\backslash\{b_1^a, b_1^b\}, \quad m\epsilon\mathcal{M}, \quad u\epsilon\mathcal{T}, \tag{7f}$$

$$x_{b_1^a mu} + \sum_{t=u}^{\lfloor u+P_{bm}+t_{mn}\rfloor} \sum_{\beta\epsilon\mathcal{B}\backslash\{b_1^a,b_1^b\}} x_{\beta mt} \leq 1, \qquad m\epsilon M, \quad u\epsilon\mathcal{T}, \tag{7g}$$

$$x_{b_1^a,m,u} + \sum_{t=u}^{\lfloor u+P_{bm}+t_{mn}\rfloor} x_{b_1^b,m,t} \leq 1, \qquad u\epsilon\mathcal{T}, \quad m\epsilon\{m_3, m_4\}, \tag{7h}$$

$$x_{b_1^b,m,u} + \sum_{t=u}^{\lfloor u+P_{bm}+t_{mn}\rfloor} x_{b_1^a,m,t} \leq 1, \qquad u\epsilon\mathcal{T}, \quad m\epsilon\{m_3, m_4\}, \tag{7i}$$

$$x_{bmu}\epsilon\{1,0\}, \qquad m\epsilon\mathcal{M}, \quad u\epsilon\mathcal{T}. \tag{7j}$$

The objective function (7a) is minimizing the leadtime and the constraints (7b) and (7c) make sure that each batch passes the appropriate machines including one of the machines m_7 and m_8. The constraints (7d) use the values in Table 10 to find the possible routes and directions between the machines. The values of the parameters A_{mn} are indicated in Table 10. (It is not possible to go backwards in the channel, neither is it possible to go from one machine to another machine in the same process. In other words, batches are moving in the forward direction. Moreover, each batch should pass each of the processes. It is not possible to jump from the first process to the third process without passing the seond.) Because of the constraints (7d), each batch can enter the next machine after leaving the current machine and being transported to the next machine. The rest of the constraints are formulated because of the existance of b_1, a big batch. According to these constraints, the next batch cannot enter a machine as long as the machine is occupied by another batch or is being reset. The batches b_1^a and b_1^b should be together in all the machines, but in m_3 or m_4 they should enter the machine individually.

	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9	m_{10}
b_{1a}	0	1	0	1	0	1	1	1	0	1
b_{1b}	0	1	0	1	0	1	1	1	0	1
b_2	1	0	1	0	1	0	1	1	1	0
b_3	1	0	0	1	1	0	1	1	1	0

Table 8: The values of the parameters λ_{bm} indicating which batch should pass which machine.

	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9	m_{10}
b_{1a}	0	2.1	0	2	0	1.8	0.8	0.8	0	1.8
b_{1b}	0	2.1	0	2	0	1.8	0.8	0.8	0	1.8
b_2	1	0	1.3	0	1.4	0	1.4	1.4	1.3	0
b_3	0.7	0	0	1.5	1	0	0.7	0.7	1	0

Table 9: The values of the parameters P_{bm}, i.e., the processing times of the machines on the respective batches).

5.2 Channel schedule and discussion

We solve this example for four distinct values of the time step intervals to be able to compare the results. By changing the time step intervals, the values of the parameters P_{bm} and r_m will be changed. In general, if the length of each time step is divided by l, the processing time (as measured in number of time steps) will be multiplied by l. If the processing time of a batch in a machine is P_{bm} hours, and each time step is one hour, the processing time is P_{bm} time steps as well. It is also possible to consider each half hour as one time step. In that case, the processing time is $2P_{bm}$. In order to be more exact, we consider the cases of 15 and 5 minutes, length of the time steps as well (i.e., processing time of $4P_{bm}$ and $12P_{bm}$ time steps, respectively).

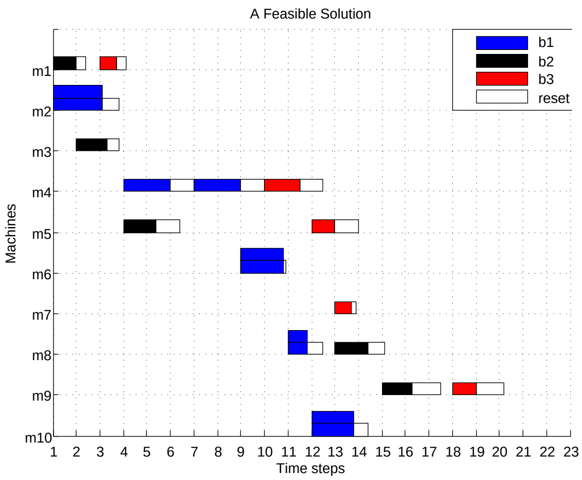
Figure 8 shows the resulting schedules of the heuristic for different lengths of the time steps. Since b_1 is a big batch, it should be split into two batches in the machine m_4. Both halves of b_1 are processed together in all machines except in the machine m_4. The heuristic is developed to find a suitable value for the number of time steps. From Table 13, it can be seen that by decreasing the length of the time steps, the value of T increases, so the number of variables and constraints also increases.

The values of the objective function of the scheduling model are indicated in Table 14 . It is clear that when the time steps are shorter, the production of the batches is finished faster which means that the leadtimes of the batches are shorter.

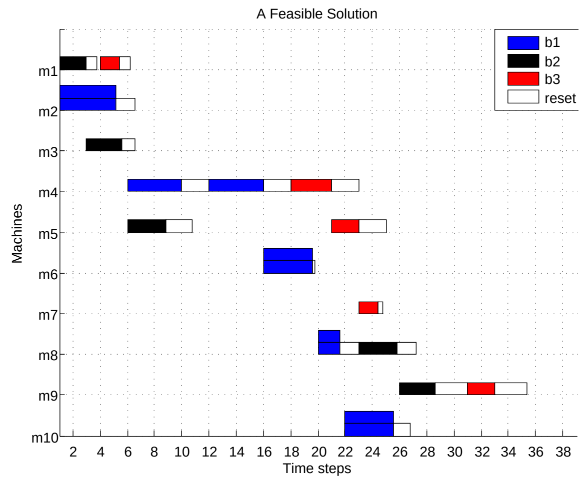
By giving the suitable number of time steps (T) obtained from the heuristic, indicated in Table 13, to the model (7), and solving it, the optimal schedules shown in Figure 9 are obtained. The schedules show that the order of the batches in the optimal solutions in Figure 9 is different from the feasible solution computed by the heuristic, presented in Figure 8, and the model chooses between the machines m_7 and m_8 for each batch such that the schedule is optimized.

Tables 14 shows that for a fixed length of the time steps, the value of the objective function in the optimal schedule is lower than that of the feasible schedule computed by the heuristic. It also shows that by decreasing the length of the time steps, the optimal objective value decreases as well.

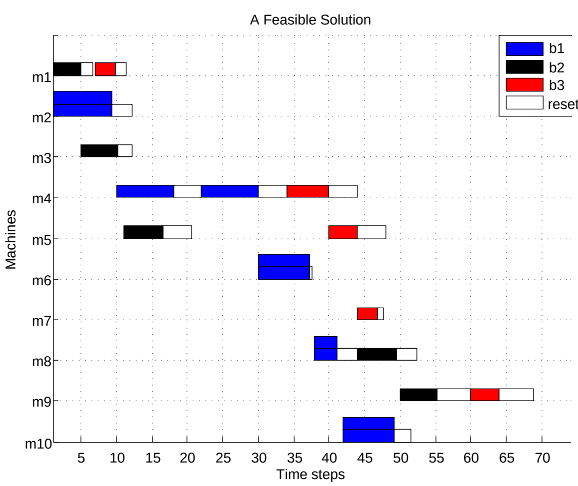
The schedules found by the model (7) are optimal with respect to the values chosen for the time step intervals, so the schedules computed are approximations of the optimal schedule when time is considered to be continuous. By using the squeezing algorithm we modify and enhance the schedules computed by the model. Figure 10 shows the squeezed optimal schedules. From Table 14, it is



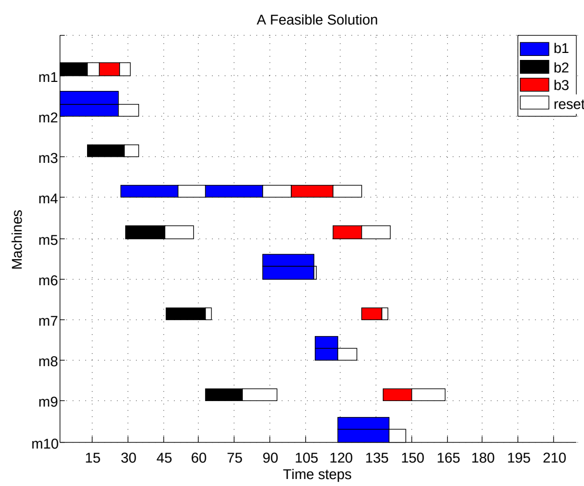
(a) Discretization interval: one hour.



(b) Discretization interval: 30 minutes.

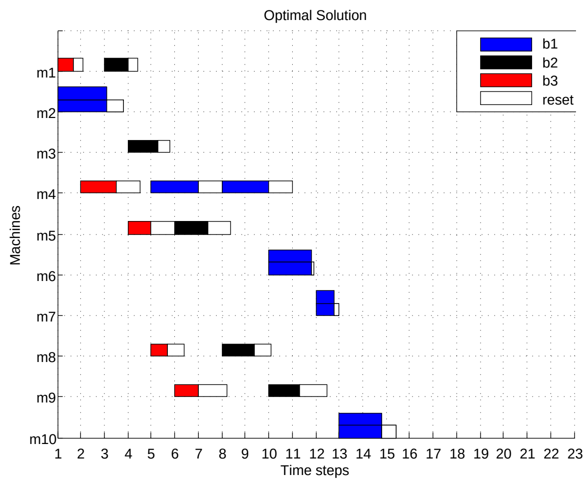


(c) Discretization interval: 15 minutes.

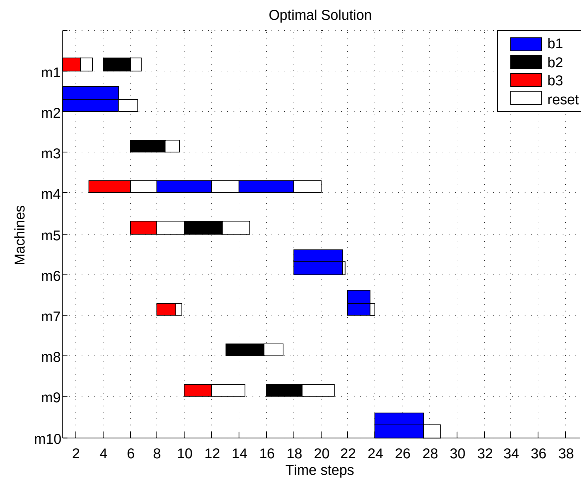


(d) Discretization interval: 5 minutes.

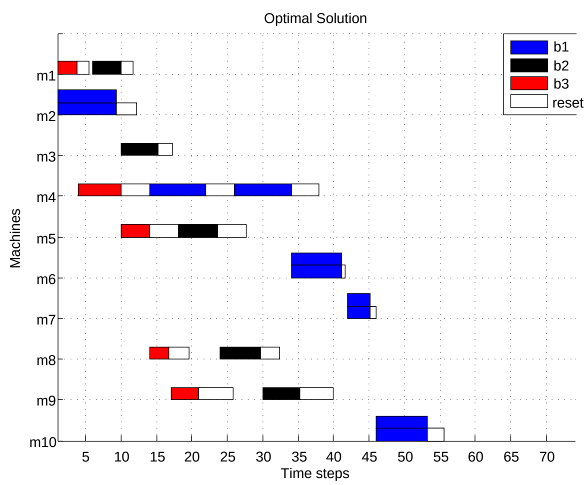
Figure 8: Schedules resulting from the heuristic for different lengths of the time steps. The big batch is denoted by two rectangles.



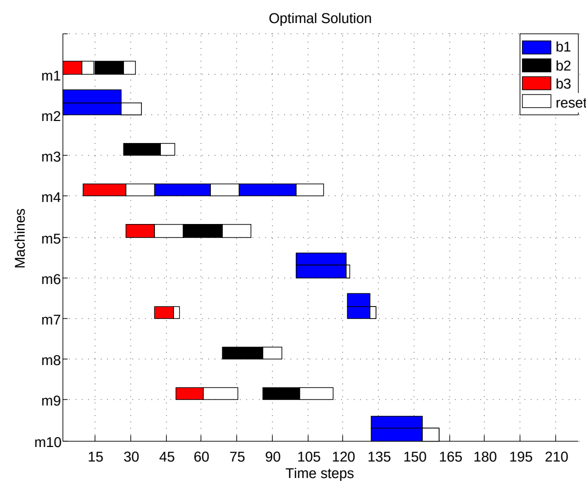
(a) Discretization interval: one hour.



(b) Discretization interval: 30 minutes.



(c) Discretization interval: 15 minutes.



(d) Discretization interval: 5 minutes.

Figure 9: Schedules resulting from solving the model (7) for different lengths of the time steps.

	m_3	m_4	m_5	m_6	m_7	m_8	m_9	m_{10}
m_1	1	1	0	0	0	0	0	0
m_2	1	1	0	0	0	0	0	0
m_3	0	0	1	1	0	0	0	0
m_4	0	0	1	1	0	0	0	0
m_5	0	0	0	0	1	1	0	0
m_6	0	0	0	0	1	1	0	0
m_7	0	0	0	0	0	0	1	1
m_8	0	0	0	0	0	0	1	1

Table 10: The parameters A_{mn} that denote the possible routes between the machines.

	b_1	b_2	b_3
# Boxes	10	2	3

Table 11: The number of boxes that each batch occupies.

clear that for a fixed length of the time steps, the value of the objective function is decreasing after squeezing the schedule, and indicates that by decreasing the length of the time steps, the value of the objective function of the squeezed schedule is decreasing.

Both Table 14 and Figure 11 show that when having shortertime step intervals, and being closer to continuous time, the difference between the objective values of the heuristic solution and the optimal solution to the model (7) decreases. In Figure 11, it can also be seen that in the heuristic step, by decreasing the length of the time steps, the value of the objective function decreases fast, while in the squeezing step, by making time steps shorter, the downward trend of the objective function is slow. This means that, although the length of time steps is important for the quality of the results, by squeezing the optimal schedule of the model (7), we obtain a good final result also for long time steps.

Both Table 14 and Figure 11 show that when the length of the time steps is larger, in each step of the instruction; see Section , (i.e., the heuristic, the solution to the model (7) and the squeezing) the difference in the obtained leadtime increases which means that in each step the improvement is increased. In other words the scheduling decreases the leadtime of the schedule of model (7) more rapidly compared to the leadtime decrease of the model (7) after the heuristic.

It can also be seen that in the heuristic, by decreasing the duration of the time steps, the leadtime is decreasing fast, while in the squizing by making each time step shorter, the downward trend of the lead time is slow. This means that, although the length of time step is important, by squeezing the optimal solution of the model, we obtain a good final result even for longer time steps.

Figure 12 indicates that by increasing the length of the time steps, The solution time required to solve the model (7) decreases rapidly. From Figures 11 and 12, it can be concluded that by having longer time steps and squeezig the schedule of the model (7), a good schedule can be obtained with acceptable computation time.

5.3 Buffers

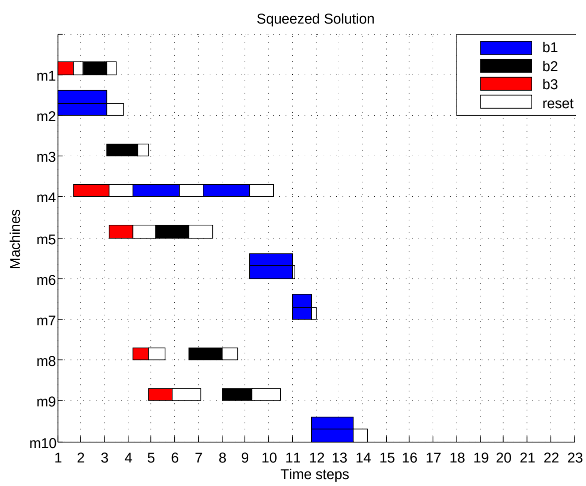
When one batch is ready to go to the next machine, and the next machine is not ready (either it is busy by another batch or it is being reset), the batch goes to the buffer, which is placed before

	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9	m_{10}
Resetting time	0.4	0.7	0.5	1	1	0.1	0.2	0.7	1.2	0.6

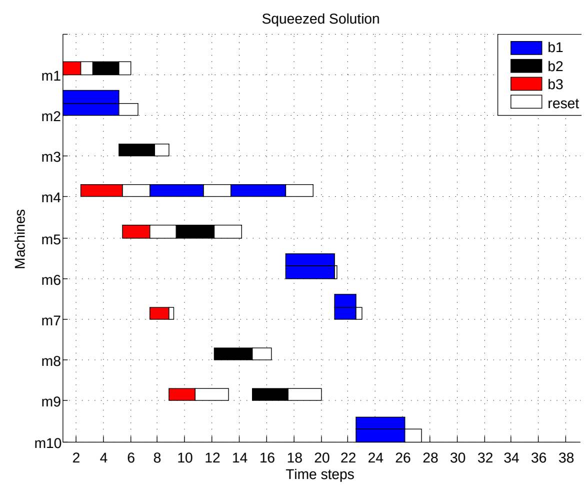
Table 12: The resetting times for the machines (hours).

length of time steps (minutes)	60	30	15	5
T	21	36	69	165

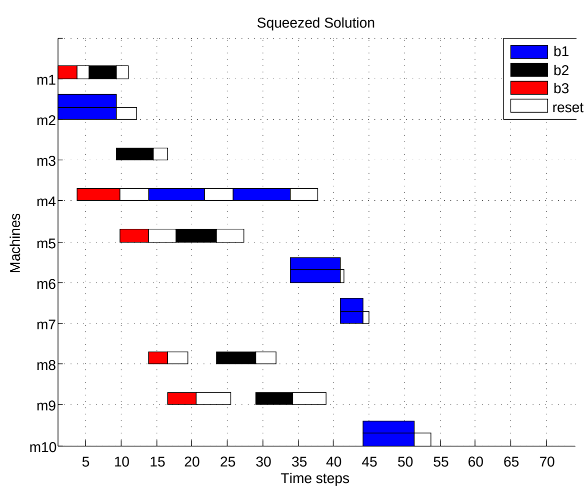
Table 13: The number of time steps in relation to the length of the time steps.



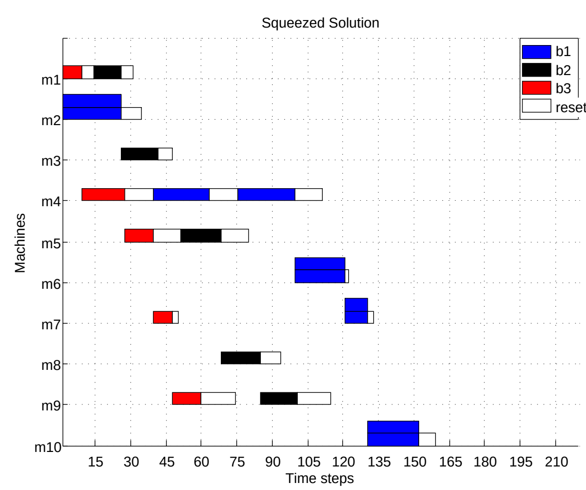
(a) Discretization interval: one hour.



(b) Discretization interval: 30 minutes.



(c) Discretization interval: 15 minutes.



(d) Discretization interval: 5 minutes.

Figure 10: Squeezed schedules from solving the model (7) for different lengths of the time steps.

Length of the time steps	60 min	30 min	15 min	5 min
Heuristic	196.3	174.3	165.8	142.9
Integer Linear Programming	162.3	142.8	138.3	121
Squeezing	145.7	137.2	131.5	115.2

Table 14: The value of the objective function obtained from each method in hours. The value of the objective function decreases in each step in each column. The values of the objective function decreases also by decreasing the length of the time steps, which can be seen in each column.

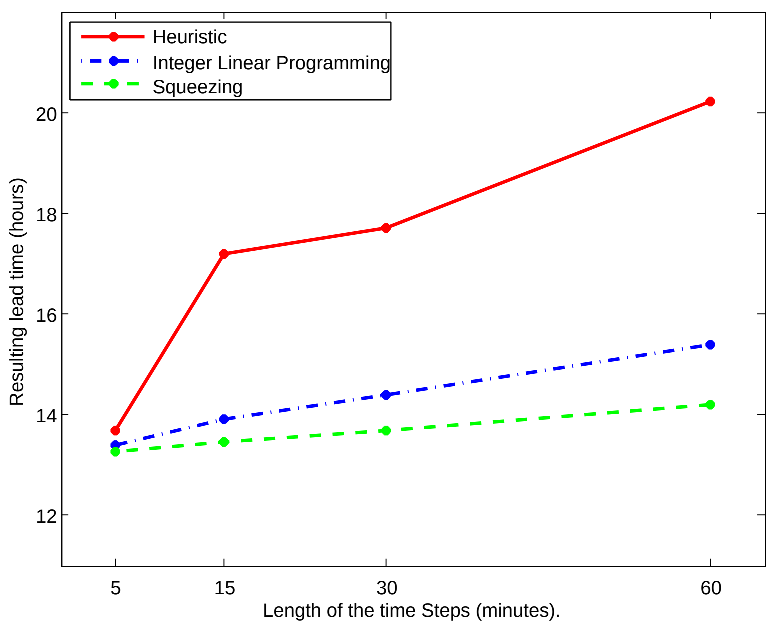


Figure 11: Lead times resulting from the heuristic, the solution of the mathematical model 7, and the squeezing, for different lengths of the time steps.

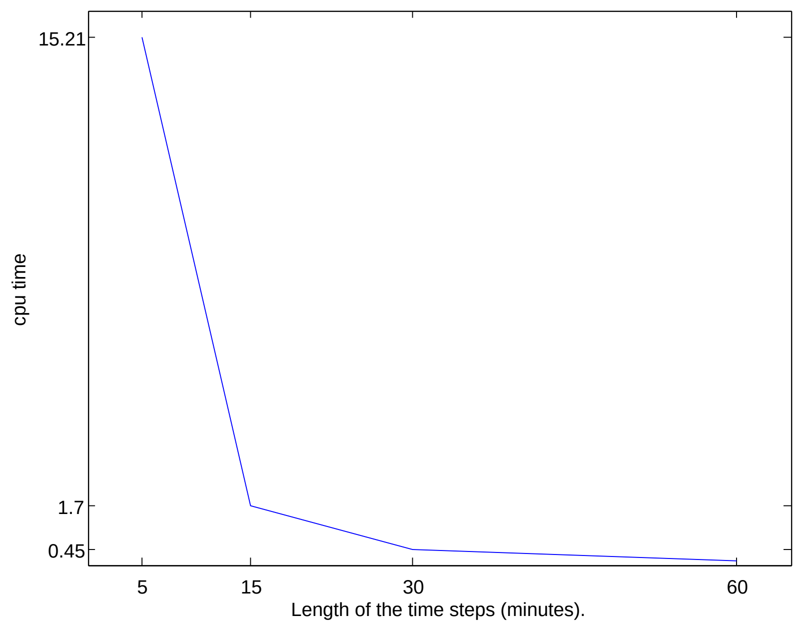


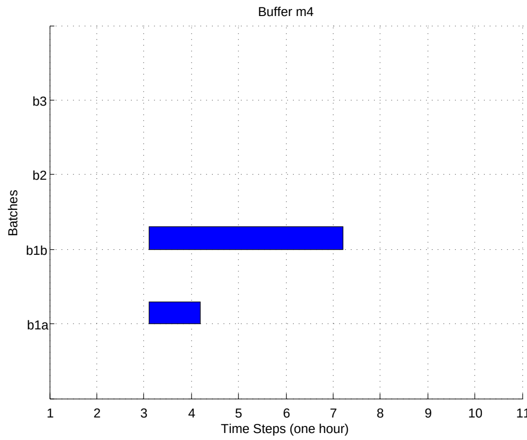
Figure 12: The cpu time required for solving the mixed integer linear programming problem for different length of the time steps.

the next machine. As soon as the next machine is ready (after finishing its job on another batch and resetting), the batch leaves the buffer, and enters the next machine. In other words, if a batch is ready to go to the next machine at time t_1, and the next machine is ready to accept it at time $t_2 > t_1$, the batch should stay in the buffer during this time (during the time interval (t_1,t_2)). If both the batch and the next machine are ready, the batch goes directly to the next machine, and not to the buffer. Figure 13 shows the schedules for the buffers placed before m_4, m_5, and m_6 (when each time step is assumed to be one hour). The other buffers (located before m_1, m_2, and m_3) will be empty during the whole planning period.

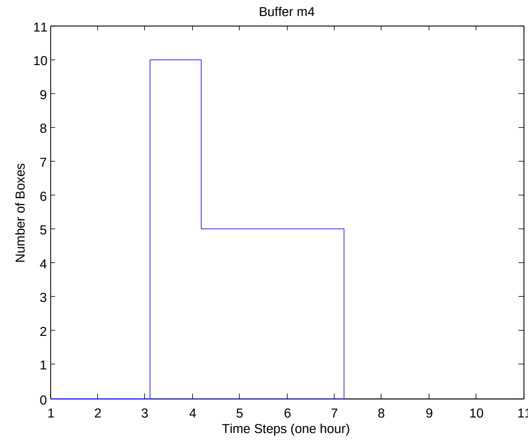
Figure 13 shows that the buffer m_4 contains both halfs of the batch b_1 but with different durations. When the batch b_1 should enter the machine m_4, this machine is occupied by the batch b_3; therefore, the batch b_1 goes to the buffer. When the machine m_4 has finished its job on the batch b_3 and is being reset, it is ready to accept half of the batch b_1, since it does not have the capacity to accept the whole batch b_1. Therefore, the first half of b_1 leaves the buffer and goes to m_4, while half of the batch b_1 stays in the buffer. When the first half of the batch b_1 leaves the machine m_4, this machine is reset, then it is ready for the second half of the batch b_1, which then also leaves the buffer.

When the first half of the batch b_1 leaves the machine m_4, it should enter the machine m_6. Figure 13 shows that the machine m_6 is ready to accept it immediately, but both parts of the batch b_1 should be together in all the machines except m_4. Therefore, the first half of b_1 should stay in the buffer until the second half of b_1 enters the buffer. As soon as the second half of b_1 leaves m_4, the first half of it leaves the buffers, and they enter the machine m_6 as one batch.

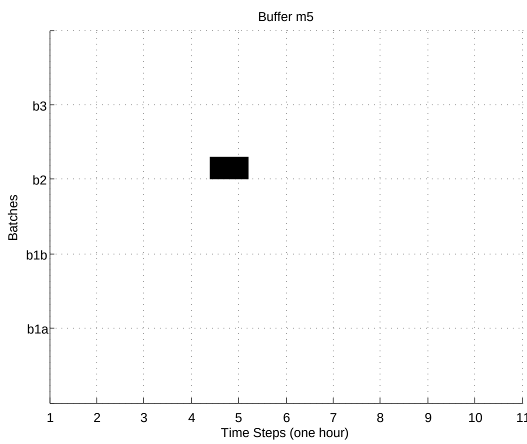
Figure 13 shows that, when the batch b_2 leaves the machine m_3 and is ready to go to the machine m_5 (the next machine), m_5 is being reset, and it cannot accept any batch. Therefore, the batch b_2 must stay in the buffer until the machine m_5 is reset and ready for processing.



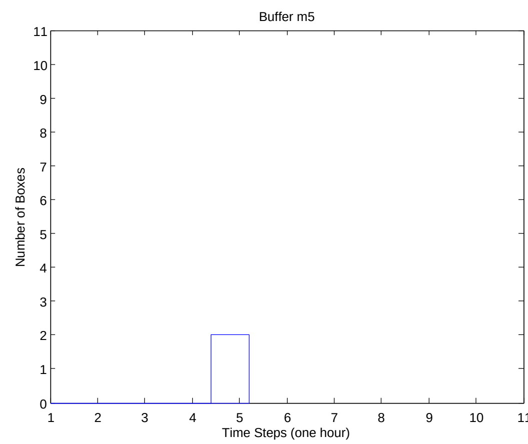
(a) Schedule



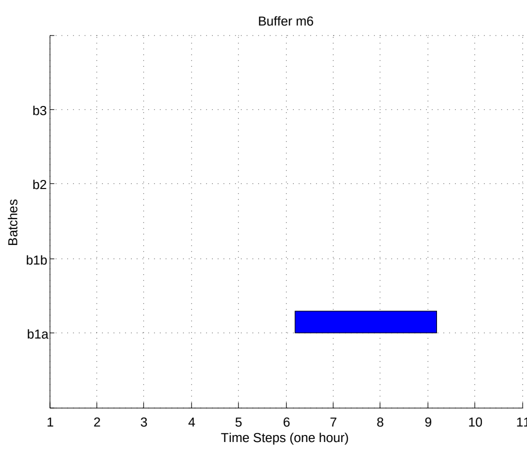
(b) Graph



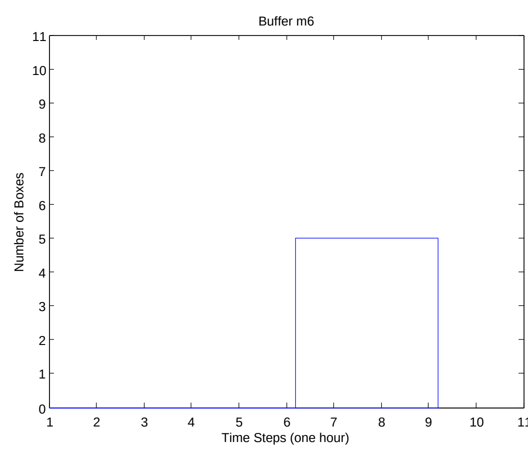
(c) Schedule



(d) Graph



(e) Schedule



(f) Graph

Figure 13: The schedules of the buffers before the machines m_4, m_5 and m_6 specifying the batches in the buffers and the graphs of the buffers before the machine m_4, m_5 and m_6 representing the number of boxes in the buffers.

Figure 13 indicates that the batch b_1 (composed by b_1^a and b_1^b, each containing five boxes) enters the buffer before the machine m_4. Then the batch b_1^a (the first half of the batch b_1 containing five boxes) leaves the buffer, and five boxes remain in the buffer. After almost three time steps the batch b_1^b (the second half of b_1) leaves the buffer which then becomes empty. Since the maximum number of batches in the buffer during this time is ten, the buffer should be big enough for ten boxes.

Batch b_2 enters the buffer located before m_5. Since the batch b_2 occupies two boxes, as long as it is in the buffer, there are two boxes in the buffer. Therefore, the buffer should have enough space for 2 boxes.

From Figure 13, it is clear that the first half of b_1 is in the buffer for almost three time steps which means that there are five boxes in the buffer during this time, so the volume of the buffer must be large enough to contain five boxes.

6 A case study at SKF

In Figure 2 the four processes of the roller production channel at SKF are indicated. The aim is to find the appropriate sizes of the buffers located before the new cells and before the fine grinding channels. The buffers before the heat treatment already exists, so their sizes does not have to be investigated.

Since the capacity of the heat treatment machines is limited, some batches of rollers should be sent to the external heat, which is outside the factory. There is no buffer for these batches because as soon as they are ready for the heat treatment, they are sent to the external heat.

Each machine processes special types of rollers according to the shape, size or material, so each batch has a specific route in the channel, and for each process, it should pass a specific machine. The new cells are exceptions since they can process all types of rollers, and there is no specific cell for a specific type of roller. Each batch of rollers should pass exactly one of the new cells. Therefore, in the model (4), one of the new cells is chosen for each batch.

6.1 The case

Because of the huge amount of data required to represent one year's planning the number of variables and constraints of the model (4) sum up to more than the capacity of the computer [1]. Therefore, the model (3) is required to be run to find out which types of rollers should be produced in which month. The result from the model (3) is then used as input data for the model (4). In fact, the model (4) should be run for each month of the year, and the roller types to be produced each month are determined by the model (3). The result from the model (4) for each month affects the input data for the next month. When one machine is still occupied by the batches of the previous month, no batch can enter the machine. The parameters F_m in the constraints (4f) indicate when machine m is free after processing the batches of the previous month, so F_m should be changed in the input data of each month according to the schedule computed for the previous month. In the first month, it is assumed that all machines are available from the first time step. It is *not* guaranteed that the schedule for the whole year is optimal when splitting the model into one month-periods which are solved in sequence; this actually means sub-optimization.

[1] The computations were carried out using AMPL-CPLEX12 on a computer with two 2.50GHz Intel Xeon 5650, each with two cores, and its total memory was 4 Gbyte RAM

The instruction used to find the number of pallets in the buffers is as follows:

1. The model (3) determines which types of rollers should be produced in which months.
2. The heuristic in which F_m is updated according to the schedules of the previous month provides a feasible schedule of the channel for each month which leads to a suitable value of T.
3. The model (4), with updated values of F_m, is solved to find an optimal schedule of the month according to the length of the time steps.
4. Since the model (4) is time-indexed and the schedule developed is an approximation, squeezing the schedule resulting from the model (4) provides a better approximation, and a schedule with a (possibly) shorter lead time.
5. The number of pallets in the buffers during the planning period is determined based on the data from the squeezed schedule. Go to step 2 if the last month has not been scheduled.

6.2 Tests

In our test case, 30 batches should be produced each month. Since there are no "big" batches in the channel under consideration it is not necessary to divide any batch for the heat treatment (see Section 2). There are 17 machines in the channel including the external heat. When a batch is sent to the external heat, it takes five days to get it back to the factory, so the processing time of the external heat on all batches is considered to be five days. The resetting time of the external heat and the transportation times are considered to be zero since these are included in the processing times. The capacity of the external heat is considered to be unlimited, and it can have more than one batch at a certain time, while other machines cannot process more than one batch at a certain time. The transportation times between the machines are assumed to be zero because all the machines of the channel are located in the same building, so the transportation times are short and ignorable.

Solving the model (4) even for one month at a time required computation times that were too long. Although shorter time step intervals, the results in more realistic schedules, for practical purposes it turned out that it is better to choose long time step intervals, and then squeezing the schedule. The schedules obtained will be fairly realistic but require less computing time. In this section, each time step is assumed to be 100 hours, which is chosen by experience (when chosing a time step interval of 50 hours the model (4) did not acheive an optimal schedule even after two days of computing).

6.3 Results and analysis

Figures 14, 15, and 16 show the result from the heuristic, the model (4), and after the squeezing procedure, repsectively, for the first month. Since the result of one whole year contains too many details to be possible to grasp from this type of visualization, the result of one month, which is tweleve times smaller, is shown. The resetting times of the machines are hardly recognizable since they are very short compared to the processing times.

Figure 14 shows the schedule for one month of the heuristic for the production line at SKF. The schedule is feasible for the model (4) which is a time-indexed model. It can be seen that the batches enter the machines at the beginning of the time steps. Each machine does not process more than one batch at a certain time, even the external heat. In the heuristic, the capacity of the external heat is assumed to be one, like for the other machines. When the process of one machine on a batch is finished, the batch can enter the next appropriate machine, as soon as the machine is available, at the beginning of the time step.

We have highlighted one batch, in the schedules of the heuristic, model (4), and the squeezing, indicated by thick arrows to follow the route of the batch in the machines and compare the leadtime of the batch in different schedules. The fourth batch in OKN, a press machine, enters this machine at time step 4. The next process is heat treatment. This batch should be sent to the external heat which is specified by λ_{bm} in the data. The pressing process takes less than one time step, so the batch can enter the external heat at the beginning of the next time step. Since the extrenal heat is already occupied, this batch enters the external heat after a few time steps. The heat treatment process takes longer time in the external heat compared to in Ipsen and Siti, the heat treatment machines in the factory. The heat treatment is finished after 1.2 time steps, so the batch is ready for the next process at the beginning of the next time step. It enters cell 2, which is chosen by random, at the next time step because cell 2 is not occuppied. The final process is the fine grinding process, and the appropriate channel of fine grinding for this batch is R9 which is specified by λ_{bm}. This batch cannot enter R9 at the next time step because R9 is occupied by another batch. This batch should be kept in the buffer for one more time step to enter R9. The leadtime of this batch in the production channel according to the schedule of the heuristic is shown in Table 17.

Figure 15 shows the schedule for one month of the model (4) for the production line at SKF. Since the capacity of the external heat is unlimited, it can process more than one batch at a certain time. In this schedule, the order of the batches on the machines are different from that of the heuristic. The model has chosen one of the new cells for each batch such that the schedule is optimal for the model (4).

The batch which entered OKN at the fourth time step in the schedule of the heuristic enters OKN at the seventh time step in the schedule in Figure 15. After the pressing process, it enters the external heat at the next time step. The external heat is occuppied by another batch, though (the capacity of the external heat is unlimited). Then, it enters cell 1, which is chosen by the model (4) such that the schedule is optimal. This batch enters R9 at the beginning of the next time step because R9 is available. In Table 17, it is clear that the leadtime of this batch is decreased in the schedule of the time-indexed model (4).

Figure 16 shows the squeezed version of the schedule represented in Figure 15, so the order of the batches is the same as Figure 15. In this schedule, when one batch is ready to enter a machine, it can enter as soon as the machine is available. In other words, it is not necessary to enter a machine at the beginning of the time step. Therefore, the schedule in Figure 16 is feasible for the production line at SKF, but it is not feasible for the model (4), which is time-indexed. The batch under consideration enters OKN at time 2.94, which is not the beginning of a time step. From Table 17, it is clear that the leadtime of the batch under consideration in the squeezed schedule is shorter than in the schedule resulting from the model (4).

Table 15 indicates the value of the objective function in each stage, and it is clear that the difference of the objective value between the schedule of the squeezing and that of the model (4) is much larger than the difference between the schedule of the heuristic and that of the model (4). It means that the squeezing algorithm improves the result of the model (4), being an approximation of the optimal schedule, significantly.

Table 16 indicates the leadtime of the channel, i. e., the time interval from the first batch entering the channel until the last batch leaving the channel, for the schedules represented in Figures 14, 15, and 16. The last batch in R4 is the last batch in the channel. The leadtime of the channel is the time when this batch leaves R4. From Table 16 it is clear that by squeezing the schedule of the model (4), the leadtime of the channel is decreased more rapidly compared to the difference between

	heuristic	model (4)	squeezing
Objective value	1448.2	1152.2	463.8

Table 15: The value of the objective function obtained from each method in number of time steps

	heuristic	model (4)	squeezing
leadtime	27.01	22.9	12.656

Table 16: The leadtime for the channel in number of time steps

the leadtime of the schedule of the heuristic and that of the model (4). Therefore, squeezing is a fast algorithm which leads to a good approximation of the optimal schedule of the model (4), with continuous time, in shorter time (instead of solving the scheduling model with shorter time steps.). By chosing shorter time steps and being close to continuous time, the solution time of the model (4) grows rapidly (see Section 5) while the squeezing developes a reasonable approximation of the optimal schedule in acceptable time.

Figures 17 and 18 show the graphs of the buffers located before the new cells and fine grinding channels, respectively. In Figure 2 there are four buffers before the new cells and seven buffers before the fine grinding channels, but in Figures 17 and 18 these buffers are aggregated to two areas of physical space. When a batch enters the buffer, the number of pallets (each pallet including four boxes) in the buffer increases, and when a batch leaves the buffer, the number of pallets decreases. The number of pallets in the buffer is not constant, and there are different number of pallets in the buffer at different times. The buffer should have enough space for the maximum number of pallets placed in this buffer, so the maximum number of pallets indicates the suitable space required for the buffer.

The numbers 37 and 21, which are the maximum number of pallets found for the buffers before the new cells and before fine grinding, are computed from the specific case that we have solved. Certainly SKF will face other (similar, but not equivalent) scenarios. so the recommendation should be larger than 37 and 21, respectively. In practice it is necessary to have some marginal capacity.

On the other hand, even in the squeezed schedule of the channel in Figure 16, there are some empty gaps in the machines. This is not realistic because in practice the machines never stop working. Therefore the value for the sizes of the buffers found in this project might be pessimistic enough to be able to ignore the marginal capacity.

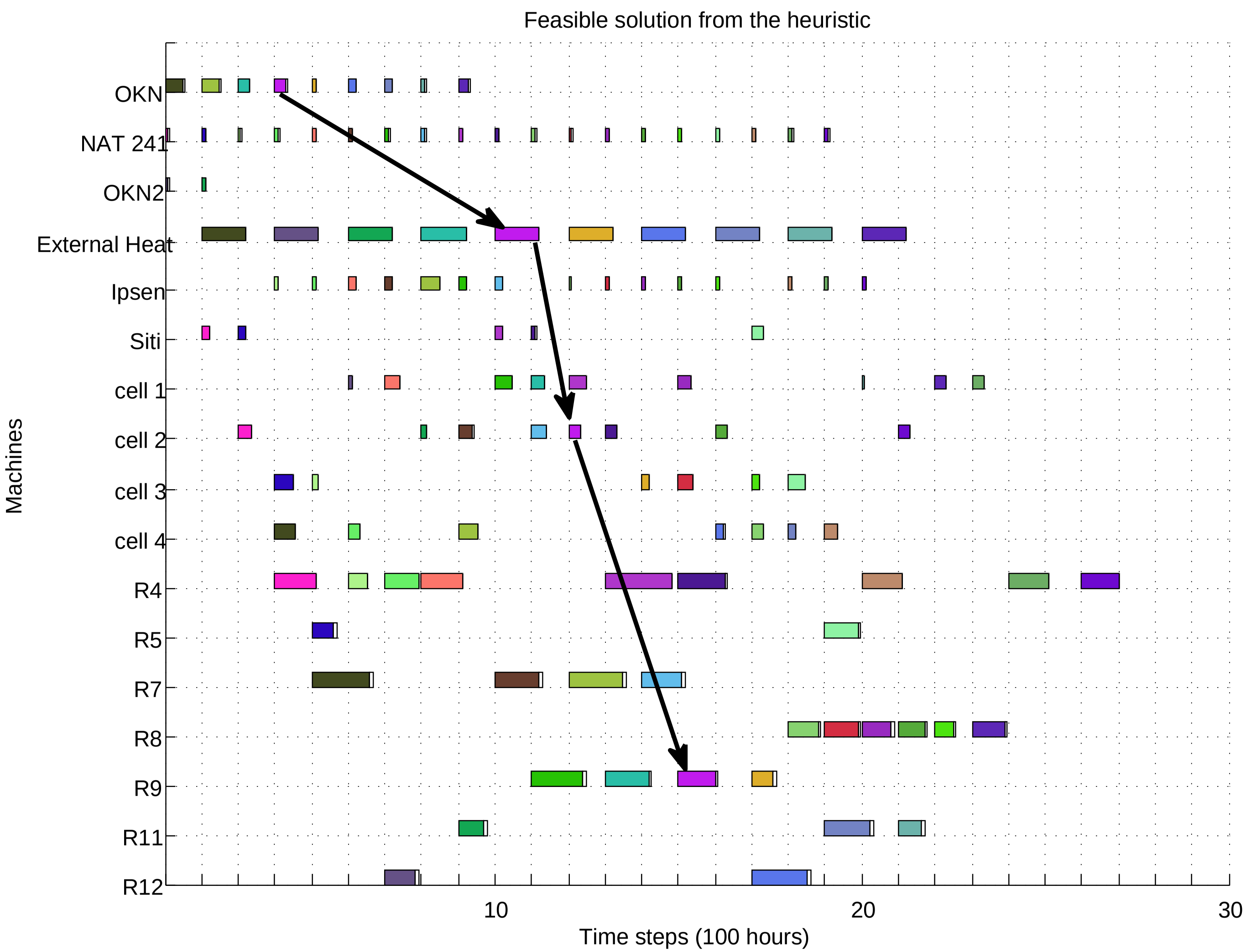


Figure 14: The feasible schedule from the heuristic. A specific batch is followed in the channel.

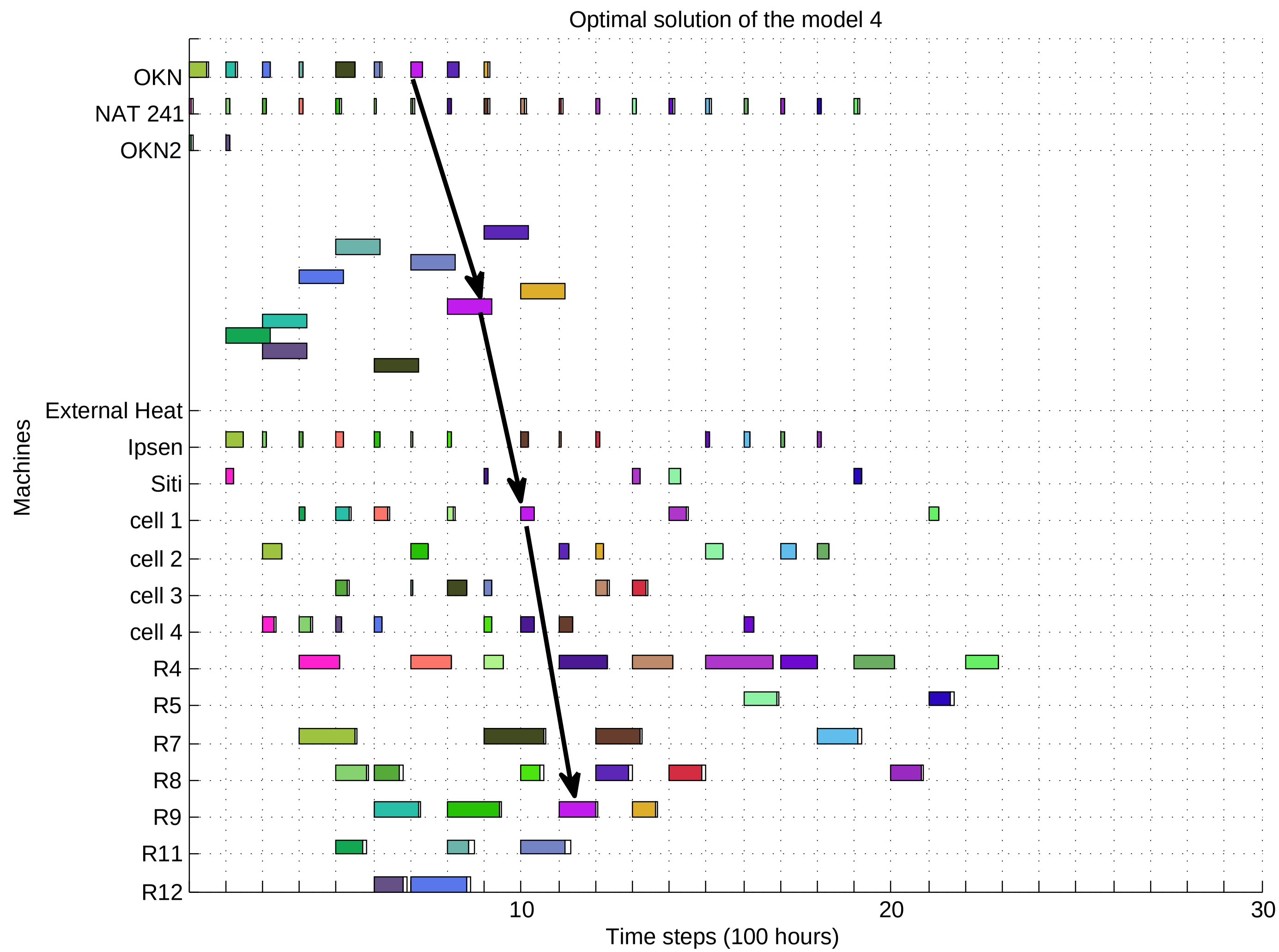


Figure 15: The optimal schedule of the model (4). A specific batch is followed in the channel.

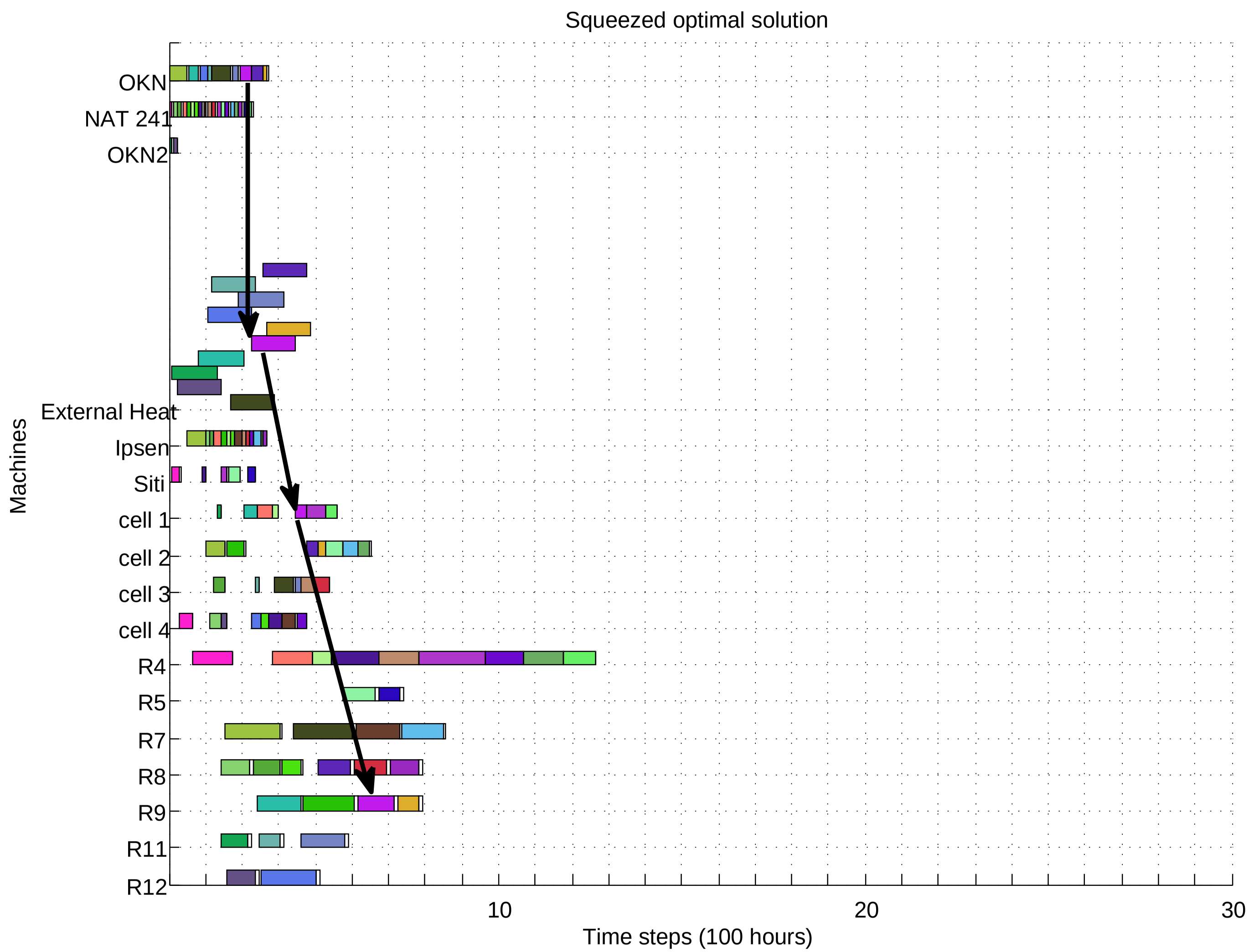


Figure 16: The squeezed optimal schedule. A specific batch is followed in the channel.

	heuristic	model (4)	squeezing
enter	4	7	2.9
leave	16	12	7.2
leadtime	12	5	4.2

Table 17: The leadtime of the batch highlighted by thick arrows in Figures 14, 15 and 16. The time that the batch enters and leaves the channel.

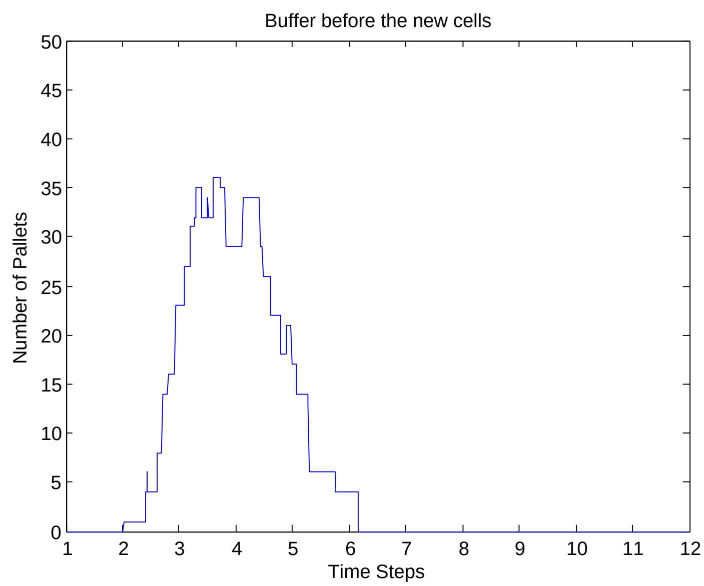


Figure 17: The total number of pallets in the buffers before the new cells.

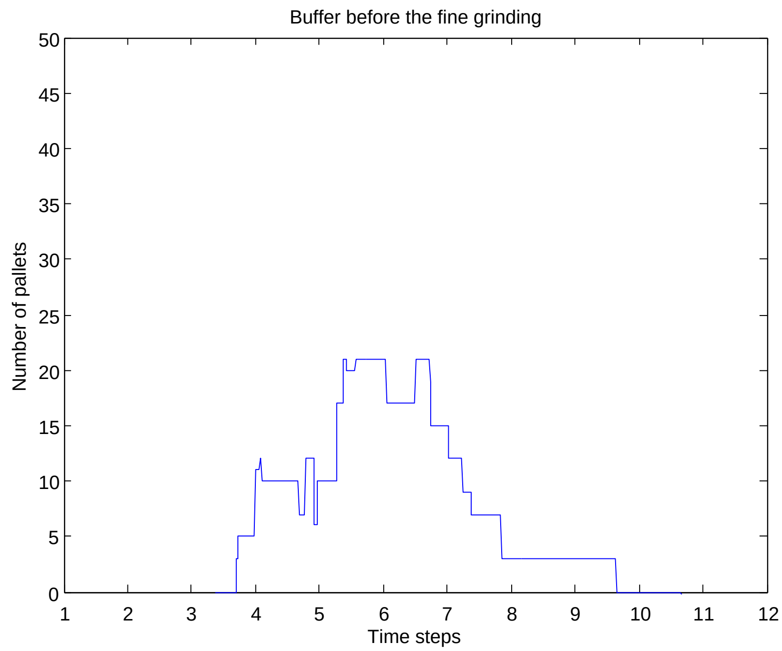


Figure 18: The total number of pallets in the buffers before the fine grinding channels.

7 Summary and conclusion

The main goal of this thesis project is to find the suitable sizes of the buffers of one of the future production channels at SKF's factory in Gothenburg. The buffers should be as small as possible while they should be large enough to store the unfinished products while waiting to be processed further.

The sizes of the buffers are achieved in four steps. First, the heuristic develops a feasible schedule for the channel which leads to a suitable number of time steps for the scheduling model in the next step. In the second step, a scheduling model, in which time is considered to be discrete, provides an optimal schedule with respect to the length of the time step intervals. The model minimizes the leadtimes of the batches, the time period that the batches are in the channel. Since the model is time-indexed, the schedule computed by the model is an approximation of the real case, in which time is continuous. By squeezing the optimal schedule of the model, the lead times of the batches decreases, and the schedule of the channel gets tighter. Finally, the schedules of the buffers are obtained from the schedule of the machines, and the number of pallets in the buffers at all points in time during the planning period is indicated.

The number of the variables and the constraints of the ILP model, for one year's production data of the case at SKF, is too large for the capacity of the computer. Therefore, the optimal schedule of each month of the year is computed separately. An ILP model specifies the types of rollers that should be produced in each month. The result of this model is used to provide the input data of each month for the four main steps represented above.

By decreasing the length of the time steps, the resulting lead times of the batches decreases, but the solution time of the mixed integer linear programming model increases exponentially, which means that the length of the time steps may not be chosen too short, for computational reasons. On the other hand, in each stage of the procedure (i.e., heuristic, ILP model, and squeezing), the lead time decreases, and by decreasing the length of time steps, the decreasing trend of the lead time in each stage gets slower. It takes longer computing time for the ILP model with shorter time step intervals to reach the same result that we can achieve by squeezing the schedule of the model with longer time steps.

8 Future work

In this thesis, we minimized the sizes of the buffers by minimizing the lead times of the batches, so the sizes of the buffers have not been minimized directly. Considering the sizes of the buffers in the objective function of the ILP model can be a subject for the future work. A multi-objective model might be good idea for this project. Considering both the leadtime and the sizes of the buffers in the objective function can develope more practical results.

Another topic to investigate is to employ some mathematical decomposition of the problem, instead of the simple splitting into one-month periods, in order not to risk sub-optimizing. By sequentially computing the optimal schedules of each month, the resulting schedule for one year is not necessarily optimal.

Moreover, the frequencies of the types of batches in one year are considered to be parameters and not variables. Considering the frequencies as variables, to investigate how increasing or decreasing the frequencies may affect the sizes of the buffers, can also be worked for future research. The frequencies should be variables in a master problem and the corresponding subproblems should consist of (essentially) the scheduling problem that we have modelled here. The problem is that the channel under consideration produces rollers which should be installed in the bearings, which is the main product of SKF. Changing the frequency is not independent for the channel under consideration. The frequency should be regulated with the other parts of the factory.

Since the areas between the machines are limited, It is possible to add some constraints to the model and include limitations for the sizes of the buffers. It means that the sizes of the buffers may not exceed a specified space.

Since the goal of the factory is to satisfy the customers, a good idea is to regulate the schedule according to the demand of the customers.

References

[1] M. S. BAZARAA, J. J. JARVIS, H. D. SHERALI, *Linear Programming and Networks Flows*, John Wiley & Sons Inc, 2nd edition, 1990.

[2] E. H. BOWMAN, *The schedule-sequencing problem*, Operations Research, 7 (1959), pp. 621–624.

[3] P. BRUCKER, *Scheduling Algorithms* , Springer-Verlag, Berlin Heidelberg, 5th ed., 2007.

[4] P. FATTAHI, M. SAIDI MEHRABAD, AND F. JOLAI, *Mathematical modeling and heuristic approaches to flexible job shop scheduling problems*, Journal of Intelligent Manufacturing, 18 (2007), pp. 331–342.

[5] A. JAIN AND S. MEERAN, *Deterministic job-shop scheduling: Past, present and future*, European Journal of Operational Research, 113 (1999), pp. 390–434.

[6] S. KEDAD-SIDHOUM, Y. R. SOLIS, AND F. SOURD, *Lower bounds for the earliness-tardiness scheduling problem on single and parallel machines*, European Journal of Operational Research, 189 (2008), pp. 1305–1316.

[7] J. LUNDGREN, M. RÖNNQVIST, AND P. VÄRBRAND,*Optimization*, Studentlitteratur, 2010.

[8] A. S. MANNE, *On the job-shop scheduling problem*, Operations Research, 8 (1960), pp.219–223.

[9] J. ROSLÖF, I. HARJUNKOSKI, T. WESTERLUND, AND J. ISAKSSON, *Solving a large-scale industrial scheduling problem using MILP combined with a heuristic procedure*, European Journal of Operational Research, 138 (2002), pp. 29–42.

[10] J. SOUSA AND L. WOLSEY, *A time indexed formulation of non-preemptive single machine scheduling problems*, Mathematical Programming, 54 (1992), pp. 353–367.

[11] E. STAFFORD, JR., F. T. TSENG, AND J. N. D. GUPTA, *Comparative evaluation of MILP flowshop models*, The Journal of the Operational Research Society, 56 (2005), pp. 88–101.

[12] K. THÖRNBLAD, *On the Optimization of Schedules of a Multitask Production Cell*, Licentiate thesis, Chalmers University of Technology, Sweden, 2011.

[13] J. VAN DEN AKKER, C. HURKENS, AND M. SAVELSBERG *Time-indexed formulations for machine scheduling problems: Column generation*, INFORMS Journal on Computing, 12 (2000), pp. 111–124.

[14] H. M. WAGNER, *An integer linear programming model for machine scheduling*, Naval Research Logistics Quarterly, 6 (1959), pp.131–140.

[15] L. A. WOLSEY *MILP modelling of changeovers in production planning and scheduling problems*, European Journal of Operational Research, 99 (1997), pp. 154–165.

[16] Z. ZHU AND R. HEADY, *Mimimizing the sum of earliness/tardiness in multi-machine scheduling: a mixed integer programming approach*, Computers & Industrial Engineering, 38 (2000), pp. 297–305.

[17] C.ÖZGÜVEN, L. ÖZBAKIR, AND Y. YAVUZ, *Mathematical models for job-shop scheduling problems with routing and process plan flexibility*, Applied Mathematical Modelling, 34 (2010), pp. 1539–1548.

Printed and Bound at
Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg
2012