DISSERTATION

Scheduling of a Multi-Line Steel Hot Rolling Mill

Ausgeführt zum Zwecke der Erlangung des akademischen Grades eines
Doktors der technischen Wissenschaften (Dr.techn.)

unter der Leitung von
Associate Prof. DI Dr.techn. Andreas Steinböck
E376
Institut für Automatisierungs- und Regelungstechnik

eingereicht an der
Technischen Universität Wien
Fakultät für Elektrotechnik und Informationstechnik

von
Dipl.-Ing. Martin Kowalski BSc
Matrikelnummer: 9926469

Wien, im Juli 2024

Studiendekan:	Univ.-Prof. Dr.sc. Silvan Schmid
Betreuer:	Associate Prof. DI Dr.techn. Andreas Steinböck
Tag des Kolloquiums:	3. Dezember 2024
Prüfungsvorsitzender:	Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Thilo Sauter
Erster Gutachter:	Associate Prof. Derik le Roux
Zweiter Gutachter:	Univ.-Prof. Dipl.-Ing. Dr.techn. Andreas Kugi

Vorwort

Diese Dissertation entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Automatisierungs- und Regelungstechnik (ACIN) an der TU Wien. Das Thema der Arbeit ergab sich aus einer Forschungskooperation mit der voestalpine BÖHLER Edelstahl GmbH & Co KG, bei der ich mich für die Unterstützung und Finanzierung bedanke.

Ich bedanke mich bei Associate Prof. Dr. Andreas Steinböck für zahlreiche Inputs, Diskussionen und die gute sowie konstruktive Betreuung meiner Arbeit. Weiters bedanke ich mich bei Univ.-Prof. Dr. Andreas Kugi für die Aufnahme am Institut und auch für seine vielfältige Unterstützung meiner Forschungstätigkeit. Bei Herrn Associate Prof. Dr. Derik le Roux bedanke ich mich für die Erstellung seines Gutachtens zu dieser Arbeit.

Auf Seiten der voestalpine BÖHLER Edelstahl GmbH & Co KG möchte ich mich herzlich bei Dr. Josef Bernauer, DI Michael Magritzer, DI Florian Dierer und Dr. Sigfried Kleber für die konstruktive und wertschätzende Zusammenarbeit sowie den stets freundlichen Kontakt bedanken. Meine Besuche im Werk und die Erarbeitung der praktischen Erfordernisse waren eine spannende Ergänzung meiner wissenschaftlichen Arbeit.

Weiters bedanke ich mich bei allen Kollegen am Institut für das freundliche Umfeld, den fachlichen Austausch, zahlreiche spannende Diskussionen und schöne Zeiten abseits der Arbeit. Besonders bedanken möchte ich mich bei meinen Kollegen vom 4er-Büro, die den gemeinsamen Arbeitsalltag stets bereicherten.

Schließlich gilt mein Dank meiner Familie, die mich stets unterstützte und in meinen Vorhaben bestärkte. Ich danke besonders meiner Frau Emi, dass sie mir in zahlreichen arbeitsintensiven Zeiten den Rücken freigehalten hat. Unseren Kindern Kodai und Yuma danke ich, dass sie mich den Blick für das Wesentliche nicht vergessen lassen.

Kurzfassung

Die vorliegende Arbeit beschäftigt sich mit der Prozesszeit- und Produktreihenfolgenoptimierung eines Mehrlinienwalzwerks der voestalpine BÖHLER Edelstahl GmbH & Co KG. In der betrachteten Anlage werden hochlegierte Stähle verarbeitet. Kennzeichnend für die Produktion sind eine große Produktvielfalt, kleine Losgrößen und eine alternierende Produktion an einer Profilwalzlinie und einer Flachwalzlinie. Die Arbeit behandelt zwei wesentliche Problemstellung der Produktion: 1. die Bestimmung des frühestmöglichen Startzeitpunkts jedes Produkts bei gegebener Produktreihenfolge, 2. die Optimierung der Produktauswahl und Reihenfolge für gegebene Produktionsaufträge.

Bei der Fertigung durchlaufen die Produkte die Anlage entlang verschiedener Routen. Es wird zunächst das Anlagen- und Prozesszeitmodell beschrieben. In diesem Modell wird der Prozessverlauf eines Produkts durch diskrete Zeitpunkte beschrieben, zu denen die Produktenden entlang der Fertigungslinien definierte Checkpoints passieren. Das entwickelte Modell erlaubt eine für die betrachtete Planungsaufgabe ausreichend detaillierte Erfassung des Prozessablaufs jedes Produkts, sowie eine hinreichend schnelle Simulation der Fertigung einer gegebenen Reihenfolge von Produkten.

Derzeit werden die Produkte nach manueller Freigabe durch das Bedienpersonal gestartet, d. h. der Anlage zugeführt. Die zeitlichen Abstände zwischen den Produktstarts werden dabei anhand von Erfahrungswerten festgelegt. Um diese Startzeitpunkte genauer planen und in weiterer Folge Leerlaufzeiten zwischen den Produkten verringern zu können, wird eine Methode zur Berechnung optimaler Startzeitpunkte präsentiert. Die Methode erlaubt die Berücksichtigung von sowohl deterministischen als auch stochastischen Bearbeitungszeiten. Das Ziel ist die gleichzeitige Minimierung der Gesamtfertigungszeit und der Anzahl an Verletzungen von sogenannten no-wait Beschränkungen. Verletzungen von no-wait Beschränkungen treten auf, wenn ein Produkt zu früh gestartet wird und es deshalb im Bearbeitungsverlauf mit stochastischen Prozesszeiten zu einer ungeplanten Wartezeit kommt. Typische Prozessschritte, bei denen ungeplante Wartezeiten unerwünscht oder inakzeptabel sind, sind Erwärm-, Glüh- oder Warmumformprozesse. Die Optimierung der Startzeitpunkte für einige wenige Maschinen kann analytisch erfolgen. Für die Optimierung von Systemen mit einer höheren Anzahl von Maschinen wird eine auf Monte-Carlo Simulationen basierende Methode beschrieben. Die Optimierung eines Fertigungsprozesses mit stochastischen Prozesszeiten in einem Warmwalzwerk wird anhand eines praktischen Beispiels gezeigt. Schließlich werden die Vorteile der Optimierung unter systematischer Berücksichtigung der stochastischen Prozesszeiten gegenüber einfacheren ad-hoc Methoden durch einen Vergleich verdeutlicht.

Zur Startzeitpunktplanung anhand des entwickelten Anlagen- und Prozesszeitmodells wurde ein Prototyp entwickelt und dem Bedienpersonal der Anlage zur Verfügung gestellt. Dabei wurden die frühestmöglichen Startzeitpunkte der nachfolgenden Produkte berechnet und entsprechend angezeigt. Die Funktion wurde im praktischen Betrieb erprobt und validiert. Aufgrund der positiven Ergebnisse wurde die Startzeitpunktplanung in das Produktivsystem übernommen, wobei die Integration in das bestehende System während des Verfassens dieser Arbeit noch im Gange war.

Bei der Erstellung von Einsatzplänen werden Jobs verplant, welche aus ein oder mehreren zusammengehörenden Produkten bestehen. Diese Jobs werden innerhalb von Gruppen verplant. Die Gruppen werden alternierend an der Profil- und an der Flachwalzblocklinie gefertigt. Eine Gruppe, die an der Profilwalzlinie der Anlage gefertigt wird, besteht aus Jobs eines oder mehrerer zusammengehörender Enddurchmesser. Nach der Fertigung einer Gruppe an der Profilwalzlinie wird diese umgerüstet. Während des Umrüstvorgangs wird eine aus einem Job Pool passend zusammengestellte Gruppe von Jobs an der Flachwalzlinie gefertigt. Die Erstellung eines Produktionsplans erfordert eine Jobauswahl, Reihung und für manche Jobs die Festlegung einer Erwärmungsmethode. Ein optimaler Produktionsplan minimiert unproduktive Zeiten während der Produktion und nützt die Umrüstzeiten an der Profilwalzlinie zur Fertigung an der Flachwalzlinie bestmöglich aus. Gleichzeitig müssen im Produktionsplan zusätzliche Regeln, z.B. hinsichtlich der Anordnung von Produktmarken, berücksichtigt werden. Das resultierende Optimierungsproblem ist eine Kombination aus Problemen ähnlich dem Traveling Salesperson Problem und dem Orienteering Problem. Für einen gegebenen Produktionsplan können basierend auf dem entwickelten Anlagen- und Prozessmodell unproduktive Zeiten sowie die Ausnutzung der Umrüstvorgänge ermittelt werden. Weiters werden in einer Kostenfunktion alle zusätzlich erforderlichen Regeln mit Prioritäten versehen und berücksichtigt. Die Kostenfunktion wird in einem speziell für dieses Problem entwickelten Algorithmus, bestehend aus einer Simulated Annealing Metaheuristik gefolgt von einem Algorithmus zur lokalen Suche, minimiert. Die effektive Anwendung und der Nutzen der Optimierung wird anhand eines realen Anwendungsfalls und einem Vergleich mit einem manuell erstellten Einsatzplan gezeigt.

Der entwickelte Optimierungsalgorithmus wurde auf Planungen der realen Produktion angewandt. Die optimierten Einsatzpläne wurden von erfahrenen Planern überprüft und bewertet. Die Schwierigkeiten im praktischen Einsatz liegen insbesondere in der Vielzahl von Produkten, für die gesonderte Vorgaben und Produktionsregeln berücksichtigt werden müssen. Im Rahmen dieser Arbeit war es nicht möglich, all diese Regeln zu dokumentieren und in weiterer Folge in den Optimierungsalgorithmus zu integrieren. Dennoch waren die Rückmeldungen des Personals positiv. Hervorgehoben wurden die Zeitersparnis bei der automatisierten Erstellung der Einsatzpläne (auch wenn diese aufgrund einzelner Produkte manuell kontrolliert und gegebenenfalls angepasst werden müssen) sowie die Minimierung der unproduktiven Zeiten. Zur Minimierung der unproduktiven Zeiten muss eine Vielzahl an Produktionsparametern der Produkte gleichzeitig berücksichtigt werden, was manuell schwierig ist, aber durch ein rechnergestütztes Planungstool vorteilhaft erfolgen kann.

Abstract

The present work deals with optimized scheduling of processing times and product sequences for a multi-line steel hot rolling mill of voestalpine BÖHLER Edelstahl GmbH & Co KG. In the considered plant, high-alloyed steel is processed. The production is characterized by a wide variety of products, small batch sizes, and an alternating production on a profile rolling line and a flat rolling line. The work addresses two main production problems: 1. determining the earliest possible start time for each product in a given product sequence, 2. optimizing product selections and sequences for given scheduling problems.

During production, the products follow different routes through the plant. First, the model of the production process and the plant is described. This model represents the processing of a product by discrete points in time when the product ends pass defined checkpoints. In view of the considered scheduling task, the model allows for a sufficiently detailed representation of the production process of each product and facilitates fast simulation of the production of a given sequence of products.

Currently, the products are started, i. e. submitted to the plant, after a manual command by the operating personnel. In this way, the time intervals between the starts of consecutive products are chosen based on empirical values. To more accurately plan these start times and reduce idle times between products, a method for calculating optimal start times is presented. The method allows the consideration of deterministic and stochastic processing times. The objective is to simultaneously minimize the total production time and the number of violations of so-called no-wait constraints. Such violations occur when a product is started too early, leading to unplanned waiting time during processing. Typical process steps where unplanned waiting times are undesired or unacceptable are heating, annealing-, or hot forming processes. Two methods to solve the optimization problems were developed. The optimization of start times with only a few machines can be solved by analytical computation. For the optimization of systems with more machines, a method based on Monte Carlo simulations is described. The optimization of a production process in a hot rolling mill is demonstrated, and the advantages of considering stochastic process times systematically over simpler ad-hoc methods are illustrated by a comparison.

For the practical start time planning, a prototype was created and made available to the operating personnel of the plant. The earliest possible start times for the subsequent products were calculated and appropriately displayed. The function was tested and validated in practical operation. Due to the positive results, the start time scheduling was integrated into the production system, with the integration into the existing system still ongoing at the time this work was written.

In production schedules, jobs which consist of one or more products that belong together are scheduled in groups. The groups are alternately processed at the profile and at the flat rolling line. A group processed on the profile rolling line consists of jobs with one or more related final diameters. After manufacturing a group at the profile rolling line, this line is retooled for the next final diameter. During this retooling, a group of jobs selected from a job pool is processed at the flat rolling line. Thus, the creation of production schedules requires job selection, sequencing, and, for some jobs, the assignment of a heating method. An optimal schedule minimizes unproductive times during production and utilizes the retooling times of the profile rolling line as best as possible for manufacturing at the flat rolling line. Additionally, in a schedule additional rules, such as the arrangement of product grades, must be considered. The resulting optimization problem is a combination of problems similar to the Traveling Salesperson Problem and the Orienteering Problem. Based on the developed model of the production process and the plant, unproductive times and the utilization of retooling times can be determined for a given schedule. Furthermore, a cost function takes all additional required rules into account with different priorities. The costs function is minimized by a tailored optimization algorithm, which consists of a simulated annealing metaheuristic followed by a local search. The effective application and benefit of this algorithm are demonstrated in a case study with a comparison to a manually created production schedule.

The developed optimization algorithm was applied to real production planning. The optimized schedules were reviewed and evaluated by experienced planners. The practical challenges lie primarily in the multitude of products, each requiring specific guidelines and production rules to be considered. Within the scope of this work, it was not possible to document all these rules and subsequently integrate them into the optimization algorithm. Nevertheless, the feedback from the staff was positive. They highlighted the time savings in the automated creation of

schedules (even though these must be manually checked and adjusted if necessary for certain products) as well as the minimization of unproductive times. To minimize unproductive times, a multitude of production parameters of the products must be considered simultaneously, which is difficult to manage manually but can be effectively handled by a computer-aided planning tool.

Contents

Nomenclature

Acronyms and abbreviations

ATSP	Asymmetric traveling salesperson problem
CDF	Cumulative distribution function
CP	Constraint programming
F-group	Group of jobs processed at the flat rolling line
FP	Finished product
IM	Intermediate product (after reversing rolling stand)
INDU	Induction furnace
MIP	Mixed integer programming
OP	Orienteering problem
PDF	Probability density function
P-group	Group of jobs processed at the profile rolling line
PM	Primary product (at the beginning of the processing)
TSP	Traveling salesperson problem
VRP	Vehicle routing problem
WBF	Walking beam furnace

Function and operators

$c(\Theta)$	Cost function evaluating a schedule Θ
$\mathrm{E}(\cdot)$	Expected value of a random variable
$H(t)$	Heaviside function
$\mathrm{mean}(\cdot)$	Mean value
$\mathcal{N}(\Theta)$	Neighborhood function applied to schedule Θ
$\mathcal{N}_F(\boldsymbol{\sigma}_i, \boldsymbol{\tau}_i)$	Neighborhood function applied to the schedule $\boldsymbol{\sigma}_i$ and furnace assignments $\boldsymbol{\tau}_i$ of a group i processed at the flat rolling line
$\mathcal{N}_P(\boldsymbol{\sigma}_i, \boldsymbol{\tau}_i)$	Neighborhood function applied to the schedule $\boldsymbol{\sigma}_i$ and furnace assignments $\boldsymbol{\tau}_i$ of a group i processed at the profile rolling line
$O(\cdot)$	Order of approximation
$\Pr(A)$	Probability of an event A
$\mathrm{unif}(a, b)$	Uniform distribution with the parameters a and b

Latin symbols

A_fp	Cross sectional area of a finished product
A_im	Cross sectional area of a product after the reversing rolling stand
A_pm	Cross sectional area of a product at the beginning of the processing
c_i	Cost of group i
d_i	Required retooling time of the profile rolling line after P-group i
e_i	Time by which the retooling duration of a group i processed at the profile rolling line is exceeded
$F_{P_i^k}$	Cumulative distribution function of the random process time of product i at machine k
$f_{P_i^k}$	Probability density function of the random process time of product i at machine k
$\boldsymbol{I}_\mathrm{F}$	Set of indices of the groups processed at the flat rolling line
$\boldsymbol{I}_\mathrm{P}$	Set of indices of the groups processed at the profile rolling line
J	Cost function value
k_T	Number of neighbors examined for a constant temperature
l_fp	Length of the finished product
l_im	Product length after the reversing rolling stand
l_pm	Product length at the beginning of the processing
m	Number of machines or checkpoints
n	Number of products or jobs
$\overline{N}_\mathrm{c}$	Mean value of the number of conflicted products
N_c	Random number of conflicted products
n_i	Number of jobs in group i
n_s	Number of samples for a Monte-Carlo simulation
n_i^p1	Number of violations of rules of priority 1 in group i
n_i^p2	Number of violations of rules of priority 2 in group i
n_i^p3	Number of violations of rules of priority 3 in group i
n_i^p4	Number of violations of rules of priority 4 in group i
p_j	Process time of job j
$p_i^{\mathrm{blk},k,l}$	Blocking time of checkpoint l after checkpoint k was passed by the tail of product i
$p_{i,j}^{\mathrm{blk},k,l}$	Blocking time of checkpoint l after checkpoint k was passed by the tail of job j in group i
p_i^exit	Time required by product i to pass the last checkpoint in the route

$p_{i,j}^{\mathrm{exit}}$	Time required by job j in group i to pass the last checkpoint in the route
$p_{i,j}^{\mathrm{HB,max}}$	Maximum retention time of a product of job j in group i inside the heating box
$p_{i,j}^{\mathrm{HB,min}}$	Minimum retention time of a product of job j in group i inside the heating box
$p_{i}^{\mathrm{in},k,l}$	Time required by product i to pass checkpoint k
$p_{i,j}^{\mathrm{in},k,l}$	Time required by job j in group i to pass checkpoint k
P_i^k	Random process time of product i at machine k
${}_{s}p_i^k$	Sample s of the process time of product i at machine k
$p_i^{\mathrm{max},k}$	Maximum value of the random process time of product i at machine k
$p_i^{\mathrm{min},k}$	Minimum value of the random process time of product i at machine k
$p_i^{\mathrm{thru},k,l}$	Time required by product i to travel between checkpoints k and l
$p_{i,j}^{\mathrm{thru},k,l}$	Time required by job j in group i to travel between checkpoints k and l
q	Percentile
q^*	Optimal percentile
r	Cooling rate
$\boldsymbol{r}_i$	Route of product i
$\boldsymbol{r}_{i,j}$	Route of job j in group i
$\boldsymbol{r}_{i,j,l}$	Route of product l of job j in group i
r_i^k	Checkpoint at position k in the route of product i
$\boldsymbol{S}_{\mathrm{F}}$	Set of jobs in the pool of all groups processed at the flat rolling line
$s_{i,j}$	Total unproductive time between job j in group i and the next job
$\bar{s}_{i,j}$	Time during which checkpoint 12 is not occupied between job j in group i and the next job
$\boldsymbol{S}_{\mathrm{P},i}$	Set of jobs group i processed at the profile rolling line
$s_{i,j}^{\mathrm{INDU}}$	Additional unproductive time when job j in group i is heated in the induction furnace
T	Annealing parameter temperature
$\boldsymbol{T}$	Subset of jobs
$\boldsymbol{T}_{i,j}$	Set of admissible furnaces of job j in group i
t_{max}	Maximum number of iterations without improvement
$t_i^{0,k}$	Nominal entry time of product i into machine k
$t_i^{\mathrm{H},k}$	Time at which the head of product i passes checkpoint k
$t_{i,j}^{\mathrm{H},k}$	Time at which the head of job j in group i passes checkpoint k

$t_{i,j,l}^{\mathrm{H},k}$ Time at which the head of product l of job j in group i passes checkpoint k

T_i^k Random time at which product i enters machine k

$\hat{T}_i^k$ Random time at which product i enters machine k in the case of conflict-free processing

t_i^k Time at which product i enters machine k

$\hat{t}_i^k$ Time at which product i enters machine k in the case of conflict-free processing

${}_s t_i^k$ Sample s of the entry time of product i into machine k

${}_s \hat{t}_i^k$ Sample s of the entry time of product i into machine k in the case of conflict-free processing

$\overline{T}_n^{m+1}$ Mean value of the makespan

$t_i^{\max,k}$ Maximum possible entry time of product i into machine k

$t_i^{\min,k}$ Minimum possible entry time of product i into machine k

$t_i^{\mathrm{R},k}$ Release time of checkpoint k after product i

$t_{i,j}^{\mathrm{R},k}$ Release time of checkpoint k after job j in group i

$t_i^{\mathrm{T},k}$ Time at which the tail of product i passes checkpoint k

$t_{i,j}^{\mathrm{T},k}$ Time at which the tail of job j in group i passes checkpoint k

$t_{i,j,l}^{\mathrm{T},k}$ Time at which the tail of product l of job j in group i passes checkpoint k

$u_{i,j}^{\mathrm{INDU}}$ Unproductive time within job j in group i heated in the induction furnace

$u_{i,j}^{\mathrm{WBF}}$ Unproductive time within job j in group i heated in the walking beam furnace

u_j Integer variable used in Miller-Tucker-Zemlin-constraints

$u_{\max}$ Total maximum number of iterations

w_{c} Weighting of the number of conflicted products

w_{e} Weighting of excess durations of P-group retooling times

$w_{j,k}$ Penalty for arranging job k after job j

w_{p1} Weighting of the number of violations of rules of priority 1

w_{p2} Weighting of the number of violations of rules of priority 2

w_{p3} Weighting of the number of violations of rules of priority 3

w_{p4} Weighting of the number of violations of rules of priority 4

w_{s} Weighting of the unproductve setup times

w_{t} Weighting of the makespan

$x_{j,k}$ Binary variable defining if job j is followed by job k

Greek symbols

γ_i	Probability of conflict free processing of product i
γ_{set}	Set probability of conflict free processing
γ^*_{set}	Optimal probability of conflict free processing
Θ	Schedule of multiple groups
Θ'	Permutation of schedule Θ
$\boldsymbol{\sigma}$	Sequence of jobs or products
$\boldsymbol{\sigma}_i$	Sequence of jobs of group i
$\boldsymbol{\sigma}'_i$	Permutation of sequence of jobs in group i
$\sigma_{i,j}$	Job at position j in group i
σ_j	Product at position j
$\boldsymbol{\tau}_i$	Furnace assignment of jobs of group i
$\boldsymbol{\tau}'_i$	Permutation of furnace assignment of jobs in group i
$\tau_{i,j}$	Furnace assigned to job j of group i

Chapter 1.

Introduction

Hot rolling is an essential processing step in the steel manufacturing industry that transforms large slabs of metal into various shapes, sections and sizes, such as plates, sheets, bars, and wire, which are used in many industrial applications. In the steel hot rolling mill of voestalpine BÖHLER Edelstahl GmbH & Co KG (hereinafter referred to as BÖHLER), high alloyed steel is processed to flat bars, round bars, and wire. The production is characterized by a large variety of products and often small batch sizes (also individual products). This makes it difficult for the operators to both generate good production schedules and accurately consider process times during the actual production, leading to significant temporal variability in production times and output.

In order to support the operating personnel and subsequently to increase the production output of the system, the current work addresses the following main topics:

1. Planning of optimal product start times under deterministic and stochastic process times

2. Development of an algorithm for the optimization of product sequences

1.1. The multi-line steel hot rolling mill of BÖHLER

The plant considered in this work produces wires, profiles, and flat products from various stainless steel grades. Important production steps are the heating of square cross-section steel billets and the subsequent hot rolling to the final shape and dimensions. Most of the billets are heated in a gas-fired walking beam furnace (WBF). Some billets can alternatively be heated in an induction furnace (INDU) with three chambers. A few billets can only be heated in the INDU.

The hot billets are then rolled out to an intermediate square cross-section by a reversing roughing mill. Finally, the products are processed to flat steels at the flat rolling line or to steel bars or wires in the profile rolling line. Figure 1.1 shows an outline of the considered multi-line hot rolling mill.

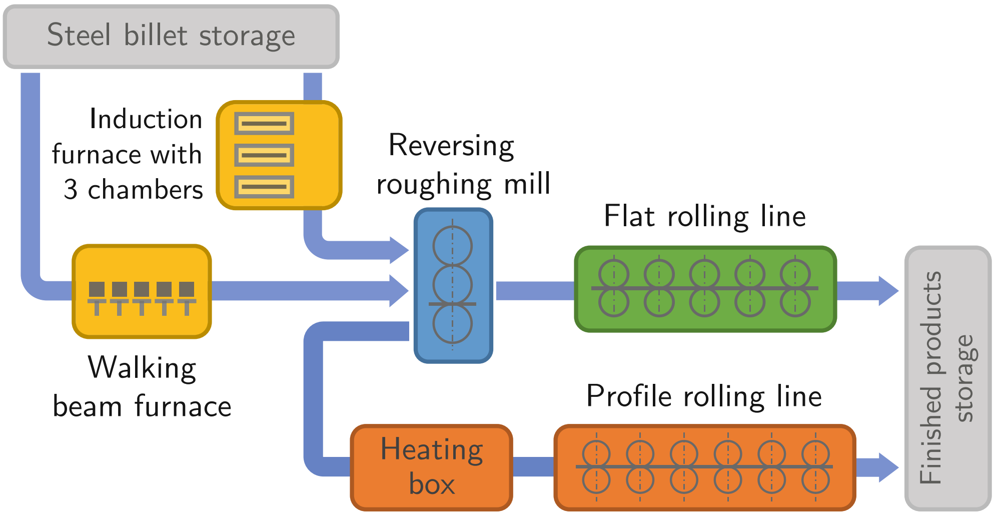

Figure 1.1.: Machines and material flow in the multi-line hot rolling mill.

If the intermediate products are further processed at the flat rolling line, the products are rolled out to rectangular steel bars and flat steels. If processed at the profile rolling line, the products are first transported through a heating box, where they can be reheated or kept at a constant temperature. The products are then rolled out to steel bars or wires, passing an appropriate number of rolling stands of the profile rolling line.

Products having identical production parameters (apart from their lengths) are grouped together into a job. A job can also consist of just one product. All products of a job are processed consecutively in a defined sequence without interruption. In this work, the splitting of jobs is not considered, because in the considered plant it occurs only in exceptional cases, e. g., if jobs consist of a large number of products.

The profile line is regularly retooled, i. e., rolling stands must be activated or deactivated when the final diameter of jobs changes. Therefore, jobs with a common final diameter are combined into a group, referred to as *P-group*, and consecutively processed. During the subsequent retooling of the profile rolling line, a group of jobs is processed at the flat rolling line, referred to as *F-group*. At the flat rolling line, the retooling durations are negligible. The sequence of the final diameters produced at the profile rolling line, i. e. the sequence of P-groups, is cyclic and fixed by a higher-level production planning system.

1.2. Scheduling problem

1.2.1. Start time planning

During the production of a given job and therefore product sequence, the processing of each product is individually started by a responsible operator. After the start request by the operator and if the product was heated in the WBF, the hot product is ejected and enters the roller table before the reversing roughing mill. If the product is heated in the INDU, the cold product is transported by a crane from a starting point to the roller table of the INDU.

The start time of each product must be planned such that there are no unplanned waiting times during processing. Unplanned waiting times would result in an undesired cool down of the hot products. Numerous factors, such as

- the length of the production line,
- individual product lengths and therefore individual process times of each product,
- compliance with requirements of the production process, e. g., safety distances between products,
- a large variety of product grades,
- special requirements of the manufacturing process of individual grades, e. g., additional cooling times,
- frequent changes of produced grades because of small batch sizes,

make it challenging to optimally choose the start times of the products. Currently, the start times are selected based on empirical values. All products of a job are started at the same time span after the previous product, meaning that the individual product lengths are not taken into account. The heating box, where most products can stay for a longer duration, helps to compensate for time delays in production at the profile rolling line. Nevertheless, due to the factors mentioned above, additional buffer times between product start times are necessary to avoid unplanned waiting times during production. These buffer times reduce the possible throughput of the plant.

1.2.2. Job selection and sequencing

Schedules, i. e., the selection and sequence of jobs, are in general created for several P- and F-groups, which follow each other in alternating order. A scheduling problem is defined by

- a sequence of P-groups with a set of jobs for every group,
- a pool of selectable jobs for all F-groups,
- the retooling durations between the consecutive P-groups.

For each P-group, all jobs have to be scheduled. A P-group can consist of jobs with more than one final diameter. In this case, only minor retooling is needed when the final diameter changes within the group. Hence, the production is not switched to the flat rolling line. Within a P-group, all jobs of the same diameter have to be consecutively processed. However, the different diameters can be arranged in an arbitrary order. Usually P-groups consist of a few jobs of a single diameter to up to about 200 jobs in the case of several different diameters.

For all F-groups, a common pool of jobs is given. For every F-group, jobs are selected from this pool in such a way that the corresponding retooling time of the profile rolling line is maximally utilized but not exceeded. In addition, all jobs should be arranged such that the unproductive times between the jobs are minimized.

The generation of schedules is currently done by experienced operators a few days before actual production. A schedule covers a manufacturing duration of one to two days. Currently, the estimation of the processing durations of individual jobs is based on empirical values. A multitude of production parameters and rules that need to be considered, make the manual generation of effective schedules challenging and time-consuming. Some of these rules must be strictly adhered to, while others should be adhered to if possible. Because of these difficulties, the quality of the generated schedules varies depending on the experience of the operator.

1.2.3. Problem description

In the scientific community, a standardized notation has evolved that captures the structure of many scheduling problems [49]. Within this notational framework, a finite number of machines and products is considered. In contrast, the model of the production process addressed in this work consists of checkpoints and products of finite length, which can occupy multiple sections of the plant at a time. Therefore, the standardized notation cannot exactly describe this problem. Approximately the considered problem can be described as *job shop* with the following characteristics:

- *Precedence constraints* require that certain groups have to be processed in a given order.

- *Sequence-dependent setup times* separate consecutive jobs.
- *Blocking* occurs because there are no buffers (except for the heating box), and there can never be more than one product at the same place along the production lines.
- *No-wait constraints* prevent heated products to cool down for undefined times during their processing.

Blocking and no-wait constraints are considered in the start time planning and process simulations of given product sequences. Precedence constraints are implicitly considered when defining the groups and their order in an optimization problem. Sequence-dependent setup times are regarded as unproductive times between consecutive products or jobs, respectively. Their definition and estimation will be described in Section 4.3.1.

The optimization of a job sequence within a P-group is similar to the asymmetric traveling salesperson problem (ATSP) [25]. The optimization of the job selection for and sequence in an F-group is similar to the orienteering problem (OP) [58]. Both are classic NP-hard problems, for which a variety of exact and heuristic solution methods have been developed. However, in the combined problem addressed in this work, there are also dependencies between the groups. The transitions between two groups contribute to the cost function, and all F-groups share a common job family. This explains why an optimization algorithm has to treat the problem as a whole, meaning that it optimizes all defined groups together.

1.3. State of the art

1.3.1. Start time planning

While the literature contains abundant results regarding scheduling in general, only a few works deal with the explicit determination and optimization of start times. In the foundational work [4] an interval-based temporal logic is described and a concept of reference intervals is introduced. This allows to capture temporal hierarchies which can be utilized for computationally effective reasoning in constraint programming. Examples of the application of this logic to databases, process modeling and systems with real-time updates are given. A method for the computation of start times of activities with stochastic durations is described in [19]. The objective is to start the activities as late as possible, while the due dates of the activities are still met with a certain probability. The sequence of the activities follows directly from the computed start times. The work [26] deals with the processing of the individual components of a product at independent parallel flow shops under stochastic process times. The aims of the optimization are to find an optimal product sequence and to determine the start time of each component, so that all components of a product are completed by a due date with a selected probability. Similar to the current work, Monte Carlo (MC) simulations are used to carry out the computations. In [8], the scheduling of a job shop with no-wait constraints and stochastic process times is described. The sequencing of the jobs includes the determination of the earliest possible start times, which are computed by a recursive timetabling algorithm. Similar to the optimization of start times with stochastic process times, [35] deals with the optimization of timetables of trains under stochastic disturbances of the travel times between stations. In contrast to the process times of many production processes, however, travel times of trains can be specifically controlled within certain limits. The purpose of the stochastic optimization is to determine an optimal travel time buffer and compute robust timetables. The calculation of start times under stochastic parameters is also essential for the so-called economic dispatch problem. This problem involves the determination of optimal start times for power generation units to minimize total generation costs. Typical constraints are plant and transmission line capacities, or restrictions on start and shutdown times. In [5], a multi-stage model predictive control approach is used to tackle this problem considering demands and generation uncertainties as stochastic variables. The problem is formulated using two stages: The initial stage determines optimal control actions based on predicted demand and wind generation, while the subsequent stage refine actions as uncertainties unfold. Restructuring the problem as a finite moving-horizon optimal control problem results in linear growth of computational requirements.

1.3.2. Scheduling and sequencing

Scheduling and sequencing is extensively described in standard works like [11, 13, 21, 49], providing concepts, methods, and results of scheduling theory. The common framework describing scheduling problems by the machining environment, the processing characteristics, and the objective is introduced. Various general problems are addressed with solution methods and outcomes discussed for both deterministic and stochastic process times. In most cases, the emphasis of the described problems is on optimization of the sequence of jobs or tasks. An overview on advanced planning and scheduling systems as well as enterprise resource planning is given in [52]. In addition to standard methods such as mixed integer programming (MIP) or constraint programming (CP), hybrid optimization approaches, in particular mixed linear logic programming, are described. An extensive overview of planning and scheduling methods for different industrial sectors is also given in [59]. The work concentrates especially on modeling of parameter uncertainties in various problems. The work [46] provides a comprehensive review of multi-objective optimization in the context of production planning and scheduling. It discusses the successful application of various evolutionary computation methods in multi-objective production scheduling.

Standard works on metaheuristic optimization methods are [22, 53]. Advanced optimization techniques achieved through combining metaheuristics with other traditional optimization methods, such as local search or large neighborhood search, are described in [14]. The simulated annealing algorithm and its typical applications are described in detail in [37].

A review of methods tailored to flow shop scheduling is given in [24]. The work describes the development of different approaches to solve scheduling problems optimally as well as approximately. Solution methods for the minimization of the makespan in no-wait flow shop scheduling problems are described in [45]. The work proposes a combination of a construction and an improvement heuristic. The results are compared with other heuristic solution methods applied to benchmark problems. An overview of works addressing the job shop scheduling problem, classified according to settings, constraints, and objectives, is given in [2]. In the work [36], four different MIP formulations for the classical job shop scheduling problem are analyzed. For the solution, different optimization tools are compared, and the benefits of multi-threading and parameter tuning are explored.

A survey of methods especially in steel production is provided in [47]. This review shows that many publications in the field cover the standardized hot strip mill production scheduling problem, which can be modeled as a price-collecting TSP [41]. The use of simulated annealing to solve parts of this problem is described in [16]. A comparison of several exact and heuristic algorithms, including simulated annealing, applied to the scheduling of steel production in a multi-line manufacturing plant, is given in [3]. The authors of [27] apply simulated annealing to solve a single machine scheduling problem with setup times between job families. The application of a tabu search algorithm to the scheduling problem of an industrial hot rolling mill, which is modeled as a job shop with no-wait constraints, is described in [9]. The extension of the scheduling method to stochastic process times is described in [10].

The TSP and its variations are described in detail in [25], covering both exact and approximate solution methods. An informative study of the TSP, which also includes local search and simulated annealing approaches, is presented in [1]. The authors of [20] focus on the TSP with profits, presenting various applications as well as heuristic and metaheuristic solution methods. The OP and its variants, along with multiple solution methods tailored to this problem, are addressed in [58]. A survey on the OP is also provided in [23], focusing on well-studied variants as the Team OP, the OP with Time Windows, but including also newer applications, for example the Tourist Trip Design Problem. The application of metaheuristic algorithms for the OP are described in [40]. The work compares ant colony optimization and the tabu search algorithm to other heuristics described in the literature, showing the effectiveness of these two algorithms. Vehicle routing problems (VRPs) are extensively described in the standard work [57]. The work defines different families and variants of VRPs and their applications, and describes a variety of exact as well as approximative (heuristic) solution methods. It also provides benchmark results for many types of VRPs.

To address real world problems which are often complex and can involve stochastic elements in the optimization problem, the combinations of simulations with metaheuristics, so called "simheuristics", are described in [17]. Using this concept, simulations are carried out to determine the characteristics of solutions obtained by the metaheuristic algorithm. To deal with uncertainties, a Monte-Carlo based approach, a discrete event simulation, or an agent-based modeling is proposed to carry out the simulations. A review of simheuristics is given in [30], including examples of their application to problems in manufacturing and production. The work [29] describes the application of the simheuristic approach where simulation is used to estimate and update the probability distributions of uncertain parameters, which are then used within heuristic search methods to find near-optimal solutions. The approach is demonstrated through case studies in routing and scheduling.

Based on the literature review, it can be concluded that the following scientific gaps exist for the problem at the considered steel hot rolling mill: First, a model must be developed that captures the production process at the hot rolling mill with sufficient accuracy, taking into account all specifics and requirements. Based on this model, a procedure to compute optimal start times for individual products must be developed. While this can be straightforward in the case of deterministic process times, a method that could be directly applied to the considered problem with stochastic process times was not found in the literature.

The sequence optimization problem consists of a combination of several Traveling Salesman Problems (TSPs) and Orienteering Problems (OPs). While there is rich literature addressing each of these problems individually, the combination of both in a hybrid optimization problem could not be found. Consequently, existing solvers for individual problems cannot be applied to the current combined problem. As a result, an algorithm tailored to this specific production process must be developed. The improvement of the obtained optimized product sequences can then be evaluated by comparison with manually created production plans.

1.4. Contribution

The start time planning addressed in this work requires an exact consideration of the production process of the individual products along their route. Based on this, the job selection and sequence optimization requires solving a combination of several NP-hard optimization problems. Existing solvers cannot be directly applied to this problem. In the optimization, a large number of requirements and rules regarding the production process and the arrangement of jobs has to be taken into account. The settings and solution methods described in literature cannot be directly applied to the present scheduling problem. Therefore, new methods for the creation of solutions were developed. The main contributions are as follows:

- A model framework was developed which describes the plant and the production process with all its peculiarities with the required accuracy. The outputs of this model contain all quantities needed for the start time planning, product selection, and sequence optimization.
- A method for the start time planning under deterministic as well as stochastic process times was developed.
- All the requirements and rules which must be adhered to during the production of multiple groups of jobs have been systematically listed and assigned priorities from one to four.
- A heuristic algorithm to automatically optimize the job selection and sequence for a defined problem with multiple groups was developed.
- The effectiveness of the optimization algorithm was examined by applying it to real production problems from an industrial multi-line hot rolling mill and comparing it to manually created solutions.

1.5. Content

This work is structured as follows. Chapter 2 describes the developed model of the production process and the plant. The processing model of products and jobs, based on passage times of defined checkpoints, is explained in detail in Section 2.1. Section 2.2 shows a topological model of the plant and describes the possible product routes, the different sections and facilities in detail. The main sections of the plant are described in Section 2.3.

Chapter 3 describes the planning of start times of products and jobs for a given sequence. Section 3.2 defines the optimization problem. In Section 3.3 computational methods for problems with deterministic as well as stochastic process times are described. The application of these methods to a use case are shown in Section 3.4. The actual start time planning for the production process in the plant of BÖHLER is explained in Section 3.5.

Chapter 4 deals with the job selection and sequence optimization for schedules with multiple groups of jobs. Section 4.2 defines the quantities regarding the production process of multiple groups, which are needed for the formulation of the optimization problem described in Section 4.3. The application of exact solution methods for the optimization of individual groups is examined in Section 4.4. These optimizations were carried out with some simplifications of the problem, in particular without the consideration of the furnace assignment to the individual jobs. The developed heuristic algorithm used for the optimization of the full problem without these simplifications is described in Section 4.5. The optimization of a use case with multiple groups of jobs using the heuristic algorithm is demonstrated in Section 4.6. The quality of the obtained solution is evaluated by a comparison with a manually created schedule. Finally, conclusions and an outlook are given in Chapter 5.

Chapter 2.

Model of the production process and the plant

In the standard framework for describing scheduling problems ([49]), a number of resources (machines) are assigned to a number of tasks (jobs). A job usually occupies exactly one machine at a time and the changeover between the machines occurs instantaneously at discrete points in time. The spatial extension of the individual products is not considered in this framework.

In the multi-line hot rolling mill considered in this work, the jobs consist of billets, which travel through production lines while being rolled out to considerable lengths (up to several kilometers in case of wire). Therefore, the considered plant was not modeled as a set of individual machines but as an arrangement of production lines with checkpoints at defined positions. The production process of a product is described by discrete points in time at which the head and tail of the product passes the checkpoints.

The model described in this chapter is used to predict the processing of the next products during ongoing operation of the system, and to simulate the processing of a sequence of products in the sequence optimization (see Chapter 4). This requires a precise consideration of all process steps along the production lines and of technical requirements of the plant (e. g., safety distances between products, product-dependent processing steps such as cropping). For these purposes, the process times in the model described in this chapter are assumed to be deterministic. In contrast, Chapter 3 describes the start time planning for a larger number of products before the production starts. For this purpose, a simpler system model is used while individual process steps are assumed to be stochastic and the corresponding process times are random variables.

2.1. Model of the production process

During the manufacturing process, products of finite length travel through the production lines along individual product routes. During this travel, they pass so-called checkpoints along the lines. A checkpoint is identified by its index k and defined at a certain position. Some checkpoints also consider specific conditions in the processing of a product. For example, the checkpoint located at the position of a reversing mill stand captures the the passing of the product's head before the first pass and the passing of the product's tail after the last pass. The endpoints head and tail end of a product are defined by its the current direction of movement. The head end is at the front, the tail end at the back. When the direction of movement changes, the positions of head and tail are reversed.

2.1.1. Processing of a single product

In the following, a single product (a billet) σ_i out of a sequence of products, given by a vector $\boldsymbol{\sigma}$, is considered. The production process of product σ_i at two consecutively passed checkpoints k and l is outlined in Figure 2.1. The head end of the product passes the checkpoint k at time $t_i^{\mathrm{H},k}$, the tail end at time $t_i^{\mathrm{T},k}$. The times $t_i^{\mathrm{H},k}$ and $t_i^{\mathrm{T},k}$ are referred to as passage times of head and tail, respectively.

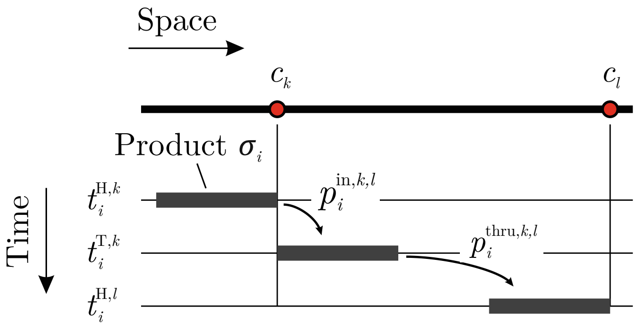

Figure 2.1.: Process and passage times of a product σ_i at two consecutively passed checkpoints k and l.

To calculate the passage times of a product σ_i, developed process time models provide two process times for every pair of consecutively passed checkpoints k and l: The *entry time*

$$p_i^{\mathrm{in},k,l} = t_i^{\mathrm{T},k} - t_i^{\mathrm{H},k} \tag{2.1}$$

is the total timespan required by product σ_i to pass the checkpoint k. The *transit time*

$$p_i^{\mathrm{thru},k,l} = t_i^{\mathrm{H},l} - t_i^{\mathrm{T},k} \tag{2.2}$$

is the timespan required by product σ_i to travel between the checkpoints k and l. Since a production line might branch at a checkpoint, two indices k and l are necessary to uniquely specify the passage times.

During the processing of product σ_i, its head and tail end pass m_i checkpoints. The route of σ_i is defined as sequence $\boldsymbol{r}_i = (r_i^1, \dots, r_i^{m_i})$, where r_i^k is the checkpoint k in the route of σ_i. The timespan required by the product σ_i to pass the last checkpoint $r_i^{m_i}$ is denoted as

$$p_i^{\text{exit}} = t_i^{\text{T},r_i^{m_i}} - t_i^{\text{H},r_i^{m_i}} \tag{2.3}$$

With a given start time t_i^{H,r_i^1}, the passage times at checkpoint r_i^k are

$$t_i^{\text{H},r_i^k} = t_i^{\text{H},r_i^1} + \sum_{j=1}^{k-1} \left(p_i^{\text{in},r_i^j,r_i^{j+1}} + p_i^{\text{thru},r_i^j,r_i^{j+1}} \right) , \qquad k = 1, \dots, m_i \ , \tag{2.4a}$$

$$t_i^{\text{T},r_i^k} = t_i^{\text{H},r_i^1} + \sum_{j=1}^{k} p_i^{\text{in},r_i^j,r_i^{j+1}} + \sum_{j=1}^{k-1} p_i^{\text{thru},r_i^j,r_i^{j+1}} , \qquad k = 1, \dots, m_i - 1 \ , \tag{2.4b}$$

$$t_i^{\text{T},r_i^{m_i}} = t_i^{\text{H},r_i^1} + \sum_{j=1}^{m_i-1} \left(p_i^{\text{in},r_i^j,r_i^{j+1}} + p_i^{\text{thru},r_i^j,r_i^{j+1}} \right) + p_i^{\text{exit}} \ . \tag{2.4c}$$

The processing of σ_i is finished after $t_i^{\text{T},r_i^{m_i}}$.

For all entry times, $p_i^{\text{in},r_i^k,r_i^{k+1}} \geq 0$ holds true. Entry times $p_i^{\text{in},r_i^k,r_i^{k+1}} = 0$ occur for modeling reasons between some checkpoints, e. g., when a product is stopped as its head arrives at checkpoint r_i^k and lifted by a crane. Transit times $p_i^{\text{thru},r_i^k,r_i^{k+1}}$ can be negative if the length of product σ_i exceeds the spatial distance between the checkpoints r_i^k and r_i^{k+1}. In this case, the head of σ_i passes checkpoint r_i^{k+1} before the tail passes checkpoint r_i^k. For the passage times of a product σ_i at two consecutive checkpoints r_i^k and r_i^{k+1}, the following relations must hold.

$$t_i^{\text{H},r_i^k} \leq t_i^{\text{H},r_i^{k+1}} \qquad \forall k = 1, \dots m_i - 1 \tag{2.5a}$$

$$t_i^{\text{T},r_i^k} \leq t_i^{\text{T},r_i^{k+1}} \qquad \forall k = 1, \dots m_i - 1 \tag{2.5b}$$

$$t_i^{\text{H},r_i^k} \leq t_i^{\text{T},r_i^k} \qquad \forall k = 1, \dots m_i \tag{2.5c}$$

2.1.2. Processing of multiple products

To consider (intentional) delays between products in the simulation of a given sequence of products, a concept of checkpoint blocking and release is used. A checkpoint blocking rule related to product σ_i defines a duration $p_i^{\text{blk},k,l} \geq 0$, where k is a checkpoint in the route of σ_i and l is an arbitrary checkpoint to be blocked (which does not necessarily have to be included in the route of σ_i). After the tail of σ_i has passed the checkpoint k, checkpoint l is blocked until its release time

$$t_i^{\mathrm{R},l} = \max_{k \in \boldsymbol{r}_i} \left\{ t_i^{\mathrm{T},k} + p_i^{\mathrm{blk},k,l} \right\} . \tag{2.6}$$

Because products cannot overtake each other, every checkpoint $k \in \boldsymbol{r}_i$ is blocked at least until $t_i^{\mathrm{T},k}$ using a blocking duration $p_i^{\mathrm{blk},k,k} \geq 0$. Moreover, checkpoint blockings are used to model the following details of the production process:

- Required minimum distances between products
- Temporal separation between products in the WBF because of different required product temperatures (see Section 2.3.1)
- Blocking of the profile rolling line due to a change of the rolling speed or retooling (see Section 2.3.7)

In the process simulation, the release times of all existing checkpoints are updated after every product. The release times of all checkpoints which are affected by blocking after product σ_i are updated according to (2.6). The release times of all checkpoints $k \in U$ which are not affected by any checkpoint blocking rule after product σ_i remain unchanged, i. e.,

$$t_i^{\mathrm{R},k} = t_{i-1}^{\mathrm{R},k} \quad \forall k \in U \ , \tag{2.7}$$

where U is the set of indices of unaffected checkpoints. For the next product σ_{i+1}

$$t_{i+1}^{\mathrm{H},k} \geq t_i^{\mathrm{R},k} \quad \forall k \in \boldsymbol{r}_{i+1} \tag{2.8}$$

must hold.

The concept of checkpoint blocking is illustrated using an example of a plant with 5 checkpoints outlined in Figure 2.2. The processing of two products σ_1 followed by σ_2 in this plant is shown in a Gantt-like chart in Figure 2.3. The route of σ_1 is given by the checkpoint sequence $\boldsymbol{r}_1 = (1, 2, 4, 5)$, the route of σ_2 by $\boldsymbol{r}_2 = (1, 2, 3)$. The bars in Figure 2.3 represent the entry times $p_i^{\mathrm{in},k,l}$, during which checkpoints are physically occupied by each product σ_i from $t_i^{\mathrm{H},k}$ to $t_i^{\mathrm{T},k}$.

Product σ_1 is started at $t_1^{\mathrm{H},1}$ and passes checkpoint 1 during the entry time $p_1^{\mathrm{in},1,2}$. As the transit time is $p_1^{\mathrm{thru},1,2} < 0$, the head of σ_1 passes checkpoint 2 at $t_1^{\mathrm{H},2}$ before the tail passes checkpoint 1 at $t_1^{\mathrm{T},1}$. The transit time to the next checkpoint 4 is $p_1^{\mathrm{thru},2,4} > 0$. The processing of σ_1 is finished at $t_1^{\mathrm{T},5}$ (not labeled in Figure 2.3).

After passed by the tail of σ_1, checkpoint 1 is blocked for the period $p_1^{\mathrm{blk},1,1} > 0$ until the release time $t_1^{\mathrm{R},1}$. After checkpoint 4 is passed by the tail of σ_1, checkpoint 3, which is not in the route of σ_1, is blocked for the duration $p_1^{\mathrm{blk},4,3} > 0$

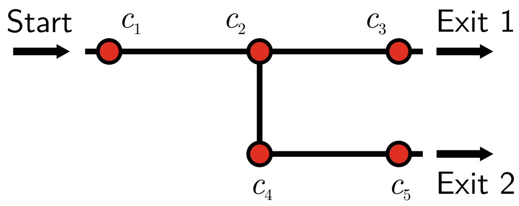

Figure 2.2.: Example of a plant with 5 checkpoints.

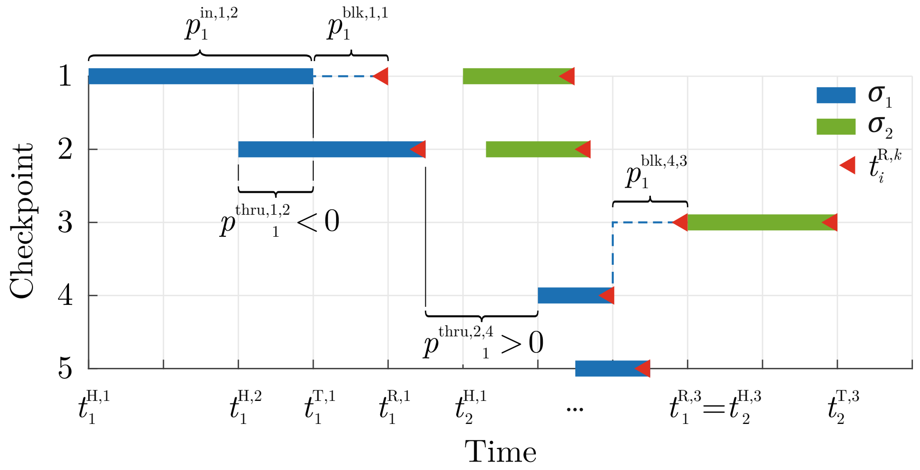

Figure 2.3.: Gantt chart illustrating the processing of two products σ_1 and σ_2.

until the release time $t_1^{\mathrm{R},3}$. The possibility of blocking checkpoints away from the product route is necessary to consider technical requirements in the process simulation. The checkpoints 2, 4, and 5 are released right when the tail of σ_1 passes them. With given process times $p_2^{\mathrm{in},k,l}$ and $p_2^{\mathrm{thru},k,l}$, the earliest possible start date $t_2^{\mathrm{H},1}$ of product σ_2 is limited by the release time $t_1^{\mathrm{R},3} = t_2^{\mathrm{H},3}$. Hence, checkpoint 3 is the bottleneck in the processing of σ_1 and σ_2.

Assuming that all $p_2^{\mathrm{blk},k,l} = 0$, the release dates of the checkpoints after σ_2 are

$$\begin{aligned} t_2^{\mathrm{R},k} &= t_2^{\mathrm{T},k} \;, \qquad k = 1,2,3 \;, \\ t_2^{\mathrm{R},k} &= t_1^{\mathrm{R},k} \;, \qquad k = 4,5 \;. \end{aligned}$$

2.2. Topological model of the plant

The model of the steel hot rolling mill, consisting of production lines and defined checkpoints, is shown in Figure 2.4. For the entry of the products into the manufacturing process, there are two possible start points:

Start 1 at the last position in the WBF. This implies that products heated in the WBF enter the manufacturing process in a heated condition.

Start 2 at a table where product heated in the INDU are positioned before they are picked up by a crane and transported to the INDU. Products at Start 2 enter the system in a cold condition.

After heating, a product is rolled out to an intermediate cross-section at the reversing rolling stand before being fed to either the flat or the profile rolling line for the final processing. The product is referred to as primary material (PM) before the reversing rolling stand, as intermediate product (IM) after the reversing rolling stand, and as a finished product (FP) at the end of the line. The associated product cross-sectional areas are denoted as A_{pm}, A_{im}, and A_{fp}, and the product lengths as l_{pm}, l_{im}, and l_{fp}.

In the model, products can exit the plant at the following positions:

Exit 1 at the end of the flat rolling line.

Exit 2 after stand 4 of the profile rolling line.

Exit 3 after stand 6 of the profile rolling line.

Exit 4 after stand 7 of the profile rolling line. After stand 7, the profile rolling line contains additional stands which are not included in the model.

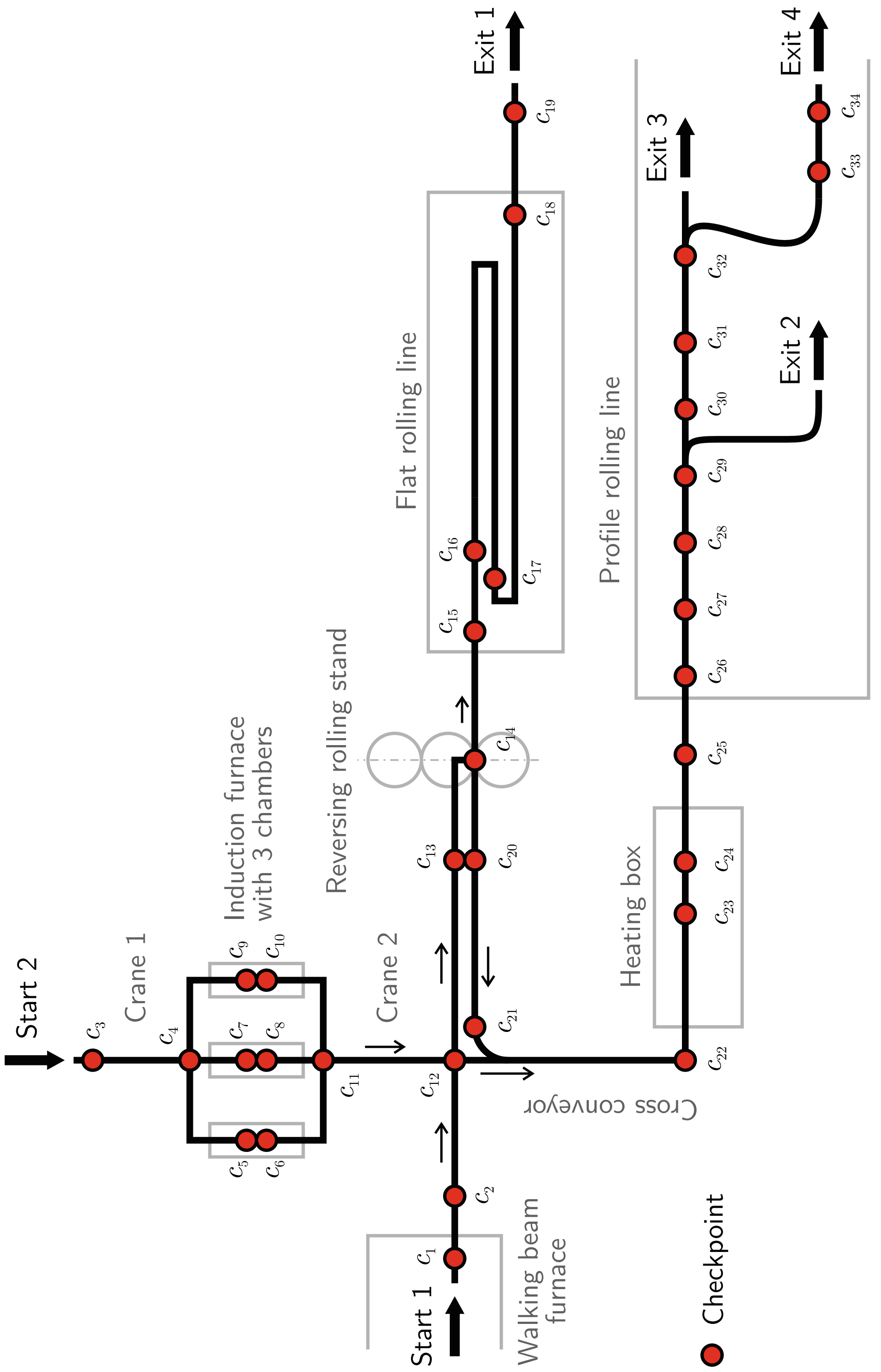

Figure 2.4.: Model of the plant.

The first part of a product route depends on the used furnace. The second part depends on the line at which the product is finished. If the product is finished at the profile rolling line, the route furthermore depends on whether the product is processed at the reversing rolling stand and on the last used stand at the profile rolling line. The possible checkpoint sequences are as follows:

1. Used furnace
    - WBF: 1, 2,
    - INDU
        - Chamber 1: 3, 4, 5, 6, 11,
        - Chamber 2: 3, 4, 7, 8, 11,
        - Chamber 3: 3, 4, 9, 10, 11,
2. Finishing line
    - Flat: 12, 13, ..., 19 → Exit 1 (products always pass the reversing rolling stand).
    - Profile
        a) Reversing rolling stand
            - Without: 12, 22,
            - With: 12, 13, 14, 20, 21, 22,

        b) Last stand of profile rolling line
            - Stand 4: 23, 24, ..., 29 → Exit 2
            - Stand 6: 23, 24, ..., 32 → Exit 3
            - Stand 7: 23, 24, ..., 34 → Exit 4

For example, the complete route of a product σ_i heated in the WBF, processed at the reversing rolling stand and afterwards at the stands 1–6 of the profile rolling line is defined by the checkpoint sequence

$$\boldsymbol{r}_i = (1, 2, 12, 13, 14, 20, 21, \dots 32) \ .$$

The checkpoints are passed in ascending order and checkpoint 12 is passed by all products.

2.3. Main sections of the plant

In the following, a overview of the main sections of the plant is given. Equations to calculate the process times $p_i^{\text{in},k,l}$ and $p_i^{\text{thru},k,l}$ for all existing sections between consecutive checkpoints k and l (i. e., that can follow one another in a product route) are given in Appendix A.

2.3.1. Walking beam furnace

The WBF is a gas-fired continuous furnace in which the majority of the products are heated. There is a larger number of positions inside the furnace. At each stroke, the products move forward by one position and the product at the last position is discharged. A new product is fed into the furnace at the first position. Figure 2.5 illustrates the arrangement of the products in the WBF and the relevant checkpoints at its exit.

All products heated in the WBF have to attain a temperature within a product specific prescribed temperature interval at the exit of the WBF. If the temperature intervals of two consecutive products differ too much, extra time is needed after the first product. This extra time allows the furnace temperature controller to change the temperature level and the subsequent product to attain the new temperature. The needed extra time (without a product being discharged) is calculated depending on the difference of the prescribed temperature intervals of two consecutive products.

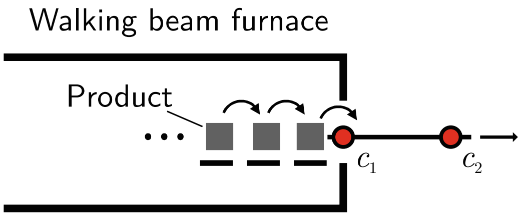

Figure 2.5.: Products and checkpoints at the WBF.

2.3.2. Induction furnace

The INDU is a batch-type furnace with three (identical) chambers. In each of these chambers, one product can be heated at a time. The chambers are loaded always in the order $1, 2, 3, 1, \ldots$. The products are transported to and from the chambers with two different cranes. The transport times of the two cranes limit the throughput of the INDU, modeled by blocking of the checkpoints 3 and 11. If more than three consecutive products are heated in the INDU, the throughput is further reduced due to the long INDU heating time. The total throughput capacity of the INDU is significantly lower than that of the WBF. Hence, the INDU is used as an auxiliary heating device and for a maximum of three consecutively processed products to avoid delays.

2.3.3. Reversing rolling stand

The reversing rolling mill stand consists of a three-high hot rolling mill with tilting tables on both sides, outlined in Figure 2.6. The product is transported to the mill stand and rolled out in n passes to an intermediate product with square cross section. After each pass $1, \ldots, n-1$, the product is transported from the mill stand to a tilting table. The tilting table is raised or lowered, such that the product can be fed into the mill stand again without changing the direction of rotation of the rolls.

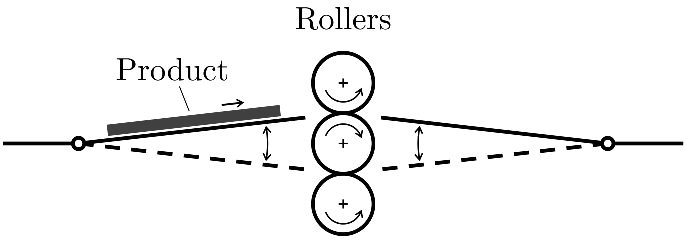

Figure 2.6.: Three-high reversing rolling mill with tilting tables on both sides.

Because the reversing rolling stand produces a square cross-section, the product is rotated 90° around its longitudinal axis when it is located at the tilting table. A closer analysis shows that the time $p_{\mathrm{T,rev}}$ between two consecutive rolling passes exhibits some fluctuations. These occur mainly due to the manual motion control inputs between two rolling passes. Other factors of influence are for example slip between the product and the roller table. Therefore, for the start time planning under stochastic processing times (use case in Section 3.4.2), the total process time at the reversing rolling stand is treated as a random variable. For the simulations carried out in the sequence optimization as described in Section 4.5, the reversing time $p_{\mathrm{T,rev}}$ is treated as deterministic value.

After being processed at the reversing rolling stand, the intermediate product is transported either to the flat rolling block (n is odd) or to the profile rolling line (n is even).

To have some operational flexibility, the roller table between the cross conveyor and the reversing rolling stand consists of two parallel tracks that can be individually operated (see Figure 2.4). These two tracks are represented by the sections

c_{12}–c_{13} and c_{20}–c_{21}, respectively. They facilitate the simultaneous transportation of two products in opposite directions.

2.3.4. Flat rolling line

The flat rolling line consists of a tandem rolling mill with five mill stands. The intermediate products are reversingly rolled out in three passes to a rectangular cross-section. In contrast to the reversing rolling stand, the products are not rotated between two passes and there are no manual manipulations in the rolling process. However, observations reveal random fluctuations of the in the reversing time $p_{\mathrm{F,rev}}$ between two passes. These fluctuations can occur for example because the products are sliding slightly across the roller table during the reversing process. Due to the lack of manual manipulations, the fluctuations are less significant than those of the reversing time $p_{\mathrm{T,rev}}$ at the reversing rolling stand. Therefore, the fluctuations of $p_{\mathrm{F,rev}}$ are neglected and a deterministic value is used. The finished products are then transported to the exit 1.

2.3.5. Cross conveyor

The cross conveyor transports products from the roller table in front of the reversing rolling stand to the roller table in front of the heating box. When a product is not processed at the reversing rolling stand, it is picked up by the cross conveyer as soon as the head reaches checkpoint 12. If a product was processed at the reversing rolling stand, it is ejected from the reversing rolling stand and transported in the direction of the WBF until the product tail reaches checkpoint 21. In both cases, the product is then transported in the transverse direction to the roller table before the heating box.

2.3.6. Heating box

The heating box is used to reheat products or to keep their temperature constant before they are processed at the profile rolling line. The model of the heating box consists of the checkpoints 22 to 25. The checkpoints 23 and 24 are located at the positions of two probes, which are used to control the position of products while they stay in the heating box. Before the exit of the heating box (checkpoint 25) a shear is located, which can be used to crop the head or the tail or both.

2.3.7. Profile rolling line

The modeled part of the profile rolling line consists of the first six stands (checkpoints 26–31), a looper (between checkpoints 32 and 33), and the next mill stand 7 (at checkpoint 34). Products can leave the profile rolling line after stand 4 (checkpoint 29 → exit 2), after the shear at checkpoint 32 (exit 3), or after a later part of the profile rolling line. In the latter case, only the processing in stand 7 is considered, after which the product leaves the modeled part of the plant at exit 4.

At the profile rolling line, the rolling speeds of all stands are synchronized, although the looper allows a separation of speeds between the first six and the remaining mill stands. The rolling speed at each stand depends on a prescribed, product-specific final rolling speed. If the final rolling speeds of two consecutive products are unequal, the whole profile rolling line must be emptied and the mill stand speeds must be changed before the next product can enter the profile rolling line. This is modeled by blocking checkpoint 25 for a specific time span. Also, the retooling of the profile line after a P-group i is modeled by the blocking of checkpoint 25 for an appropriate retooling duration. This blocking starts after the last product of the group i has left the profile rolling line.

Chapter 3.

Start time planning

Parts of this chapter were published in similar form in the author's previous publication [33].

This chapter is concerned with the calculation of start times of products processed in the multi-line steel hot rolling mill where the product routes and sequence of products are given. Following the problem description, two different computational methods of solving the problem in the case of stochastic process times are described. The application of these methods is shown in a use case. Finally, the start time planning for the considered multi-line steel hot rolling mill is described.

3.1. Introduction

If the product sequence is already given, it is generally necessary to specify the start times of the individual processing steps. This is also true for shops with blocking and no-wait constraints.

In the planning task considered in this chapter, logistical reasons require that the start times of all products of a production batch have to be set before the production starts. In a typical flow shop setting, each machine processes the products in the same order. A product cannot move to the next machine until it was completed by the current one, and machines start processing a product only when the previous product is completed and moved on. Due to this sequential processing, there are no complexities of overlapping or interrupted tasks. If the product sequence is given and the process times are deterministic, the entry and exit times at each machine can be calculated without taking uncertainties into account. In this setting with deterministic processing times, the calculation of the earliest possible start times is straightforward. However, it is a complex problem if the process times are stochastic, because these result in stochastic entry and exit times at the machines and thus lead to uncertainties and possible delays in the processing of each product. If a product is started too early, it can interfere with the previous product. If this must be strictly avoided, the start times have to be planned with the maximum possible task durations. The planning can then be done as in the case of deterministic process times but results in a large makespan. In contrast, if conflicts can be tolerated to some extent, earlier start times can be chosen to shorten the makespan.

In Chapter 2, a detailed model of the plant and the production process was presented. In this model, the finite product lengths and the lengths of the sections of the plant are taken into account. A product can occupy several plant sections at the same time. To calculate the process flow of a product, a high number of process times are necessary for each product. This detailed model is used for the production planning in the practical production, where typically only the start times of the next few products are computed in regular time intervals. For this task it showed in practice that considering the process times as deterministic values leads to good results. In contrast, for the start time planning described in this chapter, the start times of a larger number of products are calculated in advance. For this task, some process times are considered random variables, because their fluctuations cannot be ignored for the estimation of optimal start times of a large number of products in advance. To reduce the complexity of the planning task and the estimation of the stochastic process times, the system model is simplified. The plant is modeled as flow shop with blocking and no-wait constraints, discrete machines and discrete products in the Sections 3.2 to 3.4 of this chapter.

In the flow shop, if a product cannot be forwarded to the next machine immediately after the current production step finishes, the no-wait constraint is violated. This violation leads to an unplanned waiting time of the product and a blocking of the current machine. The product that violates the no-wait constraint is referred to as conflicted. Unplanned waiting times lead to an undesired cooling of the conflicted product. Such products can be processed further but may require additional post processing steps. These cause additional costs per conflicted product.

For the considered industrial plant, stochastic process times are analyzed based on data recorded by the existing material tracking system. Using the maximum possible values of the recorded stochastic process times for start time planning is a conservative strategy and would lead to a large makespan. As an alternative, the start times of the products are optimally calculated by taking into account the empirical probability distributions of the process times. In fact, a cost function is designed and minimized to balance the costs of the resulting expected makespan and the costs of the expected number of conflicted products. Finally, the results that take into account the probability distribution functions are compared with the results of simpler methods, such as using the expected values or percentile values of the random process times.

3.2. Problem statement

3.2.1. System description

Consider a flow shop with m machines arranged in series. They process n products in a given order. Each product σ_i, $i = 1, \dots, n$ successively moves through every machine $k = 1, \dots, m$ (same route for each product).

Let P_i^k be the random process time of product σ_i at the machine k. Capital letters are used for process times and time instances that are random variables. It is assumed that the random variables P_i^k are independent from each other and that they have the known cumulative distribution functions (CDFs) $F_{P_i^k}$ and probability density functions (PDFs) $f_{P_i^k}$. The values P_i^k are restricted to the interval $\left[p_i^{\min,k}, p_i^{\max,k}\right]$.

The time when a product σ_i enters machine k is a random variable denoted by T_i^k. Its realization is t_i^k. A machine can hold only one product at a time, which implies the same sequence of products at each machine. The precedence between consecutive products σ_{i-1} and σ_i is represented by

$$t_i^k \geq t_{i-1}^k \quad \forall k = 1, \dots, m \; . \tag{3.1}$$

When the processing of σ_i at machine k is finished, the product should immediately enter the next machine $k+1$ at time t_i^{k+1}. However, if machine $k+1$ is still occupied by the previous product σ_{i-1}, product σ_i cannot leave machine k. In this case, the entry of σ_i at machine $k+1$ is delayed until machine $k+1$ releases σ_{i-1}, i. e., $t_i^{k+1} = t_{i-1}^{k+2}$. The resulting conflict is attributed to product σ_i. The blocking and no-wait constraints for the processing of consecutive products σ_{i-1} and σ_i are represented by

$$t_i^1 \geq t_{i-1}^2 \; , \tag{3.2a}$$

$$t_i^{k+1} = \max\left(t_i^k + p_i^k \, , t_{i-1}^{k+2}\right) \quad \forall k = 1, \dots m-2 \; , \tag{3.2b}$$

$$t_i^m \geq t_{i-1}^m + p_{i-1}^m \; . \tag{3.2c}$$

This work distinguishes between conflict-free processing (at least up to a certain machine) and actual processing with possible conflicts. A hat ˆ indicates entry times in the case of conflict-free processing. Hence, $\hat{t}_i^k$ or $\hat{T}_i^k$ is the entry time in case of conflict-free processing of product σ_i at least until it enters machine k. The actual entry times (with possible conflicts) are denoted by t_i^k or T_i^k.

3.2.2. Optimization problem

The start times of all n products have to be chosen before the production of the batch starts. The expected makespan and the expected number of conflicted products depend on the chosen start times. Conflicted products require additional inspection and post processing and therefore cause additional costs. These costs arise once per conflicted product, even if more than one conflict are associated with the respective product.

At the beginning of a production lot, all machines are empty. Without loss of generality, product σ_1 starts at $t_1^1 = 0$. The makespan is then the end time of product n which is referred to as T_n^{m+1}. The number of conflicted products is denoted by N_c. The planning task is to choose the start times $\hat{t}_i^1$, $i = 2, \ldots, n$ of the products σ_2 to σ_n so that a specific cost function J is minimized. In general, a conflict attributed to the product σ_i, $i > 1$ is also possible at the first machine if $\hat{t}_i^1 < t_{i-1}^2$. Therefore, the scheduled start time $\hat{t}_i^1$ is referred to as start time in the conflict-free case and might not be met.

As T_n^{m+1} and N_c are random variables, their expected values $\mathrm{E}(T_n^{m+1})$ and $\mathrm{E}(N_\mathrm{c})$ are considered in the cost function J. It is designed in the form

$$J = w_\mathrm{t}\,\mathrm{E}\Big(T_n^{m+1}\Big) + w_\mathrm{c}\,\mathrm{E}(N_\mathrm{c}) \ , \tag{3.3}$$

with the weighting factors w_t and w_c. The start times $\hat{t}_2^1, \ldots, \hat{t}_n^1$ are the solution of the optimization problem

$$\min_{\hat{t}_2^1,\ldots,\hat{t}_n^1} J\Big(\mathrm{E}\Big(T_n^{m+1}\Big(\hat{t}_2^1, \ldots, \hat{t}_n^1\Big)\Big), \mathrm{E}\Big(N_\mathrm{c}\Big(\hat{t}_2^1, \ldots, \hat{t}_n^1\Big)\Big)\Big) \tag{3.4a}$$

$$\text{s.t.} \quad \hat{t}_2^1 \geq 0 \ , \tag{3.4b}$$

$$\hat{t}_i^1 \geq \hat{t}_{i-1}^1 \quad \forall i = 3, \ldots, n \ , \tag{3.4c}$$

$$\text{(3.1)–(3.2)}. \tag{3.4d}$$

Larger values of n make this optimization problem is difficult to solve. It can can be simplified as follows: Depending on the scheduled start time $\hat{t}_i^1$, a product σ_i is processed without a conflict with a certain probability $\gamma_i\Big(\hat{t}_i^1\Big)$. A probability value γ_set is specified, and it is required that each product σ_i, $i = 2, \ldots, n$ is processed without conflicts with the probability $\gamma_i \geq \gamma_\mathrm{set}$. Based on this requirement, the scheduled start times $\hat{t}_i^1$ can be determined as the earliest possible start times that fulfill

$$\gamma_\mathrm{set} \leq \gamma_i\Big(\hat{t}_i^1\Big) \quad \forall i = 2, \ldots, n \ . \tag{3.5}$$

Delays of the product σ_i affect only the entry times of the products after σ_i. Thus, start times can be scheduled product by product. This strategy reduces the number of decision variables from $n-1$ to the single probability value γ_{set} and the optimization problem can be written as

$$\min_{\gamma_{\text{set}}} J\Big(\mathrm{E}\Big(T_n^{m+1}\Big(\hat{t}_2^1,\dots,\hat{t}_n^1\Big), \mathrm{E}\Big(N_{\mathrm{c}}\Big(\hat{t}_2^1,\dots,\hat{t}_n^1\Big)\Big)\Big)\Big) \tag{3.6a}$$

$$\text{s.t.} \quad \hat{t}_i^1 = \min_{\tilde{t}_i^1} \tilde{t}_i^1 \tag{3.6b}$$

$$\text{s.t.} \quad \gamma_{\text{set}} \leq \gamma_i\Big(\tilde{t}_i^1\Big) \quad \forall i = 2,\dots,n \;. \tag{3.6c}$$

Solutions $\hat{t}_i^1$ from (3.6) may be sub-optimal in terms of (3.4). However, they are well suited for practical application as will be shown in Section 3.4. Methods to determine the scheduled start times $\hat{t}_2^1,\dots,\hat{t}_n^1$ based on γ_{set} are described in the following section.

3.3. Computational methods

In the following, the determination of the scheduled start time and the subsequent entry times of a sequence of products is described. Section 3.3.1 describes the start time planning of products with strictly deterministic process times. Then Section 3.3.2 gives an analytical solution for the start time planning of products with stochastic process times. Since the equations derived in Section 3.3.2 can be numerically solved only for small shops with few machines, a solution for larger shops based on Monte Carlo (MC) simulations is presented in Section 3.3.3.

3.3.1. Deterministic process times

For the case of deterministic process times p_i^k, the earliest possible start times $\hat{t}_i^1$ that avoid conflicts are computed. Product σ_1 starts at the time $t_1^1 = \hat{t}_1^1 = 0$, where all machines are assumed to be empty. The remaining entry times of the product σ_1 are

$$t_1^{k+1} = \hat{t}_1^{k+1} = \sum_{j=1}^{k} p_1^j \;, \quad k = 1,\dots,m \;. \tag{3.7}$$

To calculate the earliest possible start date of a product σ_i, $i > 1$, its entry times without the consideration of conflicts are first calculated for a start time 0, denoted as $t_i^{0,k}$. In this case, the entry times of σ_i would be

$$t_i^{0,1} = 0 \ , \tag{3.8a}$$

$$t_i^{0,k+1} = \sum_{j=1}^{k} p_i^j \ , \quad k = 1, \ldots, m \ . \tag{3.8b}$$

The earliest possible start time of σ_i that avoids conflicts with the previous product σ_{i-1} is then obtained by postponing all times $t_i^{0,1}, \ldots t_i^{0,m+1}$ by the same time span. This time span d_i is chosen such that σ_i enters at least one machine exactly when the previous product σ_{i-1} leaves this machine,

$$d_i = \max_{k=1,\ldots,m} \left(t_{i-1}^{k+1} - t_i^{0,k} \right) \ . \tag{3.9}$$

Figure 3.1 illustrates the method for two consecutive products σ_{i-1} and σ_i. The resulting entry times of σ_i are

$$t_i^1 = \hat{t}_i^1 = d_i \tag{3.10}$$

$$t_i^{k+1} = \hat{t}_i^{k+1} = \hat{t}_i^1 + \sum_{j=1}^{k} p_i^j \ . \tag{3.11}$$

With the entry times t_i^k, the start time planning of the remaining products σ_{i+1} to σ_n can be carried out product by product in the same way.

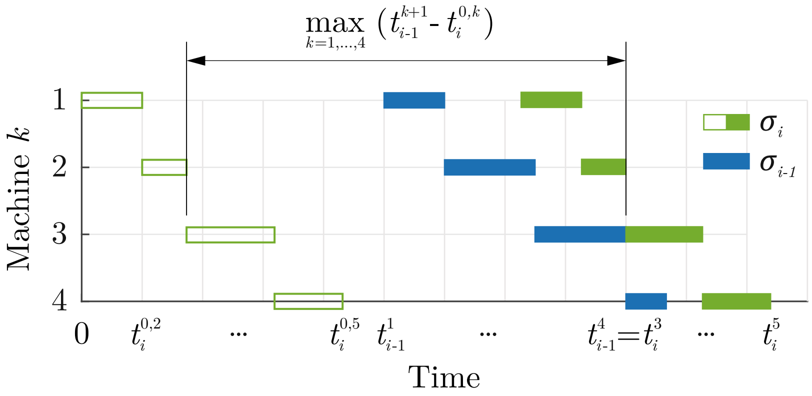

Figure 3.1.: Determination of the entry times t_i^k based on the entry times t_{i-1}^k with deterministic process times for an example of $m = 4$ machines.

In the case of stochastic process times P_i^k, the entry times T_i^k are also random variables. The satisfaction of relations like $T_i^k \geq T_{i-1}^{k+1}$ can in general be specified only with a certain probability. For this reason, the method described in this section cannot be directly applied. However, the start times can still be scheduled product by product. This is presented in the following section using the probability γ_{set} of conflict free processing.

3.3.2. Stochastic process times, analytical solution

A probability value $\gamma_{\text{set}} < 1$ can lead to conflicts and consequently to delays in the production process. The stay of product σ_i at machine k is $T_i^{k+1} - T_i^k$ and consists of the stochastic process time P_i^k plus a possibly non-zero waiting time until the machine $k+1$ becomes free. In general, the scheduled start times $\hat{t}_i^1$ cannot be exactly realized and the actual start times as well as all entry times T_i^k are random variables.

Joint cumulative distribution function of the entry times of product σ_1

In the following, $H(t)$ denotes the Heaviside function, with $H(t) = 1$ for $t \geq 0$ and $H(t) = 0$ for $t < 0$. The probability of an event A is written as $\Pr(A)$. The start time of product σ_1 is

$$t_1^1 = \hat{t}_1^1 = 0 \, . \tag{3.12}$$

The remaining stochastic entry times are

$$T_1^{k+1} = \hat{T}_1^{k+1} = \sum_{j=1}^{k} P_1^j \, , \quad j = 1, \ldots, m \, . \tag{3.13}$$

P_1^j are the independent stochastic process times with the PDFs $f_{P_1^j}$. The CDF of the entry times can be computed as

$$\begin{aligned}
&F_{T_1^2 \cdots T_1^{m+1}}\left(t_1^2, \ldots, t_1^{m+1}\right) = \Pr\left(T_1^2 \leq t_1^2 \wedge \cdots \wedge T_1^{m+1} \leq t_1^{m+1}\right) \\
&\quad = \Pr\left(P_1^1 \leq t_1^2 \wedge \cdots \wedge \sum_{j=1}^{m} P_1^j \leq t_1^{m+1}\right) \\
&\quad = \int_{p_1^{\min,1}}^{p_1^{\max,1}} \cdots \int_{p_1^{\min,m}}^{p_1^{\max,m}} \left[\Pr\left(p_1^1 \leq t_1^2 \wedge \cdots \wedge \sum_{j=1}^{m} p_1^j \leq t_1^{m+1}\right)\right. \\
&\qquad \left. \cdot \prod_{j=1}^{m} f_{P_1^j}\left(p_1^j\right)\right] \mathrm{d}p_1^m \cdots \mathrm{d}p_1^1 \\
&\quad = \int_{p_1^{\min,1}}^{p_1^{\max,1}} \cdots \int_{p_1^{\min,m}}^{p_1^{\max,m}} \left[H\left(\min\left(t_1^2 - p_1^1, \ldots, t_1^{m+1} - \sum_{j=1}^{m} p_1^j\right)\right)\right. \\
&\qquad \left. \cdot \prod_{j=1}^{m} f_{P_1^j}\left(p_1^j\right)\right] \mathrm{d}p_1^m \cdots \mathrm{d}p_1^1 \, .
\end{aligned} \tag{3.14}$$

Scheduled start time of product σ_2

If product σ_2 starts at $\hat{t}_2^1$ and is processed without a conflict, its stochastic entry times are

$$\hat{T}_2^{k+1} = \hat{t}_2^1 + \sum_{j=1}^{k} P_2^j \ , \quad j = 1, \ldots, m \ . \tag{3.15}$$

The probability that product 2 is indeed processed without a conflict is

$$\begin{aligned}
\gamma_2\left(\hat{t}_2^1\right) &= \Pr\left(T_1^2 \leq \hat{t}_2^1 \wedge T_1^3 \leq \hat{T}_2^2 \wedge \ldots \wedge T_1^{m+1} \leq \hat{T}_2^m\right) \\
&= \Pr\left(T_1^2 \leq \hat{t}_2^1 \wedge T_1^3 \leq \hat{t}_2^1 + P_2^1 \wedge \ldots \wedge T_1^{m+1} \leq \hat{t}_2^1 + \sum_{j=1}^{m-1} P_2^j\right) \\
&= \int_{p_2^{\min,1}}^{p_2^{\max,1}} \cdots \int_{p_2^{\min,m-1}}^{p_2^{\max,m-1}} \left[\Pr\left(T_1^2 \leq \hat{t}_2^1 \wedge T_1^3 \leq \hat{t}_2^1 + p_2^1 \wedge \ldots \wedge T_1^{m+1} \leq \hat{t}_2^1 + \sum_{j=1}^{m-1} p_2^j\right)\right. \\
&\qquad \left. \cdot \prod_{j=1}^{m-1} f_{P_2^j}\left(p_2^j\right)\right] \mathrm{d}p_2^{m-1} \cdots \mathrm{d}p_2^1 \\
&= \int_{p_2^{\min,1}}^{p_2^{\max,1}} \cdots \int_{p_2^{\min,m-1}}^{p_2^{\max,m-1}} \left[F_{T_1^2 \cdots T_1^{m+1}}\left(\hat{t}_2^1, \hat{t}_2^1 + p_2^1, \ldots, \hat{t}_2^1 + \sum_{j=1}^{m-1} p_2^j\right)\right. \\
&\qquad \left. \cdot \prod_{j=1}^{m-1} f_{P_2^j}\left(p_2^j\right)\right] \mathrm{d}p_2^{m-1} \cdots \mathrm{d}p_2^1 \ .
\end{aligned} \tag{3.16}$$

With a chosen probability γ_{set}, $\hat{t}_2^1$ can be computed from (3.16) using (3.6b) and (3.6c).

Joint cumulative distribution function of the entry times of product σ_2

The actual entry times of product σ_2 depend on possible conflicts with σ_1 and are random variables in the form

$$T_2^1 = \max\left(\hat{t}_2^1, T_1^2\right) \ , \tag{3.17a}$$

$$T_2^2 = \max\left(T_2^1 + P_2^1, T_1^3\right) \ , \tag{3.17b}$$

$$\vdots$$

$$T_2^m = \max\left(T_2^{m-1} + P_2^{m-1}, T_1^{m+1}\right) \ , \tag{3.17c}$$

$$T_2^{m+1} = T_2^m + P_2^m \ . \tag{3.17d}$$

Their joint CDF is obtained as

$$
\begin{aligned}
F_{T_2^2\cdots T_2^{m+1}}\left(t_2^2,\dots,t_2^{m+1}\right) &= \Pr\left(T_2^2 \le t_2^2 \wedge T_2^3 \le t_2^3 \wedge \cdots \wedge T_2^{m+1} \le t_2^{m+1}\right) \\
&= \Pr\Big(\max\Big(\max\left(\hat{t}_2^1,\ T_1^2\right) + P_2^1,\ T_1^3\Big) \le t_2^2 \\
&\quad \wedge \max\Big(\max\Big(\max\left(\hat{t}_2^1,\ T_1^2\right) + P_2^1,\ T_1^3\Big) + P_2^2,\ T_1^4\Big) \le t_2^3 \\
&\quad \wedge \cdots \\
&\quad \wedge \max\Big(\max(\cdots) + P_2^{m-1},\ T_1^{m+1}\Big) \le t_2^m \\
&\quad \wedge \max\Big(\max(\cdots) + P_2^{m-1},\ T_1^{m+1}\Big) + P_2^m \le t_2^{m+1}\Big)\ .
\end{aligned}
\tag{3.18}
$$

As an example for the case $m = 3$, (3.18) reads as

$$
\begin{aligned}
&F_{T_2^2T_2^3T_2^4}\left(t_2^2,t_2^3,t_2^4\right) = \Pr\Big(\max\Big(\max\left(\hat{t}_2^1,T_1^2\right) + P_2^1, T_1^3\Big) \le t_2^2 \\
&\quad \wedge \max\Big(\max\Big(\max\left(\hat{t}_2^1,T_1^2\right) + P_2^1, T_1^3\Big) + P_2^2, T_1^4\Big) \le t_2^3 \\
&\qquad \wedge \max\Big(\max\Big(\max\left(\hat{t}_2^1,\ T_1^2\right) + P_2^1,\ T_1^3\Big) + P_2^2,\ T_1^4\Big) + P_2^3 \le t_2^4\Big) \\
&= \Pr\Big(\hat{t}_2^1 + P_2^1 \le t_2^2 \wedge T_1^2 + P_2^1 \le t_2^2 \wedge T_1^3 \le t_2^2 \\
&\quad \wedge \hat{t}_2^1 + P_2^1 + P_2^2 \le t_2^3 \wedge T_1^2 + P_2^1 + P_2^2 \le t_2^3 \wedge T_1^3 + P_2^2 \le t_2^3 \wedge T_1^4 \le t_2^3 \\
&\quad \wedge \hat{t}_2^1 + P_2^1 + P_2^2 + P_2^3 \le t_2^4 \wedge T_1^2 + P_2^1 + P_2^2 + P_2^3 \le t_2^4 \wedge T_1^3 + P_2^2 + P_2^3 \le t_2^4 \\
&\qquad \wedge T_1^4 + P_2^3 \le t_2^4\Big) \\
&= \int_{p_2^{\min,1}}^{p_2^{\max,1}} \int_{p_2^{\min,2}}^{p_2^{\max,2}} \int_{p_2^{\min,3}}^{p_2^{\max,3}} \Big[\Pr\Big(0 \le t_2^2 - \hat{t}_2^1 - p_2^1 \\
&\quad \wedge 0 \le t_2^3 - \hat{t}_2^1 - p_2^1 - p_2^2 \wedge 0 \le t_2^4 - \hat{t}_2^1 - p_2^1 - p_2^2 - p_2^3 \\
&\quad \wedge T_1^2 \le t_2^2 - p_2^1 \wedge T_1^2 \le t_2^3 - p_2^1 - p_2^2 \wedge T_1^2 \le t_2^4 - p_2^1 - p_2^2 - p_2^3 \\
&\quad \wedge T_1^3 \le t_2^2 \wedge T_1^3 \le t_2^3 - p_2^2 \wedge T_1^3 \le t_2^4 - p_2^2 - p_2^3 \\
&\qquad \wedge T_1^4 \le t_2^3 \wedge T_1^4 \le t_2^4 - p_2^3\Big) \cdot \prod_{j=1}^{3} f_{P_2^j}\left(p_2^j\right)\Big]\,\mathrm{d}p_2^3\,\mathrm{d}p_2^2\,\mathrm{d}p_2^1 \\
&= \int_{p_2^{\min,1}}^{p_2^{\max,1}} \int_{p_2^{\min,2}}^{p_2^{\max,2}} \int_{p_2^{\min,3}}^{p_2^{\max,3}} \Big[H\Big(\min\Big(t_2^2 - \hat{t}_2^1 - p_2^1,\ t_2^3 - \hat{t}_2^1 - p_2^1 - p_2^2, \\
&t_2^4 - \hat{t}_2^1 - p_2^1 - p_2^2 - p_2^3\Big)\Big) \cdot F_{T_1^2T_1^3T_1^4}\Big(\min\Big(t_2^2 - p_2^1,\ t_2^3 - p_2^1 - p_2^2,\ t_2^4 - p_2^1 - p_2^2 - p_2^3\Big), \\
&\quad \min\Big(t_2^2,\ t_2^3 - p_2^2,\ t_2^4 - p_2^2 - p_2^3\Big), \min\Big(t_2^3,\ t_2^4 - p_2^3\Big)\Big) \cdot \prod_{j=1}^{3} f_{P_2^j}\left(p_2^j\right)\Big]\,\mathrm{d}p_2^3\,\mathrm{d}p_2^2\,\mathrm{d}p_2^1\ .
\end{aligned}
\tag{3.19}
$$

Based on $F_{T_2^2\cdots T_2^{m+1}}$, the computation of the scheduled start time and joint CDF of the entry times of the products $\sigma_3,\dots,\sigma_n$ can be carried out in the same way.

Expected values of makespan and number of conflicted products

The minimum and maximum possible entry times $t_n^{\min,k}$ and $t_n^{\max,k}$, $k = 2, \dots, m$ of the last product can be obtained by planning the start times of all products with their minimum and maximum process times $p_i^{\min,k}$, $p_i^{\max,k}$ as described in Section 3.3.1. The CDF of the makespan T_n^{m+1} is then

$$F_{T_n^{m+1}}\left(t_n^{m+1}\right) = F_{T_n^2 \cdots T_n^m T_n^{m+1}}\left(t_n^{\max,2}, \dots, t_n^{\max,m}, t_n^{m+1}\right) . \tag{3.20}$$

The expected value of the makespan is

$$\mathrm{E}\left(T_n^{m+1}\right) = \int_0^{t_n^{\max,m+1}} \left(1 - F_{T_n^{m+1}}\left(t_n^{m+1}\right)\right) \mathrm{d}t_n^{m+1} . \tag{3.21}$$

Product σ_1 is always processed without conflicts. A product σ_i, $i = 2, \dots, n$ is processed without conflicts with the probability γ_i according to (3.16). Therefore, the number of conflicted products N_c is a Poissonian binomial random variable [28] with the parameters $0, 1 - \gamma_2, \dots, 1 - \gamma_n$ and the expected value

$$E(N_\mathrm{c}) = n - 1 - \sum_{i=2}^{n} \gamma_i . \tag{3.22}$$

Because of (3.5), it follows that

$$E(N_\mathrm{c}) \leq (n-1)(1 - \gamma_\mathrm{set}) . \tag{3.23}$$

The high dimensionality of the integrals involved in (3.18) for a high number m of machines makes the evaluation of (3.18) difficult. The explicit calculation of (3.18) can be avoided by using Monte Carlo simulations, making the procedure practicable for a large number of machines and products. This is described in the following section.

3.3.3. Stochastic process times, Monte-Carlo-based solution

For the solution based on Monte-Carlo (MC) simulations, the production process of all n products is simulated n_s times. The simulation generates samples of random process times ${}_s p_i^k$, $s = 1, \dots, n_\mathrm{s}$ based on the known PDFs $f_{P_i^k}$. A set of values ${}_s p_i^k$, $i = 1, \dots, n$, $k = 1, \dots, m$, represents all random process times of the simulation sample s.

The scheduling is carried out individually for each sample s and product σ_1 as in the deterministic case described in Section 3.3.1. For all samples, product σ_1 is started at ${}_s t_1^1 = 0$. Using (3.7) to (3.11), the earliest possible start time ${}_s \hat{t}_2^1$ for a conflict free processing of σ_2 is computed for every sample s. Based in these results, a single scheduled start time $\hat{t}_2^1$ of σ_2 is then computed for the whole sample. Using the given probability γ_{set}, the scheduled start time $\hat{t}_2^1$ is the (interpolated [54]) γ_{set}-percentile value of the earliest possible start times ${}_1\hat{t}_2^1, \dots, {}_{n_\text{s}}\hat{t}_2^1$ [39]. Finally, for every sample s, the actual entry times of product σ_2 depend also on the actual entry times of σ_1:

$$
\begin{aligned}
{}_st_2^1 &= \max\left(\hat{t}_2^1,\, {}_st_1^2\right) \;, && \text{(3.24a)}\\
{}_st_2^2 &= \max\left({}_st_2^1 + {}_sp_2^1,\, {}_st_1^3\right) \;, && \text{(3.24b)}\\
&\;\vdots\\
{}_st_2^m &= \max\left({}_st_2^{m-1} + {}_sp_2^{m-1},\, {}_st_1^{m+1}\right) \;, && \text{(3.24c)}\\
{}_st_2^{m+1} &= {}_st_2^m + {}_sp_2^m \;. && \text{(3.24d)}
\end{aligned}
$$

With the actual entry times ${}_st_2^k$ for every sample s, the start time planning of the products $\sigma_3, \dots, \sigma_n$ can continue in the same way. During this computation the number of conflicted products ${}_sN_\text{c}$ for each sample s is also determined. The estimated expected values of the makespan and the number of conflicted products are indicated by a bar and obtained as the sample mean values, i. e.,

$$
\begin{aligned}
\overline{T}_n^{m+1} &= \text{mean}\left({}_1t_n^{m+1}, \dots, {}_{n_\text{s}}t_n^{m+1}\right) \;, && \text{(3.25)}\\
\overline{N}_\text{c} &= \text{mean}({}_1N_\text{c}, \dots, {}_{n_\text{s}}N_\text{c}) \;. && \text{(3.26)}
\end{aligned}
$$

For a sufficiently high n_s, (3.26) gives the same results as (3.22).

3.4. Case studies

The practical application of the proposed computational methods is presented in two case studies. In Section 3.4.1, results of the analytical optimization method and the MC-based method are compared. In Section 3.4.2, a problem similar to the actual production process in a rolling mill is solved using the MC-based optimization method.

3.4.1. Example with 3 machines

In this section, a system of $m = 3$ consecutive machines processing $n = 4$ products is considered. The stochastic process times at each machine have the same distribution for all products. At machine 1, the process time is normally distributed with mean value $\mu = 5$ and the standard deviation $\sigma = 2$, truncated to the range

[3, 10]. At machine 2, the process time is uniformly distributed between 1 and 2, and, at machine 3, it is uniformly distributed between 5 and 10.

The joint CDF $F_{T_1^2T_1^3T_1^4}(t_1^2, t_1^3, t_1^4)$ of the entry times of product 1 according to (3.14) is first computed as described in Section 3.3.2. For a given probability γ_{set}, the scheduled start time $\hat{t}_2^1$ of product σ_2 is then computed using (3.16). Based on $\hat{t}_2^1$, the joint CDF $F_{T_2^2T_2^3T_2^4}(t_2^2, t_2^3, t_2^4)$ of the entry times of product σ_2 is calculated using (3.19). These steps are then repeated for the products σ_3 and σ_4. All the equations have to be numerically solved. The occurring multiple integrals are computationally laborious even for systems with only a few machines.

Based on $F_{T_4^2T_4^3T_4^4}(t_4^2, t_4^3, t_4^4)$, the CDF $F_{T_4^4}(t_4^4)$ of the makespan is computed according to (3.20). For comparison, calculations were carried out using the analytical and the MC-based solution method (sample size $n_{\text{s}} = 1000$). Figure 3.2 shows the resulting PDF $f_{T_4^4} = \frac{\mathrm{d}}{\mathrm{d}t_4^4} F_{T_4^4}$ of the makespan in comparison with a histogram of the values obtained by the MC-based simulation.

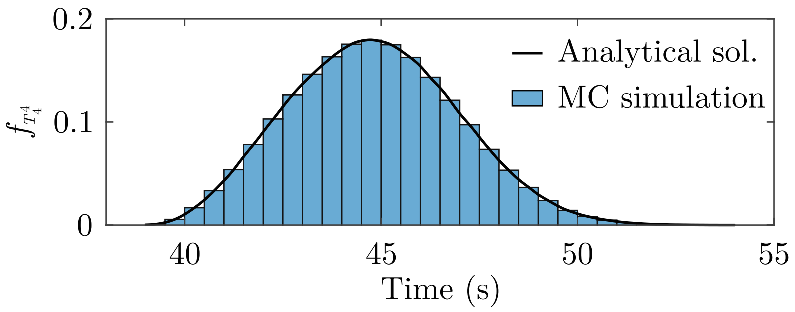

Figure 3.2.: Comparison of the PDF of the makespan T_4^4 obtained by the analytical solution and a MC-based simulation of the production process.

The expected makespan $\mathrm{E}(T_4^4)$ can be computed according to (3.21). The expected number of conflicted products $\mathrm{E}(N_{\text{c}})$ based on γ_{set} follows from (3.22). The weightings in the cost function (3.3) are $w_{\text{t}} = 1\,\text{s}^{-1}$ and $w_{\text{c}} = 10$.

To find the optimal probability γ_{set}^* a line search algorithm [12] was used. For the analytical computation as well as the MC-based solution, the optimal probability of conflict-free processing is found to be $\gamma_{\text{set}}^* = 0.85$. Figure 3.3 shows the calculated costs J using the analytical and the MC-based solution method as functions of the probability γ_{set}. The numerical evaluation of the integrals in the computation of $F_{T_1^2T_1^3T_1^4}$ are computationally demanding and required a computing time of around 10 s with a RAM requirement of a few GB. In contrast, the MC-based solution method required roughly only 0.1 % of the computation time of the analytical method without any significant memory requirement.

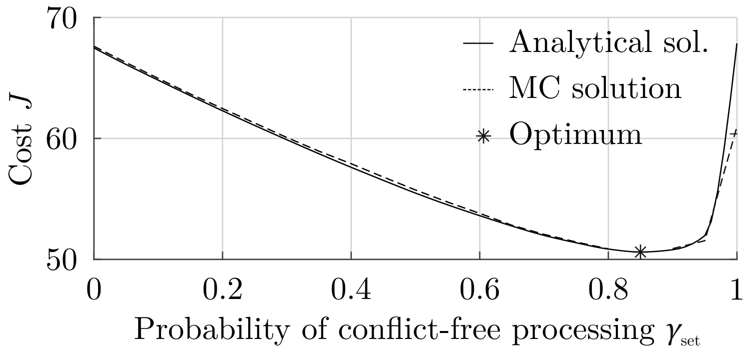

Figure 3.3.: Comparison of the cost J obtained by the analytical and the MC-based solution method.

3.4.2. Example with 7 machines

This example is based on the real production process in the considered steel hot rolling mill. However, a simpler model of the plant is used for starting time planning, as only production at one line is considered here. The model of this line consists of $m = 7$ machines in a series. Machines 3 and 7 process products with stochastic process time. All other machines operate with deterministic process times. In this scenario 100 products are processed. The deterministic process times are identical for all products and range, depending on the machine, from 8 s to 40 s. The stochastic process times at machines 3 and 7 may be different for each product and are given by estimated PDFs.

Machine 3 is the reversing rolling mill stand and processes the products in this scenario in three to six passes. Between the passes, manual control operations are required. These operations, slip between the products and the roller table during the reversal, and other factors cause uncertain process times. Hence, the process times of machine 3 are treated as random variables. Their frequency distributions were obtained for each product from process time samples recorded by the material tracking system of the plant. From these data, continuous PDFs were estimated using a kernel density estimator with a normal kernel smoothing function and a bandwidth of 0.1 s [51]. As an example, Figure 3.4 shows the empirical relative frequency distribution of recorded process times along with the estimated kernel distribution for the product σ_{40} at machine 3.

Machine 7 represents the product packaging and discharging unit at the end of the production line. Due to random waiting times, the process times at this machine also represent random variables. In accordance with the data obtained by the material tracking system, these process times are approximated by uniform distributions $\text{unif}\left(p_i^{\min,7}, p_i^{\max,7}\right)$ with $11 \leq p_i^{\min,7} \leq 28$, $20 \leq p_i^{\max,7} \leq 40$ for the considered products σ_i, $i = 1, \ldots, 100$.

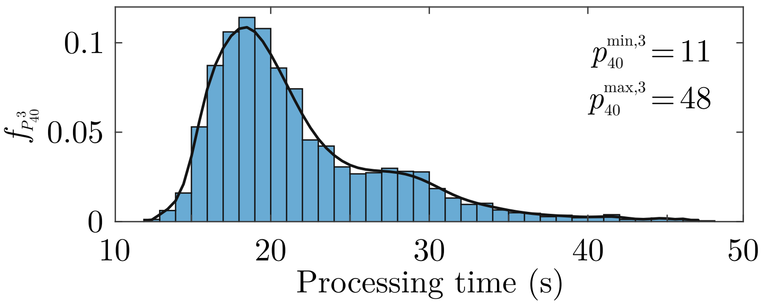

Figure 3.4.: Example of the estimated PDF of the process time of product σ_{40} at machine 3.

The optimization of the scheduled start times was done by MC simulations, as described in Section 3.3.3. The simulations were carried out using a sample size of $n_\mathrm{s} = 1000$. The weightings in the cost function (3.3) were chosen as $w_\mathrm{t} = 1\,\mathrm{s}^{-1}$ and $w_\mathrm{c} = 13$.

The optimal probability was found as $\gamma_\mathrm{set}^* = 0.74$. Figure 3.5 shows the mean values $\overline{T}_{100}^8$, $\overline{N}_\mathrm{c}$ as well as the resulting cost J as a function of the chosen probability γ_set.

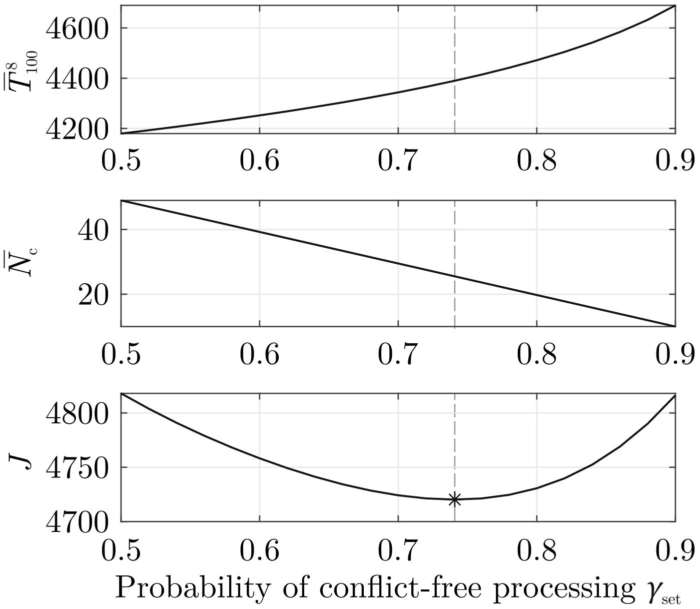

Figure 3.5.: Estimated makespan $\overline{T}_{100}^8$, estimated number of conflicted products $\overline{N}_\mathrm{c}$, and cost J as a function of γ_set for the case $m = 7$. The optimal probability $\gamma_\mathrm{set}^* = 0.74$ is marked by $*$.

In the following, these results are compared with the start time planning based on deterministic process times as described in Section 3.3.1. In this case, the expected values and the maximum values of the process times are considered as deterministic values for the planning approach. After the determination of the scheduled start times, the production process is simulated using $n_\mathrm{s} = 1000$ new samples of random process times.

Using the expected values of the process times as their assumed deterministic values leads to a low probability of processing products without conflicts and consequently to a high number of conflicted products. In contrast, using the maximum values $p_i^{\max,j}$ avoids any conflicted products at the cost of a significantly extended makespan. The production costs of this two planning solutions using deterministic process times are notably higher than the costs of the solution obtained by considering the PDFs of the stochastic process times.

Next, the planning results using various percentiles q of the random processing times as determinitic values were investigated. In analogy to finding an optimal probability γ_set^*, a line search algorithm was used to find a single optimal percentile q^* of the random process times. A value of $q^* = 0.77$ was found to be optimal. It turns out that using this value leads to a result that is slightly worse than taking PDFs into account. This may be due to the fact that the PDFs of the process times are usually asymmetrical as can be seen in the example in Figure 3.4, which is not captured by the percentile value.

The results are compared in Table 3.1. In the second column, γ is the resulting probability when using expected or maximum values of the process times and the computed optimal value when using PDFs. The use of the expected values, the maximum values, or the percentiles of the stochastic process times can be simply carried out even without the need of estimating their distribution functions. However, these strategies lead to suboptimal results.

Table 3.1.: Results of different planning strategies for stochastic process times. The variable γ is the resulting probability of conflict free processing.

Method using ...	γ	$\overline{T}_{100}^{8}$	$\overline{N}_\mathrm{c}$	J
Expected values of process times	0.12	3974	88	5118
Maximum values of process times	1	6446	0	6446
Optimal percentile of process times	0.69	4346	31	4749
PDFs of process times	0.74	4371	26	4709

3.5. Start time planning in the steel hot rolling mill

3.5.1. Calculation of earliest possible start times

For start time planning in the steel hot rolling mill of BÖHLER, the processing of the products is modeled as described in Chapter 2, i. e., with checkpoints, checkpoint blocking, and consideration of the respective product length. Although the process times at some parts of the plant (e. g. reversing mill stand, see Section A.3.2) or at the product packaging and discharge unit can be regarded as stochastic, start time planning (usually only for the next few products) during ongoing operations is carried out with purely deterministic process times $p_i^{\mathrm{in},k,l}$ and $p_i^{\mathrm{thru},k,l}$. The determination of the earliest possible start times of the product can thus be done by analogy to the deterministic method described in Section 3.3.1. However, instead of (3.2), relation (2.8) must be satisfied.

The process time in the heating box is a special case. Most products can stay in the heating box between the checkpoints 22 and 23 for a longer duration. For these products, the process time $p_i^{\mathrm{thru},22,23}$ must satisfy $p_i^{\mathrm{HB,min}} \leq p_i^{\mathrm{thru},22,23} \leq p_i^{\mathrm{HB,max}}$. Products which are not allowed to stay in the heating box pass it in the minimum possible duration, $p_i^{\mathrm{thru},22,23} = p_i^{\mathrm{HB,min}}$, and $p_i^{\mathrm{HB,max}} = p_i^{\mathrm{HB,min}}$. In the hot rolling mill, it is desired to start products as early as possible and to use the heating box as a buffer before the profile rolling line. In this way, the different process times of individual products due to different product lengths can be better taken into account and unproductive times can be minimized.

For start time planning of an arbitrary product σ_i, first its passage times $t_i^{\mathrm{H},k}$ and $t_i^{\mathrm{T},k}$ are computed for all checkpoints $k \in \boldsymbol{r}_i$ according to (2.4) with the nominal start time $t_i^{\mathrm{H},r_i^1} = 0$. If σ_i enters the heating box, $p_i^{\mathrm{thru},22,23} = p_i^{\mathrm{HB,min}}$ is used to start with. Based on the intermediate values $t_i^{\mathrm{H},k}$ and $t_i^{\mathrm{T},k}$, the earliest actually possible passage times of σ_i are computed considering the checkpoint release times $t_{i-1}^{\mathrm{R},k}$ after the previous product σ_{i-1}, as outlined in Algorithm NextStartTime.

After the computation of the nominal passage times $t_i^{\mathrm{H},k}$ and $t_i^{\mathrm{T},k}$, they satisfy (2.8). If σ_i is processed at the profile rolling line and if the bottleneck of processing σ_i lies after the entry of the heating box, σ_i can be started earlier with a longer stay in the heating box. This is considered by moving the passage times $t_i^{\mathrm{H},k}$ and $t_i^{\mathrm{T},k}$, $k \leq 22$ forward while respecting (2.8) and $p_i^{\mathrm{thru},22,23} \leq p_i^{\mathrm{HB,max}}$.

Based on the computed passage times $t_i^{\mathrm{H},k}$ and $t_i^{\mathrm{T},k}$, the checkpoint blocking rules defined for σ_i are applied and the release times $t_i^{\mathrm{R},k}$, $k = 1, \ldots, 34$ after σ_i are computed. Subsequently, start time planning of the next product σ_{i+1} can continue by analogy.

Algorithm NextStartTime: Compute passage times of a product σ_i according to checkpoint release times after a previous product σ_{i-1}.

```
Input: Release times $t_{i-1}^{\mathrm{R},k}$ $\forall k = 1, \dots, 34$, route $\boldsymbol{r}_i$, nominal passage times
    $\left(t_i^{\mathrm{H},k},\, t_i^{\mathrm{T},k}\right)$ $\forall k \in \boldsymbol{r}_i$, if heating box involved: $\left(p_i^{\mathrm{HB,min}},\, p_i^{\mathrm{HB,max}}\right)$
Output: Earliest possible passage times $\left(t_i^{\mathrm{H},k},\, t_i^{\mathrm{T},k}\right)$ $\forall k \in \boldsymbol{r}_i$

// Compute possible passage times of $\sigma_i$ by postponing given
   nominal passage times
$d_{\mathrm{pp}} = \max_{k \in \boldsymbol{r}_i}\left(t_{i-1}^{\mathrm{R},k} - t_i^{\mathrm{H},k}\right)$
$t_i^{\mathrm{H},k} \leftarrow t_i^{\mathrm{H},k} + d_{\mathrm{pp}}$
$t_i^{\mathrm{T},k} \leftarrow t_i^{\mathrm{T},k} + d_{\mathrm{pp}}$

// If heating box involved move forward passage times of $\sigma_i$
   upstream of checkpoint 22 (heating box entry)
if $22 \in \boldsymbol{r}_i$ then
    $d_{\mathrm{fw}} = \min\left(\min_{k \in \boldsymbol{r}_i \wedge k \leq 22}\left(t_i^{\mathrm{H},k} - t_{i-1}^{\mathrm{R},k}\right),\, p_i^{\mathrm{HB,max}} - p_i^{\mathrm{HB,min}}\right)$
    $t_i^{\mathrm{H},k} \leftarrow t_i^{\mathrm{H},k} - d_{\mathrm{fw}}$ $\forall (k \in \boldsymbol{r}_i \wedge k \leq 22)$
    $t_i^{\mathrm{T},k} \leftarrow t_i^{\mathrm{T},k} - d_{\mathrm{fw}}$ $\forall (k \in \boldsymbol{r}_i \wedge k \leq 22)$
end
return $\left(t_i^{\mathrm{H},k},\, t_i^{\mathrm{T},k}\right)$ $\forall k \in \boldsymbol{r}_i$
```

3.5.2. Evaluation of planning results

In the following, the simulated makespan and product throughput of the production process planned with the start times calculated as described in Section 3.5.1 are compared to the actual production process. In the evaluated production period of 4.8 h, 222 products (billets) were processed. The products were processed in groups at the profile and at the flat rolling line. The actual production is derived from timestamps obtained from the material tracking system of the plant. The planned start times are calculated based on the process time models described in Appendix A and evaluated in Section A.8

The number of products processed as a function of time is shown in Figure 3.6. The blue curve represents the production process when the products are started at their calculated start times. The green curve represents the actual production process. During the actual production, delays occur due to unplanned events in the production process. These can be identified as horizontal sections of the green curve. In order to make the planned and actual production process comparable in terms of productivity, this unplanned and unpredictable delays were removed from the actual production process. In fact, actual time intervals between product starts that were more than twice as long as the corresponding planned start time intervals were replaced by the respective planned time intervals. The production process without unplanned delays is shown as red curve.

The actual makespan of the production of the 222 products was 4.81 h. Without delays, 3.92 h would be required. The production process according to the calculated earliest possible start times would require 3.43 h. This results in a possible increase of the number of products manufactured per hour by 14 %.

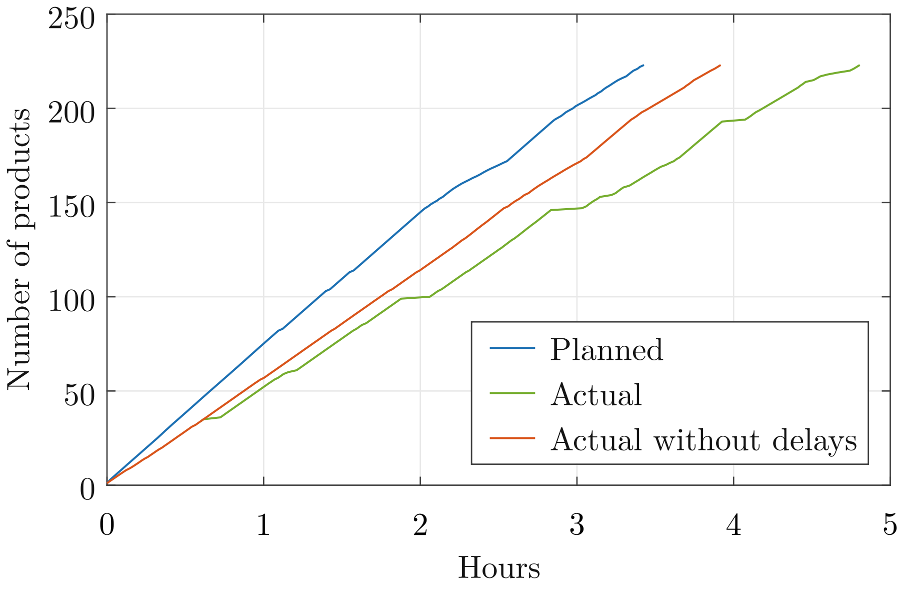

Figure 3.6.: Production process of 222 products. Planned production based on calculated earliest possible start times, actual production on records of the material tracking system. Delays in the actual production process can be identified as horizontal sections and are removed in the red curve.

Chapter 4.

Schedule optimization

Parts of this chapter were published in similar form in the author's previous publications [32] and [34].

4.1. Introduction

While Chapter 3 dealt with the start times of individual products, this chapter deals with the order of jobs. A job is the basic element to be scheduled and consists of one or more related (typically identical) products. The individual products of each job must be processed sequentially without switching to products of another job. The passage times of a job are derived from the passage times of its first and last product, as described in Section 4.2.1. A schedule consists of one or more P- and F-groups, which follow each other alternately. The notation and process simulation of jobs in a schedule is explained in Section 4.2.2.

The optimization problem is defined in Section 4.3. Section 4.4 investigates the exact optimization of individual P- and F-groups with some simplifications. The algorithm to solve the full optimization problem taking furnace assignments into account is described in Section 4.5. Its application to a use case is shown in Section 4.6.

4.2. Production process

4.2.1. Production process of jobs

Each job has a unique identification number (ID). As jobs are processed in groups, a vector $\boldsymbol{\sigma}_i$ contains the IDs of the jobs of group i. Thus, $\sigma_{i,j}$ is the ID of the job that is processed at the sequence position j within group i.

Similar to the processing of individual products, the processing of a job $\sigma_{i,j}$ is described by its passage times of all checkpoints along the route $\boldsymbol{r}_{i,j}$:

- $t_{i,j}^{\mathrm{H},k}$ is the time at which the head of the first product of job $\sigma_{i,j}$ passes checkpoint $k \in \boldsymbol{r}_{i,j}$,
- $t_{i,j}^{\mathrm{T},k}$ is the time at which the tail of the last product of job $\sigma_{i,j}$ passes checkpoint $k \in \boldsymbol{r}_{i,j}$.

If the products of a job are heated in the WBF, the routes of all products are identical. If the products of a job are heated in the INDU, the route of each product depends on the used chamber. The three INDU chambers are loaded with products always in cyclic order, i. e., $\dots, 1, 2, 3, 1, \dots$. Therefore, to fully specify the heating of a job $\sigma_{i,j}$ in the INDU, only the starting chamber in which the first product of the job $\sigma_{i,j}$ is heated needs to be defined. Consequently, there are three different possible routes of a job heated in the INDU.

The nominal passage times for all jobs and their corresponding routes in the admissible furnaces are determined individually for each job $\sigma_{i,j}$ before the schedule optimization. To determine the nominal passage times of a job $\sigma_{i,j}$, the processing of all of its $n_{i,j}$ products is simulated considering all relevant checkpoint blockings. In the following, an additional index l is used to denote the passge times $t_{i,j,l}^{\mathrm{H},k}$ and $t_{i,j,l}^{\mathrm{T},k}$, and the checkpoint release times $t_{i,j,l}^{\mathrm{R},k}$ associated with product $l = 1, \dots, n_{i,j}$ of $\sigma_{i,j}$. Before the first product 1, the plant is assumed to be empty, i. e., the release times of all checkpoints $k = 1, \dots, 34$ are $t_{i,j,0}^{\mathrm{R},k} = 0$. The first product is started at time 0. The start times of each product $l = 2, \dots, n_{i,j}$ are scheduled as early as possible according to Algorithm NextStartTime, taking all necessary checkpoint blockings into account. Because it is required to consider the different product routes representing the different chambers of the INDU (see checkpoints 5 to 10 in Figure 2.4), each product l has its individual route $\boldsymbol{r}_{i,j,l}$. The results of this computation are the nominal passage times $t_{i,j,l}^{\mathrm{H},k}$ and $t_{i,j,l}^{\mathrm{T},k}$ at all checkpoints $k \in \boldsymbol{r}_{i,j,l}$. Based on the passage times of the first and the last product, the nominal passage times of $\sigma_{i,j}$ are

$$t_{i,j}^{\mathrm{H},k} = t_{i,j,1}^{\mathrm{H},k} \ , \tag{4.1}$$

$$t_{i,j}^{\mathrm{T},k} = t_{i,j,n_{i,j}}^{\mathrm{T},k} \tag{4.2}$$

with

$$t_{i,j,1}^{\mathrm{H},r_{i,j,1}^{1}} = 0 \ .$$

To reduce the CPU time required for simulations, the nominal processing times of every job $\sigma_{i,j}$ and its possible routes through all admissible furnaces are precomputed and stored before the sequence optimization. If job $\sigma_{i,j}$ can be heated in the INDU, the nominal processings for the start of the first product in each of the three chambers are computed and stored. If $\sigma_{i,j}$ can also be heated in the WBF, the associated processing is computed and stored.

The primary goal of the sequence optimization is the maximisation of throughput and therefore the minimization of unproductive times during the processing of jobs. For the used model and the simulation results in form of passage times, the unproductive time is defined as time, during which checkpoint 12 is not occupied. Checkpoint 12 is used because it is the junction point of product routes from WBF and INDU (see Figure 2.4) and is passed by every product regardless of its route.

As described in Section 2.3.2, the throughput of the INDU is in general lower than that of the WBF, especially when more than three products heated consecutively exceed the number of INDU chambers. Therefore, the unproductive time within a job (i. e., the total time, during which checkpoint 12 is not occupied by any product) varies depending on whether the its products are heated in the INDU or in the WBF. Thus, for every job $\sigma_{i,j}$ which can be heated in both the WBF and the INDU, unproductives times $u_{i,j}^{\mathrm{WBF}}$ and $u_{i,j}^{\mathrm{INDU}}$, respectively, are computed in the form

$$u_{i,j}^{\star} = \sum_{l=2}^{n_{i,j}} t_{i,j,l}^{\mathrm{H},12} - t_{i,j,l-1}^{\mathrm{T},12} \ , \tag{4.3}$$

with

$$\star = \begin{cases} \text{WBF} & \text{if } \sigma_{i,j} \text{ is heated in the WBF,} \\ \text{INDU} & \text{if } \sigma_{i,j} \text{ is heated in the INDU.} \end{cases}$$

Finally

$$s_{i,j}^{\mathrm{INDU}} = u_{i,j}^{\mathrm{INDU}} - u_{i,j}^{\mathrm{WBF}} \ . \tag{4.4}$$

is the additional unproductive time that occurs when heating the products of job $\sigma_{i,j}$ in the INDU as compared to the WBF. The throughput of the INDU is also limited by operating speeds of the cranes 1 and 2 and is lower than the possible output of the WBF. For this reason $s_{i,j}^{\mathrm{INDU}}$ is positive for most jobs $\sigma_{i,j}$.

As an example, the nominal processing of a job $\sigma_{i,j}$ consisting of two products $\sigma_{i,j,1}$ and $\sigma_{i,j,2}$ is shown in Figure 4.1. If the products are heated in the WBF (a), the bottleneck of the processing is given by the blocking of checkpoint 2 until $\sigma_{i,j,1}$ has been processed at the reversing rolling stand and its tail has passed checkpoint 14. Part (b) of Figure 4.1 shows the processing when the products are heated in the INDU. It starts with the heating of $\sigma_{i,j,1}$ in chamber 1 (checkpoints 5 and 6). Subsequently, $\sigma_{i,j,2}$ is heated in chamber 2 (checkpoints 7 and 8). In this case, the bottleneck of the production results from the required transport time of the hot products from the roller table after the INDU to the roller table in front of the reversing rolling stand. This is represented by the blocking of checkpoint 11.

4.2.2. Production process of multiple groups

It is assumed that an optimization problem consists of m groups and that a schedule always starts with a P-group (it is straightforward to adapt the presented optimization procedures for a group sequence starting with an F-grooup). Therefore, the indices i of the P-groups are $\boldsymbol{I}_{\mathrm{P}} = \{1, 3, \ldots\}$ and the indices i of the F-groups are $\boldsymbol{I}_{\mathrm{F}} = \{2, 4, \ldots\}$.

The groups are processed in the order of their indices. A group i consists of n_i jobs. The time d_i required for retooling after a P-group $i \in \boldsymbol{I}_{\mathrm{P}}$ is known.

The IDs of all jobs of a P-group i are assembled in the set $\boldsymbol{S}_{\mathrm{P},i}$ with

$$n_i = |\boldsymbol{S}_{\mathrm{P},i}| = \text{const.} \quad \forall i \in \boldsymbol{I}_{\mathrm{P}} \ . \tag{4.5}$$

The set of IDs of all jobs that can be arranged into F-groups is $\boldsymbol{S}_{\mathrm{F}}$ with

$$\sum_{i \in \boldsymbol{I}_{\mathrm{F}}} n_i \leq |\boldsymbol{S}_{\mathrm{F}}| \ . \tag{4.6}$$

For each job with the ID $x \in \left(\bigcup_{i \in \boldsymbol{I}_{\mathrm{P}}} \boldsymbol{S}_{\mathrm{P},i} \cup \boldsymbol{S}_{\mathrm{F}}\right)$, there is a set $\boldsymbol{T}_x \subseteq \{0, 1\}$. It contains 0 if the job with the ID x can be heated in an INDU, and it contains 1 if this job can be heated in the WBF.

Based on the nominal process times, the simulation of jobs in a given schedule Θ is carried out as follows:

1. For each job $\sigma_{i,j}$, the route $\boldsymbol{r}_{i,j}$ according to the used furnace $\tau_{i,j}$ is determined. For jobs that are heated in the INDU ($\tau_{i,j} = 0$), $\boldsymbol{r}_{i,j}$ is selected such that the chambers of the INDU are always occupied in the order 1, 2, 3, . . .

2. For all consecutive jobs heated in the WBF, it is determined whether and what extra time is required between the product discharges due to different prescribed temperature intervals.

3. For all consecutive jobs within P-groups, it is determined if the speed of the profile rolling line changes between them.

4. The corresponding precalculated nominal processing $\left(t_{i,j}^{\mathrm{H},k}, t_{i,j}^{\mathrm{T},k}\right)$ $\forall k \in \boldsymbol{r}_{i,j}$ corresponding to $\boldsymbol{r}_{i,j}$ is used.

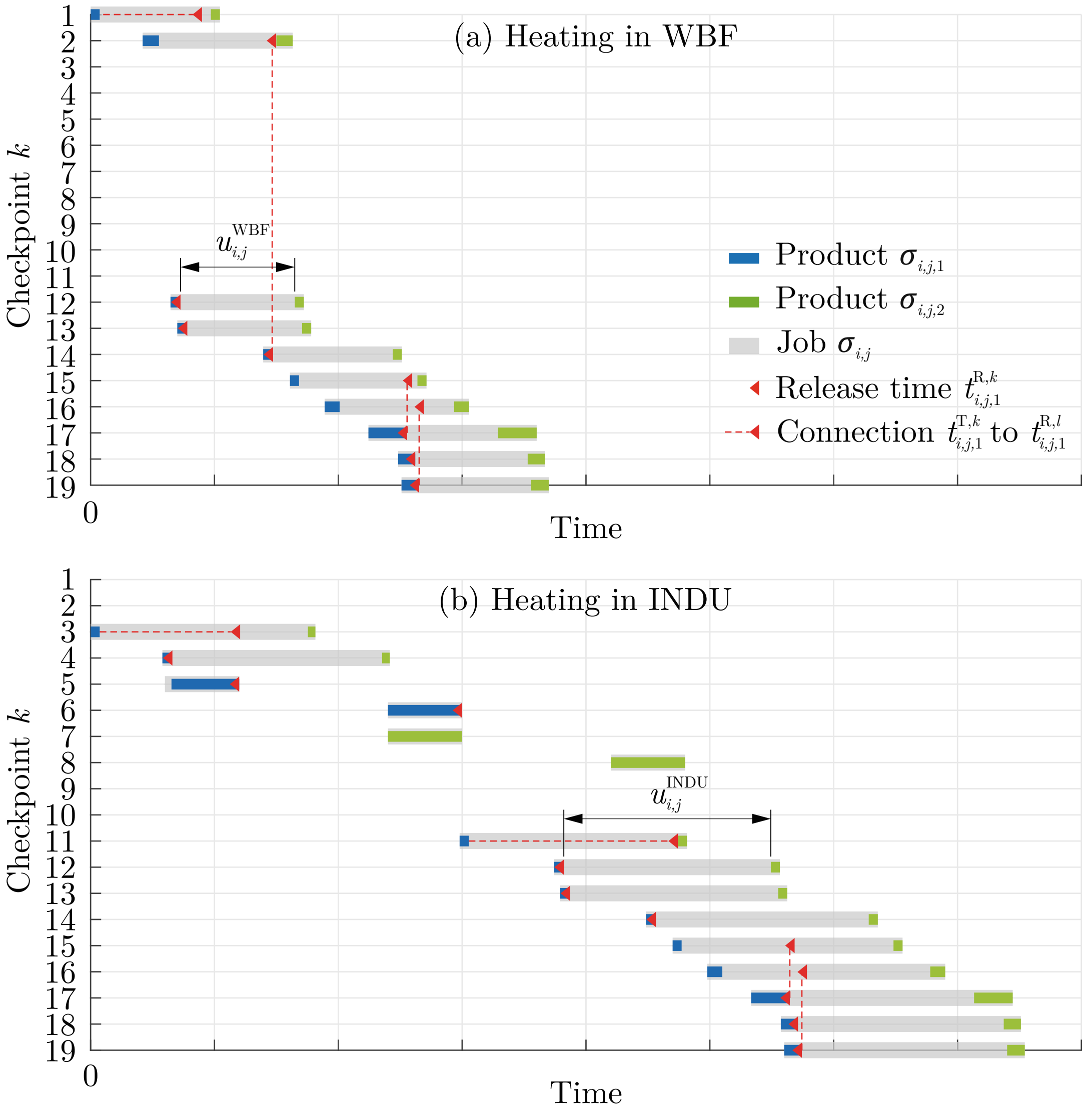

Figure 4.1.: Nominal processing of a job $\sigma_{i,j}$ consisting of two products heated in the WBF (a) and in the INDU starting with chamber 1 (b). Blue and green bars represent the occupation of each checkpoint by the products $\sigma_{i,j,1}$ and $\sigma_{i,j,2}$. The surrounding gray bars indicate the total occupation by the job $\sigma_{i,j}$. Checkpoint release times are shown in red.

5. Based on $t_{i,j-1}^{\mathrm{H},k}$ and $t_{i,j-1}^{\mathrm{T},k}$, the release times $t_{i,j-1}^{\mathrm{R},k}$ of all checkpoints $k \in C$ are determined considering all necessary checkpoint blockings after the previous job $\sigma_{i,j-1}$. Necessary extra times in the WBF are taken into account by blocking checkpoint 1, a change of the rolling speed at the profile rolling line by blocking checkpoint 25.

6. The earliest possible passage times $t_{i,j}^{\mathrm{H},k}$ and $t_{i,j}^{\mathrm{T},k}$, $k \in \boldsymbol{r}_{i,j}$, of job $\sigma_{i,j}$ are computed using Algorithm NextStartTime.

7. The steps 5–6 are computed for all jobs in the group.

8. If group i is a P-group, checkpoint 25 is blocked for the given retooling duration d_i after the last job σ_{i,n_i} of the group.

9. The passage times of all jobs in the remaining groups $i+1$ to m are computed as described in steps 5–8.

Figure 4.2 outlines this concept for the production of three groups. The occupation of a checkpoint k by a job $\sigma_{i,j}$ is represented by a horizontal bar between $t_{i,j}^{\mathrm{H},k}$ and $t_{i,j}^{\mathrm{T},k}$. The start times of the jobs are indicated by arrows ↓. The second job $\sigma_{1,2}$ consists of three products, which are heated in the chambers 1, 2, and 3 of the INDU. All other jobs are heated in the WBF. Note that while $\sigma_{1,2}$ starts before the job $\sigma_{1,1}$ because of the long INDU heating time, the job sequence specified by the schedule Θ is adhered to at checkpoint 12. After the last job $\sigma_{1,3}$ of group 1, the profile rolling line is blocked for the retooling time d_1. Because in this example $t_{1,3}^{\mathrm{R},25} = t_{3,1}^{\mathrm{H},25}$, the retooling time after group 1 is not exceeded by group 2. At the WBF, an additional extra time is required between the jobs $\sigma_{2,1}$ and $\sigma_{2,2}$. Between the jobs $\sigma_{3,1}$ and $\sigma_{3,2}$, a velocity change of the profile rolling line is required.

4.3. Optimization problem

The objective is to find a schedule Θ for m groups which

1. minimizes the sum of all unproductive times during the processing of Θ,

2. utilizes the retooling times d_i after the P-groups $i \in \boldsymbol{I}_{\mathrm{P}}$ by intermediate F-groups $i \in \boldsymbol{I}_{\mathrm{F}}$ as best as possible without exceeding d_i, and

3. considers all additional rules regarding the selection and sequencing of jobs.

The objectives 1 and 2 are evaluated based on the process simulation of Θ, i. e., the passage times $t_{i,j}^{\mathrm{H},k}$ and $t_{i,j}^{\mathrm{T},k}$, whereas objective 3 is determined directly from Θ. In the optimization process, the following variables have to be chosen.

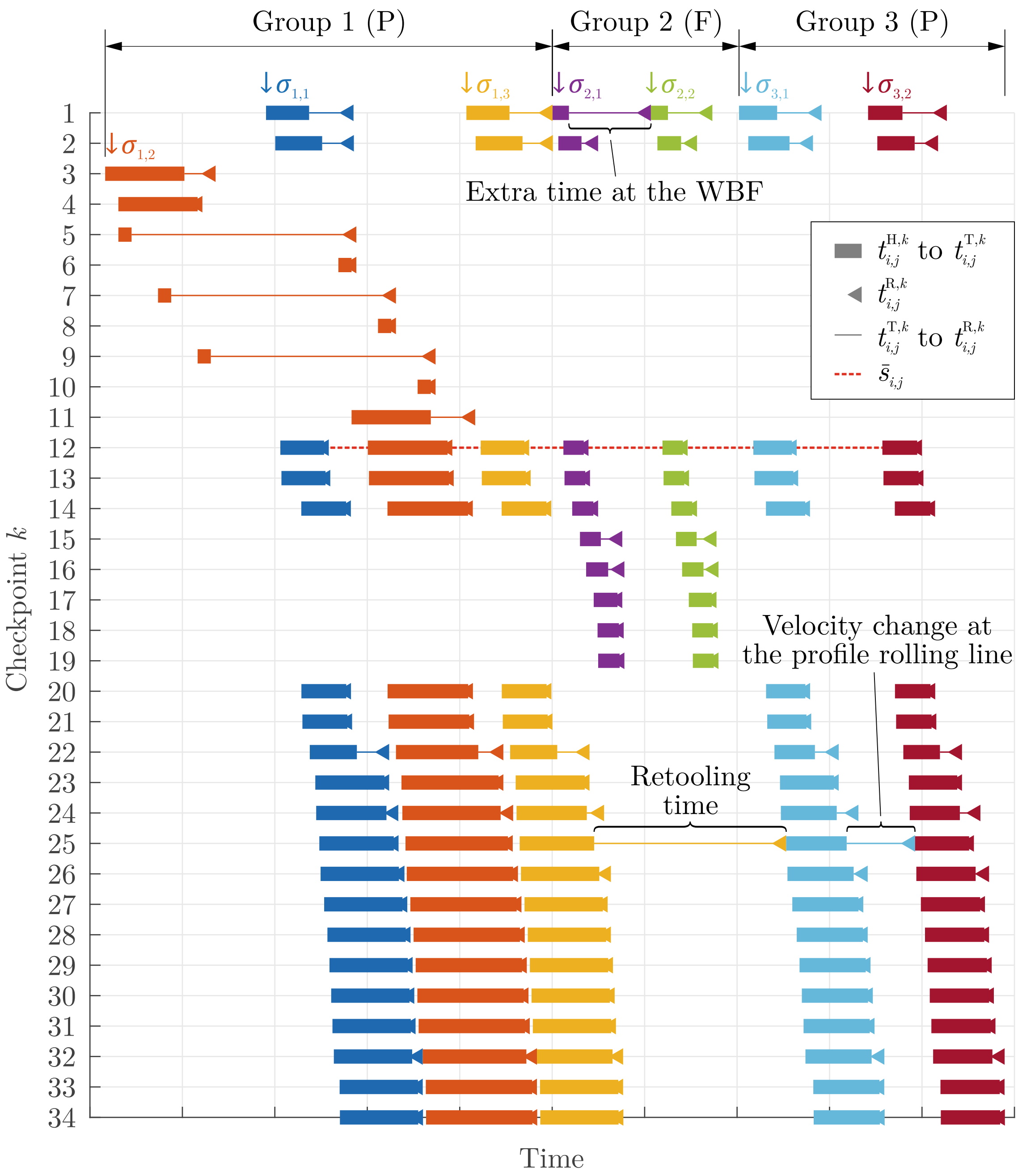

Figure 4.2.: Example of a production process, colors correspond to jobs.

- Job sequence $\boldsymbol{\sigma}_i$ within each P-group $i \in \boldsymbol{I}_\mathrm{P}$
- Job selection and sequence $\boldsymbol{\sigma}_i$ within each F-group $i \in \boldsymbol{I}_\mathrm{F}$
- Heating methods $\boldsymbol{\tau}_i$ of the jobs of all groups $i \in \{1, \ldots, m\}$

The job sequencing problem within a P-group is closely related to the ATSP [49]. Similar to the task of minimizing the sum of all unproductive times during the processing of the P-group, the ATSP involves finding the fastest possible route for a salesman to travel through a set of cities, where the travel time between two locations depends also on the direction of travel. The travel time between two cities corresponds to the setup time between consecutively processed jobs. In the model of the production process described in Chapter 2, the setup time between two jobs depends also on which of these jobs is processed first. The goal is to minimize the total travel time, which corresponds to minimizing the sum of setup times within the P-Group.

The optimization of an F-group, which also includes the selection of jobs, is similar to the OP [58]. In the OP, locations from a given set must be selected and visited. Each location has an assigned score. The goal of the optimization is to determine a route that maximizes the total score collected from visiting the selected locations, subject to constraints such as maximum travel time or distance. The travel time corresponds to the setup times between consecutively processed jobs. Differences between the optimization of an F-group and the OP aree the objective function and the maximum travel time constraint. The optimization goal for the scheduling of an F-group is to minimize the total unproductive time during the production while adhering to other defined rules regarding the arrangement of jobs. Which and how many jobs are processed is not directly considered in the cost function. An essential constraint is the given retooling time, which should not be exceeded by the makespan of the F-group. This is similar to the maximum travel time constraint in the OP, however the makespan of an F-group consists not only of the setup times between successive jobs but also of the process times of the selected jobs.

Moreover, there is another general difference between the scheduling problem addressed in this work and both the TSP and the OP. In the TSP and the OP, the distance between two consecutively visited locations (within a sequence) depends only on these two locations. In contrast, in the scheduling problem, because there are two different possible start points for some products and the fact that the capacity of the INDU is limited, the setup time between two products might also depend on other products in the job sequence. For example, consider a sequence of three jobs ($1 \rightarrow 2 \rightarrow 3$), where jobs 1 and 3 are heated in the INDU and job 2 in the WBF. If the process time of job 2 is short compared to the required heating time of job 1, the earliest possible starting time of job 3 might depend on the processing of job 1. In this situation, the passage times of all jobs 1, 2, and 3 are considered to determine the resulting unproductive times as described in Section 4.3.1. In contrast, in the objective functions of an ATSP or an OP, the distances between consecutive locations, e. g., (1 → 2) and (2 → 3), are considered separately.

4.3.1. Cost function

The individual cost items are discussed first. Later they are added up.

By analogy to the definition of unproductive time (4.3) within jobs, the sequence-dependent unproductive time $\bar{s}_{i,j}$ between jobs is defined. It is the time during which checkpoint 12 is not occupied between consecutive jobs, i. e.,

$$\bar{s}_{i,j} = \begin{cases} t^{\mathrm{H},12}_{i,j+1} - t^{\mathrm{T},12}_{i,j} & \text{if } i \in \{1,\dots,m\} \wedge j < n_i \ , \\ t^{\mathrm{H},12}_{i+1,1} - t^{\mathrm{T},12}_{i,n_i} & \text{if } i \in \{1,\dots,m-1\} \wedge j = n_i \ , \\ 0 & \text{if } i = m \wedge j = n_m \ . \end{cases} \tag{4.7}$$

Finally, the total unproductive time $s_{i,j}$ associated with job $\sigma_{i,j}$ includes, if heated in the INDU, the additional unproductive time $s^{\mathrm{INDU}}_{i,j}$ according to (4.4). This gives

$$s_{i,j} = \bar{s}_{i,j} + \begin{cases} s^{\mathrm{INDU}}_{i,j} & \text{if } \tau_{i,j} = 0 \ , \\ 0 & \text{if } \tau_{i,j} = 1 \ . \end{cases} \tag{4.8}$$

Whether the time d_{i-1} required for retooling of the profile rolling line between the P-groups $i-1$ and $i+1$ is exceeded by the production time of the F-group i is measured at checkpoint 25 by the variable

$$e_i = \begin{cases} t^{\mathrm{H},25}_{i+1,1} - t^{\mathrm{R},25}_{i-1,n_{i-1}} & \text{if } i \in \boldsymbol{I}_{\mathrm{F}} \ , \\ 0 & \text{else} \ , \end{cases} \tag{4.9}$$

with $e_i \geq 0$. If $e_i = 0$ and $\bar{s}_{i,n_i} > 0$ for $i \in \boldsymbol{I}_{\mathrm{F}}$ (as it is the case in Figure 4.2 for $i = 2$), the utilization of the retooling time by the (chosen) jobs in the F-Group i could be improved.

Depending on the type of a group, additional requirements regarding the arrangement of jobs must be considered in a schedule Θ. Based on this requirements, rules are defined and prioritized in four levels. Rules of priority 1 must be strictly adhered to and rules of priority 4 are considered least important, e. g.

- jobs with the same diameter must follow each consecutively within P-groups (priority 1),

- jobs of certain grades must or must not be placed at the beginning or the end of groups (priority 1),
- jobs of certain grades should or should not be placed at the beginning or at the end of groups (priorities 2 to 4),
- some jobs, which are selected by operators before the optimization, should be included with higher priority in the schedule of F-groups (priorities 2 to 4).

The number $n_i^{\mathrm{p}l}$ of violations of rules with priority l within group i is also considered in the cost function and weighted by a factor $w_{\mathrm{p}l}$.

The added up costs c_i of a group i are

$$c_i = w_{\mathrm{s}} \sum_{j=1}^{n_i} s_{i,j} + w_{\mathrm{e}} e_i + \sum_{l=1}^{4} w_{\mathrm{p}l} n_i^{\mathrm{p}l} \ . \tag{4.10}$$

The total costs of a schedule Θ containing m groups are obtained by summing up

$$c(\Theta) = \sum_{i=1}^{m} c_i \ . \tag{4.11}$$

The weights w_{s}, w_{e}, and $w_{\mathrm{p1}}, \ldots, w_{\mathrm{p4}}$ must be specified by the user so that they reflect the relative importance of the respective rules. For instance, it must be determined how many violations of a lower-priority rule are equivalent to a violation of a higher-priority rule. Compliance with rules of priority 1 could also be enforced by using only permutation functions (see Section 4.5.1) which do not return schedules that violate priority 1 rules. However, during the development of the sequence optimization algorithm, it turned out that choosing sufficiently large values of w_{p1} offers more flexibility.

4.3.2. Optimization problem

The optimization problem is defined as

$$\min_{\Theta,\, n_i \forall i \in \boldsymbol{I}_{\mathrm{F}}} c(\Theta) \tag{4.12a}$$

$$\text{s.t.} \quad \sigma_{i,j} \in \boldsymbol{S}_{\mathrm{P},i} \quad \forall (i \in \boldsymbol{I}_{\mathrm{P}},\ j \in \{1, \ldots, n_i\}) \ , \tag{4.12b}$$

$$\sigma_{i,j} \in \boldsymbol{S}_{\mathrm{F}} \quad \forall (i \in \boldsymbol{I}_{\mathrm{F}},\ j \in \{1, \ldots, n_i\}) \ , \tag{4.12c}$$

$$\sigma_{i,j} \neq \sigma_{v,w} \quad \forall (i \neq v \lor j \neq w) \ , \tag{4.12d}$$

$$\tau_{i,j} \in T_{\sigma_{i,j}} \ , \tag{4.12e}$$

$$\text{(4.1)–(4.6).} \tag{4.12f}$$

The constraints (4.12b) and (4.12c) ensure that each group contains only jobs from the pool assigned to it, (4.12d) ensures that each job is included in the schedule at most once. The relation (4.12e) ensures that each job is heated in an admissible furnace.

An optimization problem considered for the actual production usually includes multiple P- and F-groups. In this case, there are also dependencies between the groups which must be considered in (4.12):

- The unproductive times between subsequent groups contribute to the cost function.
- All F-groups are populated from a common set of possible jobs.

Because of these dependencies, and the fact that $s_{i,j}$ depends not only on the subsequent jobs $\sigma_{i,j}$ and $\sigma_{i,j+1}$ but also on other jobs in the schedule, an optimization algorithm has to optimize all jobs within all groups of a schedule Θ together. However, to investigate the possibilities of exact solution methods, the optimization of individual groups is first examined.

4.4. Exact optimization of individual groups

The exact optimization of individual groups is analyzed using the following simplifications:

- A P-group may only consist of jobs with a single diameter, i. e., apart from other restrictions, the jobs can be arranged in any order within the group.
- The furnace assignment is not part of the optimization. All jobs which can be heated in the WBF are heated in the WBF. All other jobs are heated in the INDU.
- If jobs comprising in total more than three products are consecutively heated in the INDU, unproductive times will increase due to the capacity of the INDU being limited to three chambers. During the optimization process, no penalties were applied to the consecutive arrangement of jobs containing a total of more than three products. However, an evaluation of the optimized job sequences revealed that such arrangements did not occur.

These simplifications are required to facilitate exact solution methods.

For the optimization, the jobs of a P-group or the jobs in the pool of an F-group are numbered from 1 to n. The schedule of a group always starts with the job with index 1 and an ends with the job with index n. The jobs 1 and n can be added as artificial jobs, with defined setup costs from job 1 and to job n. The specification of these setup costs allows for example the consideration of desired WBF temperatures before and after the group. Moreover, it allows control of the position of jobs that must or should be arranged at the beginning or at the end of the group. In the following, jobs and their parameters are referred to using these unique indices $1, \dots, n$, e. g., $t_j^{\mathrm{H},12}$ is the passage time of the head of job j at checkpoint 12. Job j is however in general not the job at the sequence position j.

Before the optimization, a single process time at checkpoint 12

$$p_j = t_j^{\mathrm{T},12} - t_j^{\mathrm{H},12} \tag{4.13}$$

with $p_0 = p_{n+1} = 0$ is calculated based on the nominal passage times for every job j. The unproductive setup time between two consecutive jobs j and k is computed from simulations in the form

$$s_{j,k} = t_k^{\mathrm{H},12} - t_j^{\mathrm{T},12} + w_{j,k} \tag{4.14}$$

with $s_{j,j} = 0$. The term $w_{j,k} \geq 0$ is a penalty added to control the position of jobs with respect to the rules listed in Section 4.3.1. For example, if a job k must not be arranged at the beginning of the group, a sufficiently high penalty $w_{1,k}$ is added to $s_{1,k}$. This computation is also done upfront, i. e., before the optimization. Note that in general $s_{j,k} \neq s_{k,j}$.

4.4.1. Integer optimization of a P-Group

Optimization problem

Because of the mentioned simplifications, the optimization of a single P-group is equivalent to solving an ATSP [25]. For the integer programming formulation of the optimization problem, a binary decision variable $x_{j,k}$ is defined in the form

$$x_{j,k} = \begin{cases} 1 & \text{if job } j \text{ is followed by job } k, \\ 0 & \text{else.} \end{cases} \tag{4.15}$$

The optimization problem follows in the form

$$\min_{x_{j,k}} \sum_{j=1}^{n} \sum_{k=1}^{n} s_{j,k} x_{j,k} \tag{4.16a}$$

$$\text{s. t.} \quad \sum_{k=2}^{n} x_{j,k} = 1 \quad \forall j = 1, \ldots, n-1 \ , \tag{4.16b}$$

$$\sum_{j=1}^{n-1} x_{j,k} = 1 \quad \forall k = 1, \ldots, n \ . \tag{4.16c}$$

The objective function (4.16a) aims at minimizing the sum of unproductive setup times between all jobs. The constraint (4.16b) ensures that every job j except the last one has exactly one successor. The constraint (4.16c) ensures that every job k except the first one has exactly one predecessor.

The constraints specified so far do not prevent subtours, i. e. multiple job sequences which are not connected. For example, in a problem with the jobs $\{1, \ldots, 5\}$ an (impractical) solution could consist of the two subtours $(1 \rightarrow 5)$ and $(2 \rightarrow 3 \rightarrow 4 \rightarrow 2)$. Two methods of subtour elemination were examined.

Iterative addition of subtour constraints

To prevent subtours, for every possible subset of jobs $\boldsymbol{T} \subseteq \{1, \ldots, n\}$, the number of connections ($x_{j,k}$-values equal to 1) between the jobs in $\boldsymbol{T}$ must be strictly smaller than $|\boldsymbol{T}|$ [18]:

$$\sum_{j \in \boldsymbol{T}} \sum_{k \in \boldsymbol{T}} x_{j,k} \leq |\boldsymbol{T}| - 1 \ . \tag{4.17}$$

For n jobs, the number of possible subssets is 2^n. Hence, a high number of additional inequality constraints would have to be considered. Instead of adding all these constraints from the beginning, the following iterative approach is used:

1. Solve problem (4.16) without additional constraints.
2. Detect all subtours in the current solution.
3. If no subtour is present, go to step 6.
4. For every subtour consisting of the jobs $\boldsymbol{T}$ add the constraint (4.17) to the problem.
5. Solve the updated problem and go to step 2.
6. Return the final solution.

Miller-Tucker-Zemlin constraints

This method was initially proposed by Miller, Tucker, and Zemlin (MTZ) as a formulation for solving the TSP [44]. In their approach, each job (or node in the original context of the TSP) is assigned a unique position, represented by ordering variables u_j. The main idea of the MTZ formulation is to enforce a linear ordering of the jobs using these variables. Specifically, if job j is processed before job k, then $u_j < u_k$ must hold. To enforce this linear order and therefore eliminate subtours, additional constraints are incorporated into the optimization model:

$$2 \leq u_j \leq n \quad \forall j = 2, \dots n \ , \tag{4.18a}$$

$$u_j - u_k + 1 \leq (n-1)(1 - x_{j,k}) \quad \forall j, k = 2, \dots n \ . \tag{4.18b}$$

If the values u_j are further restricted to integer values, every value found corresponds to the position of the job j in the solution.

Computational results

To examine the exact optimization method, test examples with different numbers of jobs were created. For this purpose, the data of a real P-group (from the industrial plant) with 50 jobs was used. Additional jobs were added by duplicating existing jobs with minor random variations of the process times.

For the numerical solution of (4.16) the MATLAB solver `intlinprog` was used at an Intel i5-6300U CPU @ 2.40 GHz. The solver `intlinprog` processes the problem in the following steps [55]:

1. *Linear preprocessing* simplifies the problem by reducing the number of variables and constraints.
2. *Linear programming* solves a relaxed version of the MILP (without integer constraints) to find a preliminary solution.
3. *Mixed-integer preprocessing* refines the solution space by tightening bounds and modifying or removing constraints based on the integrality requirements of the optimization variables.
4. During the *cut generation* additional linear constraints (cuts) are added to the problem to narrow down the feasible region closer to integer solutions. Different types of cuts are employed depending on the problem.
5. A *heuristic search* is carried out to find feasible integer solutions and if possible improve the bounds on the objective function.

6. A *branch and bound* algorithm is used to systematically explore the solution space by branching on fractional variables and subsequently bounding the objective function value.

Table 4.1 lists the computing times required to optimize different group sizes for both discussed methods of subtour elimination. The CPU times given are the mean values from three repeated optimization runs. The results in Table 4.1 indicate that, for this specific problem and large P-groups, the iterative method consumes less CPU time than the Miller-Tucker-Zemlin constraints.

Table 4.1.: CPU times for the optimization of P-groups with n jobs. Iter: iterative addition of subtour constraints, MTZ: Miller-Tucker-Zemlin subtour constraints.

n	Iter	MTZ
10	< 1 s	< 1 s
50	5 s	2 s
100	8 s	25 s
150	39 s	78 s
200	118 s	455 s
300	1380 s	-

4.4.2. Integer optimization of an F-Group

Optimization problem

For the optimization of an F-group, jobs must be selected from a pool and their order must be determined. The goal is to utilize a given maximum production time d (corresponding to the time required for the concurrent retooling of the profile rolling line) as best as possible for the processing of selected jobs. However, d must not be exceeded when processing the F-group.

The pool of the F-group contains n jobs. A schedule starts always with job 1 and ends with job n. The remaining $n-2$ jobs can be selected and arranged between the jobs 1 and n. The process time p_j of every job j is given by (4.13) and the setup time $s_{j,k}$ between the subsequent jobs j and k by (4.14). The job selection and sequence is defined by the decision variables $x_{j,k}$ according to (4.15).

Similar to p_j and $s_{j,k}$, the specified production time d is considered at the checkpoint 12. The time span d contains the total process and setup time of all selected jobs and a remaining unused timespan $r \geq 0$,

$$d = \sum_{j=0}^{n+1} \sum_{k=0}^{n+1} (p_j + s_{j,k}) x_{j,k} + r \ . \tag{4.19}$$

The goal is to minimize the total unproductive time, i. e.

$$\sum_{j=0}^{n+1} \sum_{k=0}^{n+1} s_{j,k} x_{j,k} + r = d - \sum_{j=1}^{n} \sum_{k=1}^{n} p_j x_{j,k} \ .$$

The problem is formulated as an integer programming model in the form

$$\max_{x_{j,k}} \sum_{j=1}^{n} \sum_{k=1}^{n} p_j x_{j,k} \tag{4.20a}$$

$$\text{s. t.} \quad \sum_{j=1}^{n} \sum_{k=1}^{n} (p_j + s_{j,k}) x_{j,k} \leq d \ , \tag{4.20b}$$

$$\sum_{k=2}^{n} x_{1,k} = \sum_{j=1}^{n-1} x_{j,n} = 1 \ , \tag{4.20c}$$

$$\sum_{j=1}^{n-1} x_{j,i} \leq 1 \qquad \forall i = 2, \ldots, n-1 \ , \tag{4.20d}$$

$$\sum_{k=2}^{n} x_{i,k} \leq 1 \qquad \forall i = 2, \ldots, n-1 \ , \tag{4.20e}$$

$$\sum_{j=1}^{n-1} x_{j,i} = \sum_{k=2}^{n} x_{i,k} \qquad \forall i = 2, \ldots, n-1 \ , \tag{4.20f}$$

Constraint (4.20b) limits the total process time. Constraint (4.20c) ensures that the sequence starts with job 1 and ends with job n. Constraint (4.20d) ensures that each job has at most one direct predecessor, (4.20e) that each job has at most one direct successor. Constraint (4.20f) ensures that every job $i = 2, \ldots, n-1$ which is scheduled must have a direct predecessor and direct successor. Moreover, it ensures that if job i is not scheduled, it has neither a direct predecessor nor a direct successor.

Additional constraints have to be included to prevent subtours. This can be done again using the iterative approach or the MTZ formulation, both described in Subsection 4.4.1. In the MTZ formulation, $n-2$ additional decision variables $u_2, \ldots, u_{n-1}$ are used.

Computational results

Test examples consisting of F-group job pools of different sizes, each with n jobs, were compiled based on real production data from the industrial plant. For each example, the maximum production time d was selected such that a maximum of approximately 10 % of the jobs in the pool could be selected.

The solution of the optimization problem (4.20) was carried out using the MATLAB solver `intlinprog` at an Intel i5-6300U CPU @ 2.40 GHz. Table 4.2 lists the required computing times for all considered pool sizes n and for both considered methods of subtour elimination. The results in Table 4.2 indicate that the MTZ method consumes less CPU time than the iterative method. An analysis of the search progress using the iterative method showed that a higher number of iterations (compared to the optimization of P-groups) is required. This is likely the case as there might be more possibilities to generate subtours when in general not all jobs are part of the solution (in which case (4.16b) and (4.16c) must hold). Compared to the integer optimization of a P-group (Table 4.1), only smaller instances of F-groups could be solved and their optimization required considerable more CPU time. This is consistent with the finding that the OP is of considerably higher computational complexity than the TSP [25, 6].

Table 4.2.: CPU times for the optimization of F-groups with pools containing n jobs. Iter: iterative addition of subtour constraints, MTZ: Miller-Tucker-Zemlin subtour constraints.

n	Iter	MTZ
10	< 1 s	< 1 s
20	5 s	< 1 s
25	11 s	2 s
30	58 s	22 s
40	371 s	230 s
50	2242 s	608 s
70	6695 s	2195 s
100	-	28 524 s

4.4.3. Conclusions

It was shown that the exact optimization of individual groups can be carried out given the mentioned simplifications are allowed. The size of the P-groups that occur in practice is well below the limits of the used method. However, this does not apply to F-groups, because practically relevant problems can involve a pool of more than 70 jobs. This problem size cannot be solved within reasonable CPU times using the described methods.

In practice, as described in Section 4.3.1, a problem consists of several groups which have to be optimized together. In addition, the furnace assignment has to be decided in the optimization problem, taking into account the capacity of the INDU. To satisfy these requirements, to efficiently solve the combined optimization problem with multiple P- and F-groups, and to incorporate potential additional requirements of the schedule, a heuristic optimization algorithm described in the following section was developed.

4.5. Heuristic optimization of multiple groups

The concurrent scheduling of several groups of jobs and the selection of a furnace for some jobs result in a large search space of the optimization problem. A simulated annealing metaheuristic followed by a local search were developed and implemented to solve the combined optimization problem (4.12).

Local search iteratively explores the neighborhood of a current solution. The neighborhood of a solution consists of all the solutions which can be reached by making defined changes, also called “moves”, to the current solution. The moves and neighborhood functions used for the sequence optimization in the present work are described in Section 4.5.1. Within the neighborhood of a current solution, local search looks in each iteration for a new solution that is better than the current solution. For example, this can be the firstly found better solution or the best solution within the neighborhood. The process continues until no further improvements can be achieved because a local optimum has been reached.

To to able to find even better solutions, the simulated annealing algorithm developed in this work uses a probabilistic acceptance criterion to escape local optima. Similar approaches where proposed, for example in [15, 42, 48, 50]. Kirkpatrick et al. [31] described using a simulated annealing algorithm to solve the TSP.

The simulated annealing algorithm consists of the following steps [43]:

1. Initialize a start solution Θ and the parameter annealing temperature T.

2. Randomly select a solution Θ' from the neighborhood $\mathcal{N}(\Theta)$ of Θ.

3. If Θ' is better than Θ, set $\Theta \leftarrow \Theta'$, else set $\Theta \leftarrow \Theta'$ with the probability $p(\Theta, \Theta', T)$.

4. Update T

5. Repeat the steps 2–4 until a termination criterion is met.

Different ways to compute the probability of acceptance have been proposed [7]. In this work, the common formulation

$$p(\Theta, \Theta', T) = \exp\left(\frac{c(\Theta) - c(\Theta')}{T}\right) \tag{4.21}$$

based on the Boltzmann distribution [38] is used with the cost $c(\Theta)$ according to (4.11). In step 4 of each iteration, the annealing temperature T, is reduced so that it exponentially decays to zero. This iteratively reduces the probability of accepting a worse solution.

4.5.1. Permutation moves

Both simulated annealing and local search are based on neighborhood functions, which alter a current schedule by permutation moves. Permutation moves are applied to the current schedule $(\boldsymbol{\sigma}_i, \boldsymbol{\tau}_i)$ of a group i resulting in a permuted schedule denoted as $(\boldsymbol{\sigma}'_i, \boldsymbol{\tau}'_i)$. Five different moves are used and described in the following.

3-opt-oneway

This move changes the job sequence within a group i. Based on the three indices u, v, w with $1 \leq u < v < w \leq n_i + 1$,

$$\boldsymbol{\sigma}_i = \begin{bmatrix} \cdots \sigma_{i,u-1} & \sigma_{i,u} \cdots \sigma_{i,v-1} & \sigma_{i,v} \cdots \sigma_{i,w-1} & \sigma_{i,w} \cdots \end{bmatrix}$$

is rearranged into

$$\boldsymbol{\sigma}'_i = \begin{bmatrix} \cdots \sigma_{i,u-1} & \sigma_{i,v} \cdots \sigma_{i,w-1} & \sigma_{i,u} \cdots \sigma_{i,v-1} & \sigma_{i,w} \cdots \end{bmatrix} .$$

In unison, the same permutation is applied to $\boldsymbol{\tau}_i$. This move is an adapted version of the 3-opt move [1]. The number of possible 3-opt-oneway permutations of $\boldsymbol{\sigma}_i$ is $\binom{n_i+1}{3}$ and thus grows with the order $O(n_i^3)$.

Change-furnace

This move changes the assigned heating method $\tau_{i,j}$ of a job $\sigma_{i,j}$ with $\boldsymbol{T}_{\sigma_{i,j}} = \{0, 1\}$. The number of possible permutations of $\boldsymbol{\tau}_i$ equals the number of jobs in group i which can be heated in both the WBF and the INDU.

Add

This move inserts an unscheduled job $x \in \boldsymbol{S}_\mathrm{F}$ into $\boldsymbol{\sigma}_i$ of an F-group $i \in \boldsymbol{I}_\mathrm{F}$ after a selected position $j \in \{0, \ldots, n_i\}$. In $\boldsymbol{\tau}_i$, 1 is inserted after position j if $1 \in \boldsymbol{T}_x$, otherwise 0 is inserted, i. e., the inserted job is heated in the WBF by default if possible. The number of possible add moves in $\boldsymbol{\sigma}_i$ is of the order $O(n_i|\boldsymbol{S}_\mathrm{F}|)$.

Remove

This move removes a scheduled job from a selected position $j \in \{1, \ldots, n_i\}$ from $\boldsymbol{\sigma}_i$ and $\boldsymbol{\tau}_i$ of an F-group $i \in \boldsymbol{I}_\mathrm{F}$. The number of possible remove moves in $\boldsymbol{\sigma}_i$ is n_i.

Exchange

This move exchanges a job at a selected position $j \in \{1, \ldots, n_i\}$ from $\boldsymbol{\sigma}_i$ of an F-group $i \in \boldsymbol{I}_\mathrm{F}$ by another job $x \in \boldsymbol{S}_\mathrm{F}$. If job x is currently scheduled in an F-group k, $x \in \boldsymbol{\sigma}_k$, the job IDs and heating methods are swapped. If job x is not scheduled, $\sigma_{i,j} = x$ and $\tau_{i,j} = 1$, if $1 \in \boldsymbol{T}_x$, else $\tau_{i,j} = 0$ is set. The number of possible exchange moves in $\boldsymbol{\sigma}_i$ is of the order $O(n_i|\boldsymbol{S}_\mathrm{F}|)$.

4.5.2. Neighborhood functions

Based on the permutation moves from Subsection 4.5.1, two different neighborhood functions $\mathcal{N}_P(\boldsymbol{\sigma}_i, \boldsymbol{\tau}_i)$ and $\mathcal{N}_F(\boldsymbol{\sigma}_i, \boldsymbol{\tau}_i)$ are defined. $\mathcal{N}_P(\boldsymbol{\sigma}_i, \boldsymbol{\tau}_i)$ can be applied to the schedule of a P-group $i \in \boldsymbol{I}_\mathrm{P}$ and allows the moves 3-opt-oneway and change-furnace. $\mathcal{N}_F(\boldsymbol{\sigma}_i, \boldsymbol{\tau}_i)$ can be applied to the schedule of an F-group $i \in \boldsymbol{I}_\mathrm{F}$ and allows the moves 3-opt-oneway, change-furnace, add, remove, and exchange.

A neighbor of a schedule Θ is

$$\Theta' = \{\boldsymbol{\sigma}_1, \boldsymbol{\tau}_1, \ldots, \boldsymbol{\sigma}'_i, \boldsymbol{\tau}'_i, \ldots, \boldsymbol{\sigma}_m, \boldsymbol{\tau}_m\} \tag{4.22}$$

with

$$\{\boldsymbol{\sigma}'_i, \boldsymbol{\tau}'_i\} \in \begin{cases} \mathcal{N}_P(\boldsymbol{\sigma}_i, \boldsymbol{\tau}_i) & \text{if } i \in \boldsymbol{I}_\mathrm{P} \ , \\ \mathcal{N}_F(\boldsymbol{\sigma}_i, \boldsymbol{\tau}_i) & \text{if } i \in \boldsymbol{I}_\mathrm{F} \ . \end{cases} \tag{4.23}$$

The neighborhood function $\mathcal{N}(\Theta)$ returns the set of all possible neighbors Θ' generated by multiple application of $\mathcal{N}_F$ or $\mathcal{N}_P$ to the schedules of all groups $i = 1, \ldots, m$ in Θ. The size of $\mathcal{N}(\Theta)$ depends also on the number n_i of jobs in each group and the size $|\boldsymbol{S}_\mathrm{F}|$ of the pool of the F-groups. For values of n_i and $|\boldsymbol{S}_\mathrm{F}|$ that are typical in practice, $\mathcal{N}(\Theta)$ is dominated by the number of possible 3-opt-oneway moves and is therefore of the order $O(\sum_{i=1}^m n_i^3) \approx O(mn_\mathrm{max}^3)$. Here, $n_\mathrm{max} = \max_i n_i$ is the number of jobs in the largest group in Θ.

4.5.3. Optimization algorithm

The simulated annealing algorithm developed for the scheduling in the considered steel hot rolling mill explores possible changes within all m groups in every iteration. To ensure that the final solution is locally optimal, a local search is carried out based on the solution found by simulated annealing. The implementation is shown in Algorithm OptimizeSchedule.

At the beginning of the algorithm, a start solution Θ is generated by a greedy construction heuristic. The latter starts with a randomly chosen job $\sigma_{1,1} \in \boldsymbol{S}_{\mathrm{P},1}$ for P-group 1 and iteratively adds subsequent jobs to obtain the whole schedule Θ, which includes also the groups $2, \ldots, m$. The next job added is the one that causes the smallest costs. For each P-group $i \in \boldsymbol{I}_{\mathrm{P}}$, all defined jobs in $\boldsymbol{S}_{\mathrm{P},i}$ have to be added. For each F-group $i \in \boldsymbol{I}_{\mathrm{F}}$, jobs are added until the term e_i according to (4.9) becomes non-zero because processing of the F-group i exceeds the retooling time between the P-groups $i-1$ and $i+1$. The resulting cost value $c_0 = c(\Theta)$ of the start solution is computed according to (4.11). The annealing temperature T is then initialized such that a solution with cost wc_0 with $w > 1$ would initially be accepted with a probability p_0. Based on (4.21), the initial temperature is thus set to

$$T \leftarrow c_0 \frac{1-w}{\ln(p_0)} \ . \tag{4.24}$$

In practice, values of $w = 1.2$ and $p_0 = 0.8$ proved useful.

The main part of the simulated annealing algorithm consists of two nested loops. In the inner loop a random neighbor Θ' of the current solution Θ is chosen from $\mathcal{N}(\Theta)$. If Θ' causes a smaller cost than Θ, Θ' is always accepted as new current solution. Otherwise, Θ' is accepted with a probability based on the difference of the costs of Θ and Θ' according to (4.21). The best solution found so far is stored in Θ_{best}. This process is repeated k_T times with a constant value of T. Afterwards, in the outer loop, the annealing temperature is decreased in the form,

$$T \leftarrow rT \ , \quad 0 < r < 1 \ . \tag{4.25}$$

In practice, a value of $r = 0.98$ was used. The outer loop terminates either when there is no improvement of the best schedule for more than $t_{\max}$ iterations or when the total number of iterations exceeds $u_{\max}$.

Finally, to obtain a locally optimal schedule, Θ_{best} is further improved by a local search. The entire neighborhood $\mathcal{N}(\Theta_{\text{best}})$ is explored and if an improvement is possible, the neighbor with the lowest cost replaces Θ_{best}. This is repeated until a local optimum is found, which is then returned as optimization result Θ_{best}.

Algorithm OptimizeSchedule: OptimizeSchedule

```
Input: Optimization problem with $m$ groups, job data and subsequent
  retooling durations for all P-groups $i \in \boldsymbol{I}_{\mathrm{P}}$, job data of the pool of F-jobs
Output: Optimized schedule $\Theta_{\mathrm{best}}$

// initialize parameters and values
$t_{\max}$: maximum number of iterations without improvement
$u_{\max}$: total maximum number of iterations
$k_T$: number of neighbors examined for a constant $T$
$r$: parameter cooling rate
initialize current schedule $\Theta = \{\boldsymbol{\sigma}_1, \boldsymbol{\tau}_1 \ldots, \boldsymbol{\sigma}_m, \boldsymbol{\tau}_m\}$
$\Theta_{\mathrm{best}} \leftarrow \Theta$
initialize parameter annealing temperature $T$ according to (4.24)

// simulated annealing
$t \leftarrow 1$, $u \leftarrow 1$
while $t \leq t_{\max} \land u \leq u_{\max}$ do
    for $k \leftarrow 1$ to $k_T$ do
        randomly select $\Theta'$ from $\mathcal{N}(\Theta)$
        if $c(\Theta') < c(\Theta)$ then
            $\Theta \leftarrow \Theta'$
            if $c(\Theta') < c(\Theta_{\mathrm{best}})$ then
                $\Theta_{\mathrm{best}} \leftarrow \Theta'$
                $t \leftarrow 1$
            end
        else
            set $\Theta \leftarrow \Theta'$ with the probability $p(\Theta, \Theta', T) = \exp\left(\frac{c(\Theta) - c(\Theta')}{T}\right)$
        end
    end
    $T \leftarrow rT$
    $t \leftarrow t + 1$, $u \leftarrow u + 1$
end

// local search
$\Theta' \leftarrow \arg\min_{\Theta \in \mathcal{N}(\Theta_{\mathrm{best}})} c(\Theta)$
while $c(\Theta') < c(\Theta_{\mathrm{best}})$ do
    $\Theta_{\mathrm{best}} \leftarrow \Theta'$
    $\Theta' \leftarrow \arg\min_{\Theta \in \mathcal{N}(\Theta_{\mathrm{best}})} c(\Theta)$
end
return $\Theta_{\mathrm{best}}$
```

4.6. Use case

Typical scheduling problems occurring in the considered industrial plant involve 1 to 7 groups. To fill up the available retooling time between P-groups, for each F-group typically 5–10 jobs have to be selected from of a pool of 30–200 jobs. P-groups, which contain only jobs of a single diameter usually consist of up to 50 jobs. P-groups, which contain several different diameters consists of up to 200 jobs. In total, a typical schedule consists of about 40–200 jobs.

This section shows the optimal scheduling results for a practical problem from the considered industrial plant. The optimized schedule is then compared with a schedule created manually (without the aid of an algorithm) by a senior plant engineer.

4.6.1. Problem definition

The considered optimization problem consists of three P-groups and three F-groups. P-groups 1 and 5 contain jobs having several different diameters, while all jobs of P-group 3 have the same diameter. The F-groups 2, 4, and 6 share a common job pool $\boldsymbol{S}_\mathrm{F}$. The problem definition is summarized in Table 4.3.

Table 4.3.: Problem definition of the use case. Jobs of F-groups 2, 4, 6 have to be selected from a common pool with $|\boldsymbol{S}_\mathrm{F}| = 55$ jobs.

i	Type	Pool	Diameters	Jobs n_i	Retooling time d_i
1	P	$\boldsymbol{S}_{\mathrm{P},1}$	D1, D2, D3, D4	48	10 min
2	F	$\boldsymbol{S}_\mathrm{F}$	-	-	-
3	P	$\boldsymbol{S}_{\mathrm{P},3}$	D5	15	10 min
4	F	$\boldsymbol{S}_\mathrm{F}$	-	-	-
5	P	$\boldsymbol{S}_{\mathrm{P},5}$	D6, D7, D8	15	50 min
6	F	$\boldsymbol{S}_\mathrm{F}$	-	-	-

4.6.2. Optimization

For the optimization, the following rules with priorities from 1 to 4 were taken into account in the cost function (4.10):

- *Priority 1*
  - Jobs of certain grades must be arranged at the beginning of a P-group.
  - Jobs of certain grades must not be arranged at the beginning of a P-group.

  - Jobs of the same diameter must be arranged consecutively within a P-group.
  - The first job of a group must not be heated in the INDU.

- *Priority 2*
  - Jobs of same grades should be arranged consecutively within each diameter of a P-group.
  - Jobs of certain grades should not be arranged at the end of a P-group.
  - Jobs should be heated in the WBF.
- *Priority 3:* Jobs of certain grades should be arranged at the beginning of each diameter of a P-group.
- *Priority 4:* Jobs with common additionally defined characteristics should be arranged consecutively within each diameter of a P-group.

The optimization was carried out with the parameter values specified in Table 4.4. The evolution of the cost function value during the optimization process is shown in Figure 4.3. A computation time of 528 s at an Intel i5-6300U CPU @ 2.40 GHz was required for approximately 32000 iterations (equals the number of cost function evaluations) in the simulated annealing algorithm. The subsequent local search required additional 574 s for 10 iterations.

Table 4.4.: Optimization parameters.

Iterations at constant annealing temperature	k_T	30
Cooling rate	r	0.98
Max. number of iterations improvement of best solution	t_max	5000
Total maximum number of iterations	u_max	10^5
Weighting factor for unproductive time	w_s	$1\,\mathrm{s}^{-1}$
Weighting factor for exceeding of the retooling times	w_e	$10\,\mathrm{s}^{-1}$
Weighting factor for priority 1 rules	w_p1	450
Weighting factor for priority 3 rules	w_p3	10
Weighting factor for priority 2 rules	w_p2	45
Weighting factor for priority 4 rules	w_p4	5

4.6.3. Simulation results

The schedule determined by the optimization algorithm is compared to a manually created schedule, based on which the actual production took place. Figure 4.4

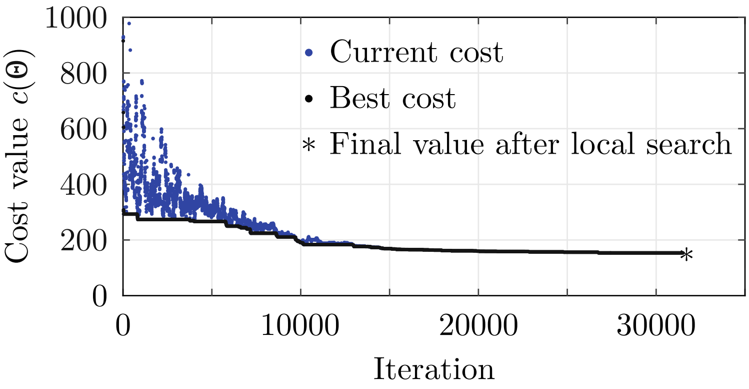

Figure 4.3.: Evolution of the cost function value c during optimization.

shows the sequences of WBF temperature intervals and final rolling speeds of the manually created schedule. Figure 4.5 shows the same data for the optimized schedule. The total processed product mass m and total process time t are given in the figures for each group.

The minimization of unproductive times in the optimization should ensure a continuous progression of the prescribed WBF temperature intervals, i. e., the intervals should overlap, to avoid required extra times at the WBF as best as possible. Jobs with significantly different WBF temperature intervals are assigned to the INDU if possible. If changes of the prescribed WBF temperature intervals cannot be avoided, the required extra times at the WBF are filled up by heating jobs in the INDU. Jobs heated in the INDU are if possible arranged such that no more than three products are heated consecutively in the INDU. For instance, in the optimized schedule, job $\sigma_{1,41}$ is not heated in the INDU because it consists of 10 individual products. During the needed extra time at the WBF before and after $\sigma_{1,41}$ jobs are heated in the INDU.

Figures 4.4 and 4.5 show that there are some changes of the final rolling speed of the profile rolling line. Because the jobs of all P-groups leave the system already after checkpoint 29, only a short time is required to empty the profile rolling line for the change of the final rolling speed. That is why these changes play a minor role in this use case.

After the last job of a P-group $i-1$, the profile rolling line (checkpoint 25) is blocked for the retooling time d_{i-1}. To evaluate the utilization of d_{i-1} by the F-group i, the time by which job $\sigma_{i+1,1}$ of the next P-group could be started earlier in case of disregarding the blocking of checkpoint 25 is measured relative to d_{i-1}. In the optimized schedule, this utilization is 95 % and 97 % for the F-groups 2 and 4, respectively, despite the relatively short retooling times $d_1 = d_3 = 10\,\text{min}$. For F-group 6, the utilization is even 99.7 %.

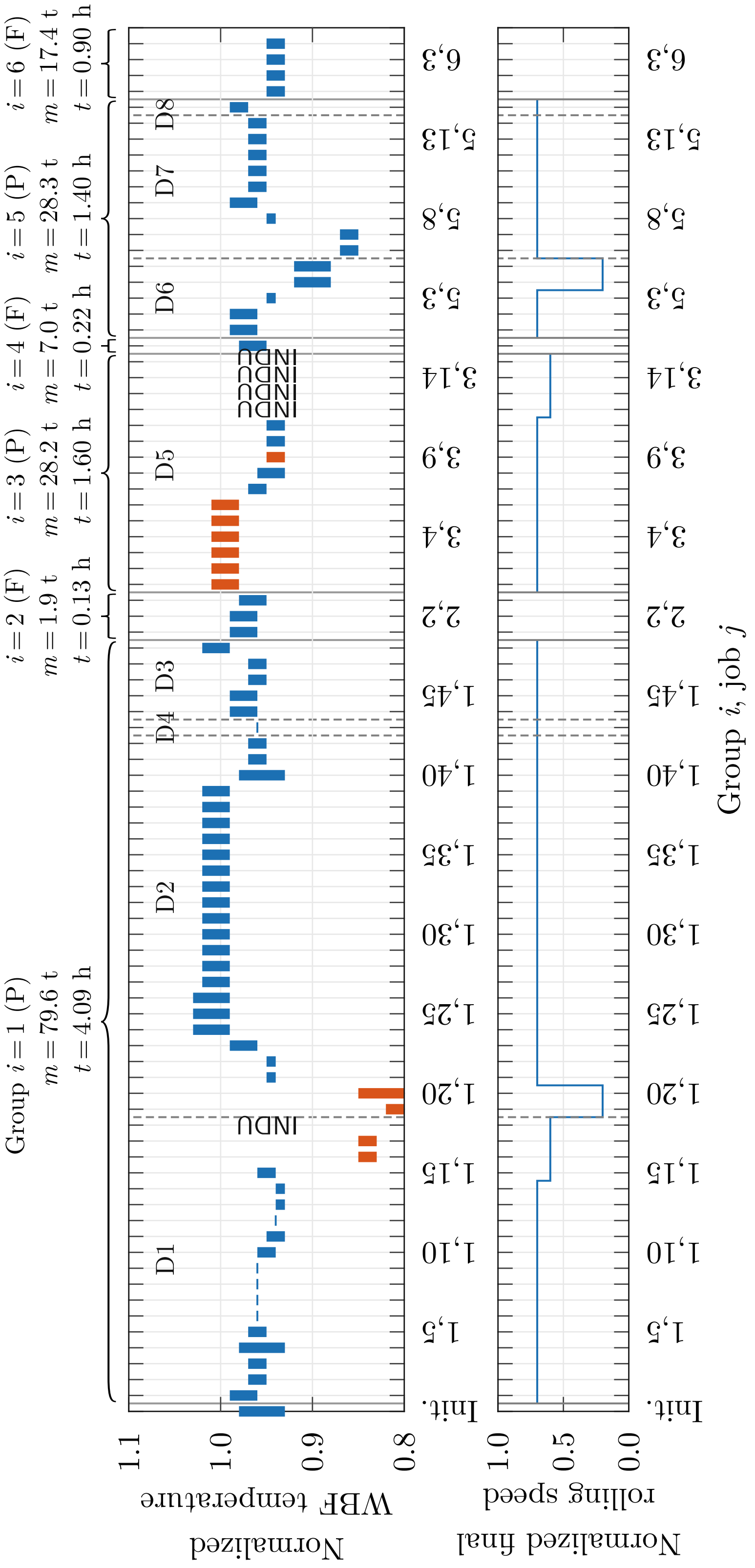

Figure 4.4.: Manually created schedule: Sequence of normalized WBF temperature intervals and final rolling speeds at the profile rolling line. 🟦 Temperature interval of a job which must be heated in the WBF. 🟧 Temperature interval of a job which is heated in the WBF but could alternatively be heated in the INDU. **INDU**: Job is heated in the INDU. | Separator between groups. ¦ Change of final diameter within a P-group.

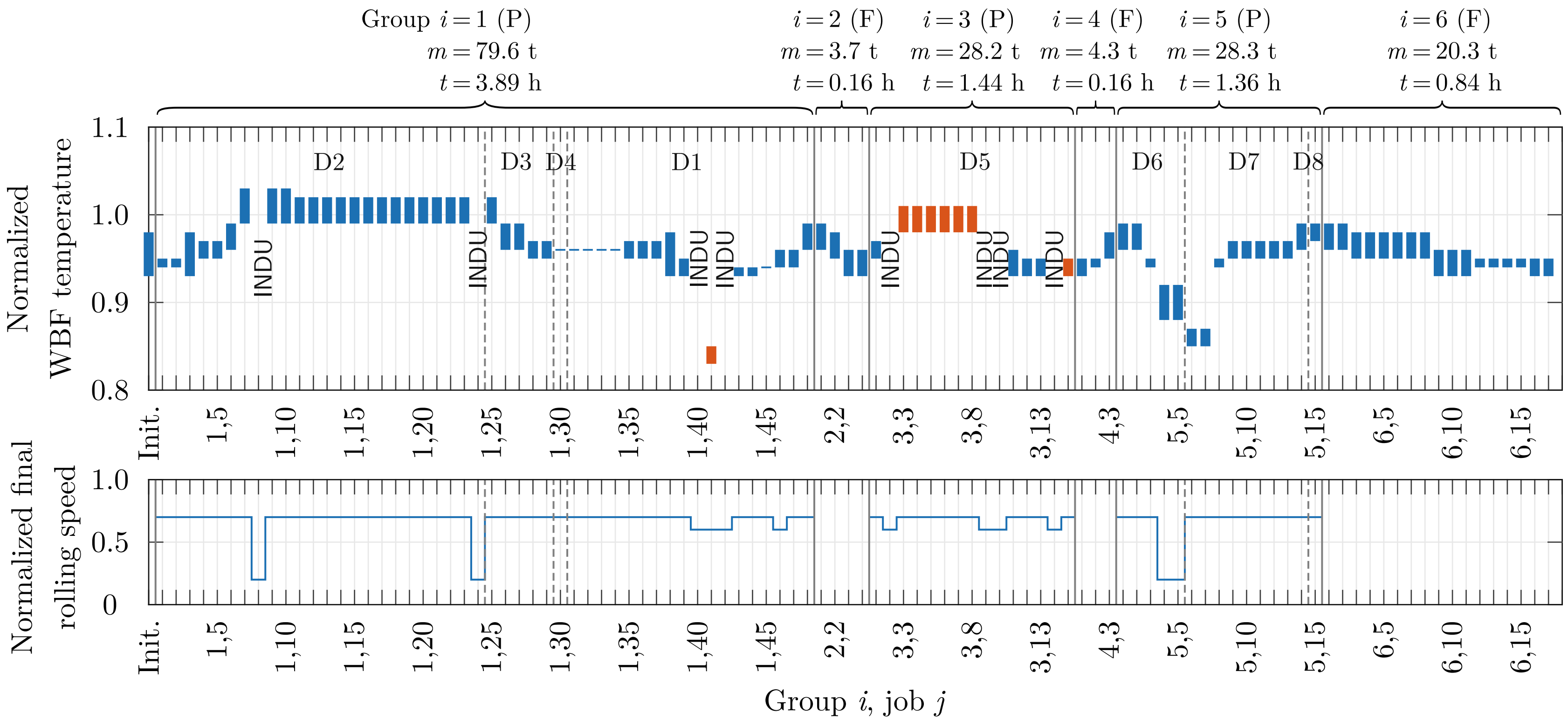

Figure 4.5.: Optimized schedule: Sequence of normalized WBF temperature intervals and final rolling speeds at the profile rolling line. 🟦 Temperature interval of a job which must be heated in the WBF. 🟥 Temperature interval of a job which is heated in the WBF but could alternatively be heated in the INDU. INDU: Job is heated in the INDU. | Separator between groups. ¦ Change of final diameter within a P-group.

Aggregate results of the schedules are compared in Table 4.5. Compared to the manual schedule, the optimized schedule contains more jobs in the F-groups. However, the total number of scheduled individual products is almost identical. The total process times are obtained from simulations of both schedules. Based on these simulation results, the optimized schedule achieves an 8 % higher throughput than the manually created schedule.

Table 4.5.: Aggregate results of the manually created and optimized schedule.

	Manual	Optimized
Number of jobs	86	102
Number of products	386	381
Total mass of products in tons	162.4	164.4
Total process time in hours	8.34	7.83
Throughput in tons/hour	19.5	21.0

4.6.4. Repeatability

Because simulated annealing is a stochastic algorithm, in general the obtained solution differs for each execution of the optimization algorithm. To assess the reliability of the obtained solutions, the optimization of the use case defined in Section 4.6.1 was repeated 100 times, always with the parameters specified in Table 4.4. The empirical frequency distribution of the obtained cost function values is shown in Figure 4.6.

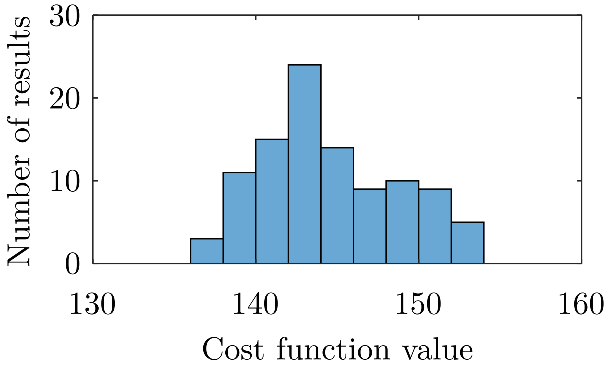

Figure 4.6.: Frequency distribution of the obtained cost function values for 100 optimizations of the use case.

The mean of the final cost value is 144.5 with a standard deviation of 4.2. The arrangement of the jobs within all groups is in all solutions very similar. The rules with priority 1 are satisfied in all cases. The number of violations of the rules with priorities 2 to 4 is in all cases almost identical. The differences in the cost function values mostly result from minor differences in the job arrangement and therefore different resulting contributions of the setup times $s_{i,j}$ in (4.10).

4.6.5. Conclusions

The suitability of the developed algorithm for optimizing problems that arise in practice was examined in this use case. In the considered scenario, the number of groups as well as the number of jobs within each P-group and contained in the pool of the F-groups is typical for scheduling tasks occurring in practice. The optimization required about 6 min of CPU time to find the final solution of the use case. This duration is acceptable for an operational use and if required allows the algorithm to be rerun regularly to adapt plans to changing conditions.

The results demonstrate that the algorithm is capable of considering all defined rules for creating job sequences while effectively minimizing setup times within and between groups. This is achieved by an appropriate arrangement of jobs, as well as the use of the INDU to fill up extra times caused by temperature changes in the WBF. If setup times due to WBF temperature changes or changes of the final rolling speed at the profile rolling line are unavoidable, the algorithm if possible arranges the jobs such that the temperature and the velocity change occur together, which causes a lower effective setup time. The jobs of the F-groups where selected and arranged such that the available retooling time between consecutive P-groups is utilized to a high degree (95 % or better). In total, a comparison with a manually created plan indicates that for the considered use case an increase of the throughput of about 8 % can be achieved.

Chapter 5.

Conclusions and outlook

This work addressed the optimal scheduling of a multi-line steel hot rolling mill of voestalpine BÖHLER Edelstahl GmbH & Co KG. The work consists of two parts: First, an algorithm was developed that calculates the earliest possible start times for a given sequence of products. Building on that, a second algorithm was designed to optimize the product selection and sequence of jobs, which are batches of similar or identical products. As a basis for both parts, a model was first developed that describes the production process of each product and job in terms of routes through the multi-line rolling plant and process times.

5.1. Model of the production process and the plant

The processing in the considered plant is characterized by individual product routes and large lengths of the products, which can occupy multiple sections of the plant at the same time. Therefore, a model consisting of an arrangement of production lines and checkpoints was developed to describe the production process. The processing of a product is described by a set of passage times of the checkpoints along the product route. The processing of a job is determined from the process simulations of its products. The model is tailored to the specific situation in the considered steel hot-rolling mill, but can be flexibly changed and adapted for other applications.

To facilitate an accurate planning of start times, the model must match the actual production times within a precision of a few seconds. The model accuracy was validated to be sufficient by comparing computed passage times of products with timestamps obtained from the material tracking system of the industrial plant.

Required (spatial and temporal) distances between jobs are ensured by blocking and releasing of appropriate checkpoints. This allows in particular the accurate consideration of indivual process times along different production routes, sequence-dependent unproductive times (e. g., required extra times at the walking beam furnace, changes of the rolling speed at the profile rolling line), retooling times of the profile rolling line, and the limited capacity of the induction furnace. To account for potential future restrictions of the production process, the model allows for the easy and flexible addition of additional checkpoint blocking and release rules.

5.2. Start time planning

Methods were developed to determine start times for products with both deterministic and stochastic process times. The start time planning under deterministic process times is used to support ongoing production and plan start times for a given sequence of jobs. Recurrent planning of the earliest possible start times for subsequent products is carried out using the latest timestamps from the material tracking system of the plant. To be practically useful, the prediction of the passing times and thus the corresponding process time models must be sufficiently accurate (errors less than a few seconds). The potential for optimizing the start times was most evident at the beginning of the first product in each job and for products heated in the induction furnace. Due to the extended required heating times in the induction furnace (compared to the typical total process time of a product), it is crucial to start a product heated in the induction furnace while a few other products heated in the walking beam furnace will be still processed. Starting the heating process too early must be avoided to prevent the product from remaining in the induction furnace beyond the prescribed heating time. Manual selection of the start times can be challenging for operators, often leading to unnecessary idle periods before the first product of a job heated in the induction furnace. Similar challenges arise when products require exceptional manufacturing steps, such as extra time for a specific cooling between rolling processes or the use of the looper at the profile rolling line. Especially in these scenarios, the start time planning tool helps to find optimal product start times, and can improve the overall throughput of the plant.

The developed start time planning algorithm was tested in the operation of the plant. In these tests, the calculated earliest possible start times of the next few products were displayed to the operator, who then initiated the start of products in the existing process control system of the plant. The feedback from the staff was positive, indicating good performance of the tool, especially under more complex production conditions as described above. A comparison of the (simulated) production using planned time intervals between product starts with the actual production showed a possible throughput increase of 14 % in terms of the number of products manufactured per unit of time. Due to these encouraging results, the start time planning is currently being integrated into the control system of the plant. Later, the start of the products will be automatically initiated.

Two different methods were presented for start time planning using stochastic process times. The start time planning under analytical calculation of the probability distributions of the random passage times is possible but computationally expensive. This is why it is only feasible for problems with a few machines and products. In contrast, the proposed Monte-Carlo-based method can be applied to much larger problem scenarios. Its advantages are its simple implementation, its low computational effort, and that it is readily transferable to similar stochastic planning problems. Based on a use-case, it was shown that the planning method considering distributions of stochastic process times can achieve a higher throughput and improve the utilization of the machines.

There is little literature on start time planning especially under stochastic process times. To the best of the author's knowledge, the start time planning under stochastic process times in the described scenario has not been solved before. Regarding the start time planning, the scientific contributions of this work are the exact solution of the stated problem as well as the development of a Monte-Carlo-based method capable of solving practical problems.

5.3. Job selection and sequence optimization

The considered scheduling problem consists of multiple groups of jobs, processed alternately at the profile and at the flat rolling line of the plant. The optimization of a group processed at the profile rolling line is similar to a traveling salesperson problem. The optimization of a group processed at the flat rolling line is similar to an orienteering problem. The primary goal of the optimization is to minimize the sum of unproductive setup times between the jobs. An additional objective for groups processed at the flat rolling line is that the given retooling time at the profile rolling line is not exceeded. Secondary goals are to adhere to all specified rules for the processing of a job sequence.

Some methods for the exact optimization of individual groups were examined. It was shown that exact solutions can be obtained for both types of groups. However, to facilitate these exact solutions, it is necessary to simplify the optimization problem (in particular, only the minimization of the unproductive setup times is possible). Moreover, the optimizations cause a high computational effort even for moderate group sizes.

Because of dependencies between the groups (e. g., unproductive times of jobs between subsequent groups, a common job pool for groups processed at the flat rolling line), an optimization algorithm has to optimize all groups of a problem together. To do so, a heuristic simulated annealing algorithm followed by a local search was developed. A cost function was designed for the optimization problem that takes both unproductive times and violations of rules with different priorities into account. The simple structure of the cost function allows the user to prioritize the specified rules differently. The optimization results are explainable by a breakdown of the costs (e. g., the output of which defined rules are violated between which jobs). If the solutions are not satisfactory or the requirements change, the individual weights of the rules can be tuned by the user. It is also possible to define additional rules and incorporate them into the cost function without having to change the main procedure of the optimization algorithm.

The optimization of F- and P-groups together introduces multiple possible permutation moves resulting in a complex neighborhood of solutions. For a reduced computational effort, the fact that simulated annealing does not examine the entire neighborhood but only a single neighboring solution in each iteration was exploited. This still allows a sufficiently fast improvement of the cost function value in the considered optimization problem. The subsequent local search improves the best found solution by simulated annealing (typically by a few percent) and yields a locally optimal schedule.

Experienced plant operators require for the manual creation of production plans of a size comparable to the use case considered in this work at least a few hours. In contrast, the developed algorithm returns a result within a few minutes. Even if currently the result of the algorithm has to be manually checked by the plant operators, the optimization algorithm leads to a significant time saving when creating the schedules. In addition, optimized schedules effectively minimize unproductive times and satisfy all specified rules. Compared to manually created schedules, the use case showed an increase of the total product throughput by approximately 8 %.

To the best of the author's knowledge, the optimization problem involving multiple job groups (similar to a combination of several traveling salesperson and orienteering problems) has not previously been addressed in the literature. Conventional solvers for combinatorial optimization cannot be directly applied to this specific problem. The main scientific contributions of this thesis are as follows:

- A cost function was developed to minimize unwanted setup times and to punish rule violations concerning the arrangement of jobs. The cost function can be flexibly adjusted.
- Specialized move operators for individual groups within the schedule were defined. While some move operators, such as the 3-opt move, are common, others were specifically designed for this problem (e. g., change-furnace). Neighborhood functions that are applied to schedules consisting of multiple production groups were created.

- An algorithm capable of optimizing the job sequences of multiple groups was developed. Using the defined cost function and the neighborhood operators, the heuristic algorithm can reliable and quickly solve practical optimization problems with consistent quality.

5.4. Outlook

Currently, the start time planning algorithm is used as a decision support tool for plant operators in the sense that the operator is presented the result of the algorithm and then manually starts the products. For further improvement, it is planned to automate the product starts based on the developed start time planning algorithm. In this way, the individual product lengths of each product can automatically and accurately be considered.

The currently created schedules include production lots that typically require one to two days for processing. In a future work, it may be explored whether the optimization of larger periods can further increase the product throughput. In addition, the optimization of the sequence of groups processed at the profile rolling line could be further investigated. This could lead to a further increase of the product throughput, as well as positive effects if for example inventory costs and logistical requirements are taken into account. However, implementing such changes of the operation in the plant would require considerable logistical effort.

Appendix A.

Process time models

In the following, the process times $p_i^{\text{in},k,l}$ and $p_i^{\text{thru},k,l}$ for all sections between checkpoints k and l that can follow one another in product routes as described in Section 2.2 are defined. The corresponding topological model of the plant is shown in Figure 2.4 (page 17).

A.1. Walking beam furnace

A.1.1. Checkpoints 1–2

At the requested start time of a product, its head is located at checkpoint 1. Initially, the product is lifted onto the exit rolling conveyor within the WBF. The furnace door is then opened, and the product is ejected from the WBF.

In the model, only the transit time until the product head reaches the WBF exit at checkpoint 2 is relevant. The entry time is

$$p_i^{\text{in},1,2} = 0 \ . \tag{A.1}$$

The transit time is independent of the product

$$p_i^{\text{thru},1,2} = p_{\text{WBF,out}} = \text{const.} \tag{A.2}$$

A.1.2. Checkpoints 2–12

A product of the length l_{pm} is ejected from the WBF with the speed v_{W} and transported through a scale breaker. After the product has been completely ejected (tail at checkpoint 2), it is transported with the speed v_{T} by the rolling conveyor to checkpoint 12. The corresponding entry time is

$$p_i^{\text{in},2,12} = \frac{l_{\text{pm}}}{v_{\text{W}}} \ . \tag{A.3}$$

The distance between the checkpoints 2 and 12 is $d_{2,12} > l_{\mathrm{pm}}$. The transit time is

$$p_i^{\mathrm{thru},2,12} = \frac{d_{2,12} - l_{\mathrm{pm}}}{v_{\mathrm{T}}} > 0 \; . \tag{A.4}$$

A.2. Induction furnace

A.2.1. Checkpoints 3–4

At the start time of a product, it is located in a loading area. In the model, this means that the product head is at checkpoint 3. Upon request, the product is transported from the loading area to one of the three INDU chambers by crane 1. Independent of the chamber, the model uses the same transport time p_{crane1}. The crane uploads the product on the rolling conveyor in front of the furnace with the product head at checkpoint 4.

The product passes checkpoint 3 as it is picked up by crane 1 and the corresponding entry time is

$$p_i^{\mathrm{in},3,4} = 0 \; . \tag{A.5}$$

The transit time is

$$p^{\mathrm{thru},3,4} = p_{\mathrm{crane1}} \; . \tag{A.6}$$

A.2.2. Checkpoints 4–5, 4–7, 4–9

Checkpoint 4 was introduced into the model to enable modeling of the transport process of crane 1 under consideration of possible delays due to movements of crane 2. These delays can be captured by appropriate blocking of the checkpoints 4 and 11. The sections 4–5, 4–7, and 4–9 have no pyhsical lengths. The process times are defined as

$$p_i^{\mathrm{in},4,5} = p_i^{\mathrm{in},4,7} = p_i^{\mathrm{in},4,9} = 0 \; , \tag{A.7}$$

$$p_i^{\mathrm{thru},4,5} = p_i^{\mathrm{thru},4,7} = p_i^{\mathrm{thru},4,9} = 0 \; . \tag{A.8}$$

A.2.3. Checkpoints 5–6, 7–8, 9–10

The modeled process times are identical for all three INDU chambers. The time during which the product is pushed into the respective chamber is $p_{\mathrm{I,load}}$ and is independent of the product length l_{pm}. The heating time $p_{i,\mathrm{heat}}$ generally depends on the product σ_i. The process times are

$$p_i^{\text{in},5,6} = p_i^{\text{in},7,8} = p_i^{\text{in},9,10} = p_{\text{I,load}} \;, \tag{A.9}$$

$$p_i^{\text{thru},5,6} = p_i^{\text{thru},7,8} = p_i^{\text{thru},9,10} = p_{i,\text{heat}} \;. \tag{A.10}$$

A.2.4. Checkpoints 6–11, 8–11, 10–11

The hot product is ejected from the INDU chamber during a product independent time $p_{\text{I,unload}}$. After that, the product head is located at checkpoint 11, where it waits to be transported by INDU crane 2. The process times are

$$p_i^{\text{in},6,11} = p_i^{\text{in},8,11} = p_i^{\text{in},10,11} = p_{\text{I,unload}} \;, \tag{A.11}$$

$$p_i^{\text{thru},6,11} = p_i^{\text{thru},8,11} = p_i^{\text{thru},10,11} = 0 \;. \tag{A.12}$$

A.2.5. Checkpoints 11–12

The product is picked up by crane 2 and transported to the rolling conveyor in front of the reversing rolling stand. The transport time is p_{crane2} for any product and any INDU chamber. After the product is unloaded, the product head is positioned at checkpoint 12.

The product passes checkpoint 11 as it is picked up by crane 2 and the corresponding entry time is

$$p_i^{\text{in},11,12} = 0 \;. \tag{A.13}$$

The transit time is

$$p_i^{\text{thru},11,12} = p_{\text{crane2}} \;. \tag{A.14}$$

A.3. Reversing rolling stand

A product is rolled out in n passes at the reversing rolling stand. The process for sequential rolling passes is outlined in Figure A.1.

A.3.1. Checkpoints 12–13

The product of the length l_{pm} is transported on the roller conveyor with the speed v_{T} to the reversing rolling stand. The entry time is

$$p_i^{\text{in},12,13} = \frac{l_{\text{pm}}}{v_{\text{T}}} \;. \tag{A.15}$$

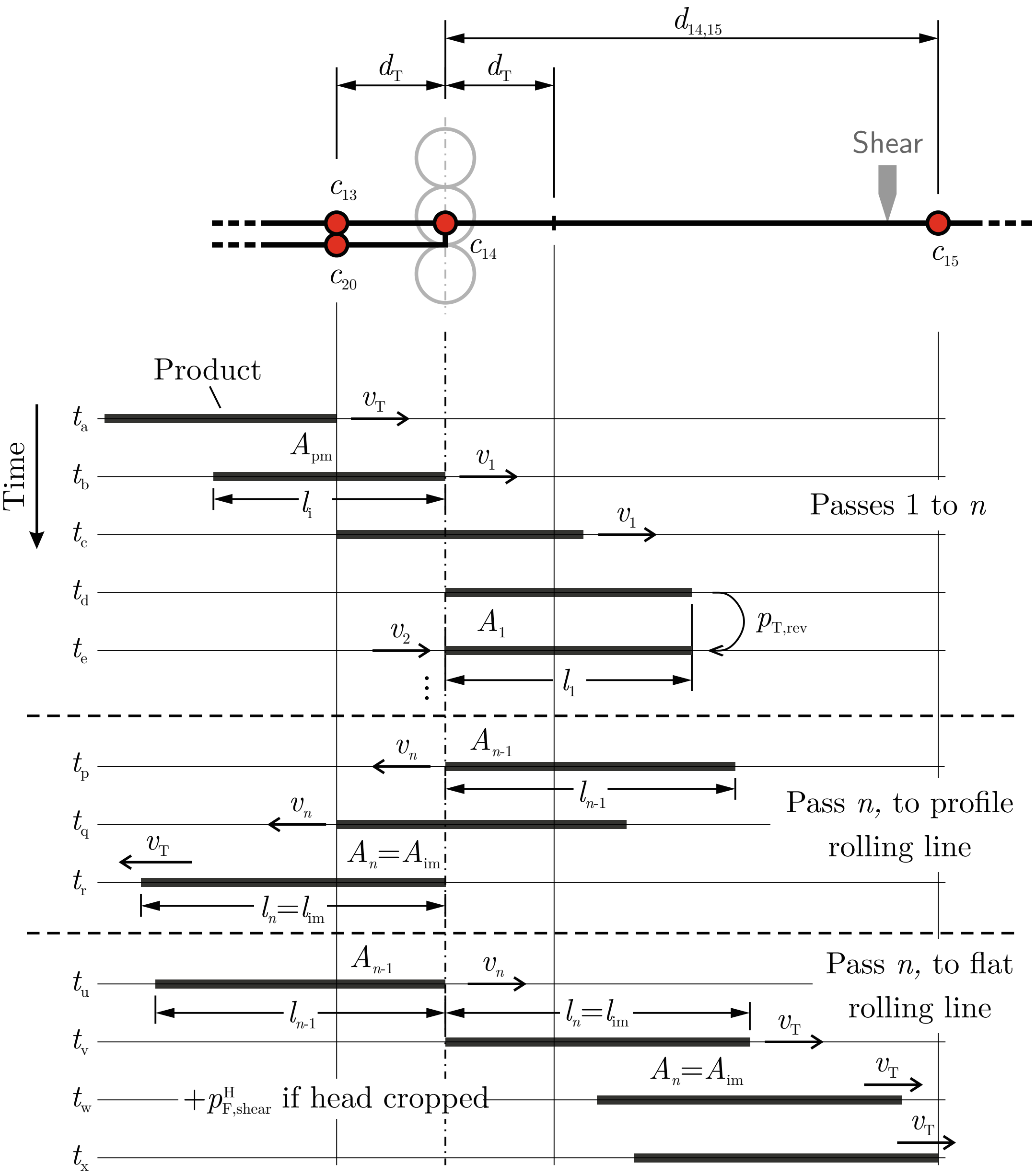

Figure A.1.: Processing a product at the reversing rolling stand.

The distance between the checkpoints 12 and 13 is $d_{12,13} > l_{\mathrm{pm}}$ and the corresponding transit time is

$$p_i^{\mathrm{thru},12,13} = \frac{d_{12,13}}{v_{\mathrm{T}}} - p_i^{\mathrm{in},12,13} > 0 \ . \tag{A.16}$$

A.3.2. Checkpoints 13–14

This section represents the first $n-1$ passes at the reversing rolling stand, illustrated by the time interval $[t_a, t_e]$ in Figure A.1. The head of the product reaches checkpoint 13 at the time t_a. Afterwards the product is conveyed to the stand with the velocity v_{T} until the head reaches the rolls at the time t_b.

The product is then rolled out, the exit velocities v_j and the cross-sectional areas A_j after each pass $j = 1, \dots, n$ are defined in a product-specific pass schedule. After the last pass, the cross-sectional area is $A_j = A_{\mathrm{im}}$. The time between two consecutive passes is $p_{\mathrm{T,rev}}$. During this time (e. g. time interval $[t_d, t_e]$ between pass 1 and 2), the product is outside the roll gap. Immediately before the last pass n the head of the product is located at checkpoint 14.

With the infeed speed at the first pass

$$v_0 = \frac{A_1}{A_{\mathrm{pm}}} v_1 \ , \tag{A.17}$$

the entry time is

$$p_i^{\mathrm{in},13,14} = t_c - t_a = \frac{d_{\mathrm{T}}}{v_{\mathrm{T}}} + \frac{l_{\mathrm{pm}} - d_{\mathrm{T}}}{v_0} \ . \tag{A.18}$$

With the product length after pass j,

$$l_j = \frac{A_{\mathrm{pm}}}{A_j} l_{\mathrm{pm}} \ , \tag{A.19}$$

the transit time ($t_p - t_c$ if the product is processed at the profil rolling line, $t_u - t_c$ if the product is processed at the flat rolling line) is

$$p_i^{\mathrm{thru},13,14} = \frac{d_{\mathrm{T}}}{v_{\mathrm{T}}} + \sum_{j=1}^{n-1} \frac{l_j}{v_j} + (n-1) p_{\mathrm{T,rev}} - p_i^{\mathrm{in},13,14} > 0 \ . \tag{A.20}$$

A.3.3. Checkpoints 14–15

This section represents the last pass n at the reversing rolling stand and the transport of a product in the direction of the flat rolling line, as shown in Figure A.1.

At the beginning of the last pass n, the head of the product is located at checkpoint 14. During the last pass, the exit velocity of the product is v_n. After the last pass the length of the intermediate product is l_{im}. Once the product tail has left the mill stand, the product is transported towards the flat rolling line with the speed v_{T}.

A shear is located upstream of checkpoint 15. It is used to crop the head or both the head and the tail of a product. Cropping the head is considered in the process times between the checkpoints 14 and 15, while cropping the tail is considered in the process times between the checkpoints 16 and 17. Cropping of the head requires the extra time $p^{\mathrm{H}}_{\mathrm{F,shear}}$ (independent of the product). The reduction of the product length l_{im} due to cropping is neglected.

The entry time is

$$p_i^{\mathrm{in},14,15} = t_v - t_u = \frac{l_{\mathrm{im}}}{v_n} \ . \tag{A.21}$$

The transit time depends on whether the head of the product is cropped, i. e.,

$$p_i^{\mathrm{thru},14,15} = t_x - t_v = \begin{cases} \frac{d_{14,15}-l_{\mathrm{im}}}{v_{\mathrm{T}}} \ , & \text{if the product head is not cropped,} \\ \frac{d_{14,15}-l_{\mathrm{im}}}{v_{\mathrm{T}}} + p^{\mathrm{H}}_{\mathrm{F,shear}} \ , & \text{is the product head is cropped.} \end{cases} \tag{A.22}$$

Because $l_{\mathrm{im}} < d_{14,15}$, the transit time is $p_i^{\mathrm{thru},14,15}$ is always positive.

A.3.4. Checkpoints 14–20

This section represents the last pass n at the reversing rolling stand and the transport of a product in the direction of the profile rolling line, as shown in Figure A.1.

At the beginning of the last pass n, the head of the product is located at checkpoint 14 (t_p). During the last pass, the product leaves the stand with the velocity v_n specified in the pass schedule. After the last pass, the product has the length $l_n = l_{\mathrm{im}}$. The process times of the section are

$$p_i^{\mathrm{in},14,20} = t_r - t_p = \frac{l_{\mathrm{im}}}{v_n} \ , \tag{A.23}$$

$$p_i^{\mathrm{thru},14,20} = t_q - t_r = \frac{d_{\mathrm{T}}}{v_n} - p_i^{\mathrm{in},14,20} < 0 \ . \tag{A.24}$$

A.3.5. Checkpoints 20–21

While the product tail is still in the mill stand, the product is transported with the velocity v_n specified by the pass plan. Afterwards, it is transported with the velocity v_{T} of the roller table towards checkpoint 21.

The entry time is

$$p_i^{\mathrm{in},20,21} = \frac{l_{\mathrm{im}} - d_{\mathrm{T}}}{v_n} + \frac{d_{\mathrm{T}}}{v_{\mathrm{T}}} \ . \tag{A.25}$$

The distance between the checkpoints 20 and 21 is $d_{20,21} = d_{12,13}$. The time which is required for the head to move between these checkpoints is

$$p_i^{\text{head},20,21} = \begin{cases} \frac{l_{\text{im}} - d_{\text{T}}}{v_n} + \frac{d_{\text{T}} + d_{20,21} - l_{\text{im}}}{v_{\text{T}}} & \text{if } l_{\text{im}} \leq d_{\text{T}} + d_{20,21} \ , \\ \frac{d_{20,21}}{v_n} & \text{if } l_{\text{im}} > d_{\text{T}} + d_{20,21} \ . \end{cases} \tag{A.26}$$

The transit time is then

$$p_i^{\text{thru},20,21} = p_i^{\text{head},20,21} - p_i^{\text{in},20,21} \ . \tag{A.27}$$

A.4. Flat rolling line

The processing of a product after it has reached the flat rolling line is shown in Figure A.2.

A.4.1. Checkpoints 15–16

The intermediate product has the length l_{im} while it passes checkpoint 15. The product tail passes a shear shortly before checkpoint 15. If the tail is cropped, this requires the extra time $p^{\text{T}}_{\text{F,shear}}$ (independent of the product). After the tail has passed checkpoint 15, the product is transported with the speed v_F.

The entry time is

$$p_i^{\text{in},15,16} = \begin{cases} \frac{l_{\text{im}}}{v_{\text{T}}} \ , & \text{if the product tail is not cropped,} \\ \frac{l_{\text{im}}}{v_{\text{T}}} + p^{\text{T}}_{\text{F,shear}} \ , & \text{is the product tail is cropped.} \end{cases} \tag{A.28}$$

The transit time is

$$p_i^{\text{thru},15,16} = \frac{d_{15,16} - l_{\text{im}}}{v_{\text{F}}} \ , \tag{A.29}$$

where $d_{15,16}$ is the distance between checkpoints 15 and 16.

A.4.2. Checkpoints 16–17

At the flat rolling line, the product is rolled in three passes and runs through all five stands in each pass. This section represents the first and parts of the seconds pass, as shown for the time interval $[t_a, t_d]$ in Figure A.2. At the position of checkpoint 17, there is a probe to detect the position of the product during the rolling process.

The distance between two consecutive stands is d_{s} throughout the tandem rolling mill. The distance between checkpoint 16 and stand 1 as well as between stand 5

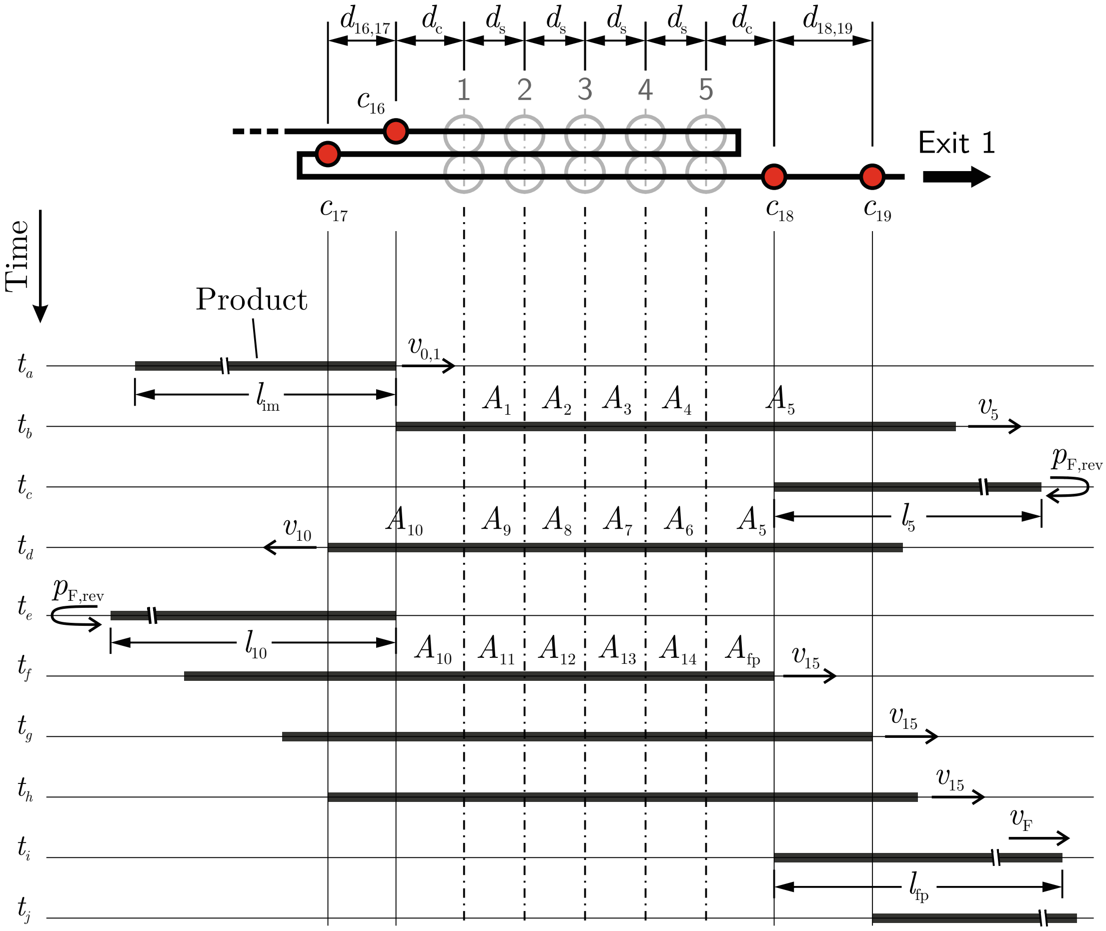

Figure A.2.: Processing a product at the flat rolling line.

and checkpoint 18 is d_c. After each pass, the product is transported until the product tail reaches checkpoints 18 or 16. A product-independent time span $p_{F,\mathrm{rev}}$ is required for the product to change its motion direction and to reach again the first stand of the flat rolling line for the next pass.

Before the first pass, the product has the length l_im and the cross-sectional area A_im. The cross-sectional areas A_j, $j = 1, \dots, 15$ after each stand (where $A_{15} = A_\mathrm{fp}$) and the exit velocities v_5, v_{10}, v_{15} after each pass are defined in the product-specific pass plan. The product lengths are

$$l_5 = \frac{A_\mathrm{im}}{A_5} l_\mathrm{im} \qquad \text{after the first pass,} \tag{A.30}$$

$$l_{10} = \frac{A_\mathrm{im}}{A_{10}} l_\mathrm{im} \qquad \text{after the second pass.} \tag{A.31}$$

The entry velocities into the first stand of the respective side are

$$v_{0,1} = \frac{A_5}{A_\mathrm{im}} v_5 \qquad \text{at the first pass,} \tag{A.32}$$

$$v_{0,2} = \frac{A_{10}}{A_5} v_{10} \qquad \text{at the second pass,} \tag{A.33}$$

$$v_{0,3} = \frac{A_\mathrm{fp}}{A_{10}} v_{15} \qquad \text{at the third pass} \tag{A.34}$$

The exit velocities at each stand are

$$v_j = \frac{A_5}{A_j} v_5 \,, \qquad j = 1, \dots, 4 \,, \qquad \text{at the first pass,} \tag{A.35}$$

$$v_j = \frac{A_{10}}{A_j} v_{10} \,, \qquad j = 6, \dots, 9 \,, \qquad \text{at the second pass,} \tag{A.36}$$

$$v_j = \frac{A_\mathrm{fp}}{A_j} v_{15} \,, \qquad j = 11, \dots, 14 \,, \qquad \text{at the third pass.} \tag{A.37}$$

Before the first pass, the product is transported from checkpoint 16 to stand 1 with the speed $v_{0,1}$. The entry time is thus

$$p_i^{\mathrm{in},16,17} = t_b - t_a = \frac{l_\mathrm{im}}{v_{0,1}} \,. \tag{A.38}$$

The head of the product reaches checkpoint 17 after it left stand 1 at the second pass (t_d). The transit time consists of the time for the product tail to reach the position of checkpoint 18 (t_c), the reversing time $p_{\mathrm{F,rev}}$, and the time until the product head reaches checkpoint 17 (t_d). Note that checkpoint 18 is not part of this section.

$$p_i^{\text{thru},16,17} = t_d - t_b = \frac{d_\text{c}}{v_{0,1}} + \sum_{j=1}^{4} \frac{d_\text{s}}{v_j} + \frac{d_\text{c}}{v_5} + p_{\text{F,rev}} + \frac{d_\text{c}}{v_{0,2}} + \sum_{j=6}^{9} \frac{d_\text{s}}{v_j} + \frac{d_\text{c} + d_{16,17}}{v_{10}} > 0 \quad \text{(A.39)}$$

Observations reveal random fluctuations of the in the reversing time $p_{\text{F,rev}}$. These fluctuations occur because the products are sliding slightly across the roller table. Due to the lack of manual manipulations, the fluctuations are less significant than those of the reversing time at the reversing rolling stand. For the process model described here, the fluctuations of $p_{\text{F,rev}}$ are neglected and a deterministic value is used.

A.4.3. Checkpoints 17–18

This section represents the remaining part of the second pass and the third pass at the flat rolling line. The section starts with the product moving in the direction of the reversing rolling stand when the head is located at checkpoint 17 (t_d in Figure A.2). As the product tail reaches checkpoint 16 (t_e), the direction of movement is reversed, which takes the time $p_{\text{F,rev}}$. Afterwards, the third pass is carried out. Note that checkpoint 16 is not part of this section.

The entry time consists of the time until the product tail reaches the position of checkpoint 16, the reversing time $p_{\text{F,rev}}$, and the time until the product tail (after reversing the direction of movement) reaches checkpoint 17 (t_h). The entry time is thus

$$p_i^{\text{in},17,18} = t_h - t_d = \frac{l_{10} - d_{16,17}}{v_{10}} + p_{\text{F,rev}} + \frac{l_{10} - d_{16,17}}{v_{0,3}} \, . \quad \text{(A.40)}$$

To calculate the transit time, we first calculate the time $p_i^{\text{head},17,18}$ that the head of the product needs to move from checkpoint 17 in the second pass to checkpoint 18 in the third pass,

$$p_i^{\text{head},17,18} = t_f - t_d = \frac{l_{10} - d_{16,17}}{v_{10}} + p_{\text{F,rev}} + \frac{d_\text{c}}{v_{0,3}} + \sum_{j=11}^{14} \frac{d_\text{s}}{v_j} + \frac{d_\text{c}}{v_{15}} \, . \quad \text{(A.41)}$$

The transit time is

$$p_i^{\text{thru},17,18} = t_f - t_h = p_i^{\text{head},17,18} - p_i^{\text{in},17,18} < 0 \, . \quad \text{(A.42)}$$

A.4.4. Checkpoints 18–19

The product with the length l_{fp} passes the checkpoint 18 with the speed v_{15} as shown in Figure A.2. The process times of the section are

$$p_i^{\mathrm{in},18,19} = t_i - t_f = \frac{l_{\mathrm{fp}}}{v_{15}} \;, \tag{A.43}$$

$$p_i^{\mathrm{thru},18,19} = t_g - t_i = \frac{d_{18,19} - l_{\mathrm{fp}}}{v_{15}} < 0 \;. \tag{A.44}$$

A.4.5. Checkpoint 19 → Exit 1

The product with the length l_{fp} arrives at checkpoint 19 with the speed v_{15} as indicated in Figure A.2. As soon as the product tail has passed the checkpoint 18, the product is moved with the speed v_{F} by the roller table. The entry time is

$$p_i^{\mathrm{in},19,\mathrm{out}} = t_j - t_g = \frac{l_{\mathrm{fp}} - d_{18,19}}{v_{15}} + \frac{d_{18,19}}{v_{\mathrm{F}}} \;. \tag{A.45}$$

When the product tail has passed the checkpoint 19, the product leaves the system at Exit 1. Therefore, a transit time does not exist.

A.5. Cross conveyor

A.5.1. Checkpoints 12–22

The product with the length l_{pm} is transported by the cross conveyor from the roller table after the WBF to the roller table next to the heating box, which takes the product independent time p_{cc}. The process times are

$$p_i^{\mathrm{in},12,22} = 0 \;, \tag{A.46}$$

$$p_i^{\mathrm{thru},12,22} = p_{\mathrm{cc}} \;. \tag{A.47}$$

A.5.2. Checkpoints 21–22

Depending on the length l_{im} of the intermediate product, it may pass checkpoint 21 partially with the exit velocity v_n of the last pass at the reversing rolling stand. After that, the product moves with the velocity v_{T} of the roller table. After the tail of the product comes to rest at checkpoint 21, the product is picked up by the cross conveyor and transported to the roller table in front of the heating box, which takes the product independent time p_{CC}.

The entry time is

$$p_i^{\mathrm{in},21,22} = \begin{cases} \frac{l_{\mathrm{im}}}{v_{\mathrm{T}}} & \text{if } l_{\mathrm{im}} \leq d_{\mathrm{T}} + d_{20,21} \ , \\ \frac{l_{\mathrm{im}} - d_{\mathrm{T}} - d_{20,21}}{v_n} + \frac{d_{\mathrm{T}} + d_{20,21}}{v_{\mathrm{T}}} & \text{if } l_{\mathrm{im}} > d_{\mathrm{T}} + d_{20,21} \ . \end{cases} \tag{A.48}$$

The transit time is

$$p_i^{\mathrm{thru},21,22} = p_{\mathrm{CC}} \ . \tag{A.49}$$

A.6. Heating box

A.6.1. Checkpoints 22–23

The product of length l_{im} is transported into the heating box with the velocity $v_{\mathrm{HB,in}}$. The entry time is thus

$$p_i^{\mathrm{in},22,23} = \frac{l_{\mathrm{im}}}{v_{\mathrm{HB,in}}} \ . \tag{A.50}$$

The distance between checkpoints 22 and 23 is $d_{22,23}$. The transit time is

$$p_i^{\mathrm{thru},22,23} = \frac{d_{22,23} - l_{\mathrm{im}}}{v_{\mathrm{HB,in}}} > 0 \ . \tag{A.51}$$

This is the minimum possible time for a product to stay inside the heating box. A longer duration may be chosen during the start time planning of the products. In this case, the model assumes that the product head remains at checkpoint 23 during the extra heating time.

A.6.2. Checkpoints 23–24

After the product head passes checkpoint 23, it passes the checkpoint 24 and then leaves the heating box. Between checkpoints 23 and 25, the product has the velocity $v_{\mathrm{HB,out}}$. Between checkpoints 25 and 26 it has the velocity $v_{\mathrm{P,in}}$. After the product head has reached stand 1 of the profile rolling line (checkpoint 26), it is fed into stand 1 with the velocity $v_{\mathrm{P,in1}}$.

If the head of the product is cropped at the shear after checkpoint 24, this requires the additional time $p_{\mathrm{P,shear}}$, which is independent of the product. The shortening of the product length l_{im} due to cropping can be neglected.

With the distance $d_{k,l}$ between two checkpoints k and l, the entry time is

$$p_i^{\text{in},23,24} = \begin{cases} \frac{d_{23,25}}{v_{\text{HB,out}}} + \frac{d_{25,26}}{v_{\text{P,in}}} + \frac{l_{\text{im}}-d_{23,26}}{v_{\text{P,in1}}} & \text{if the product head is not cropped,} \\ \frac{d_{23,25}}{v_{\text{HB,out}}} + \frac{d_{25,26}}{v_{\text{P,in}}} + \frac{l_{\text{im}}-d_{23,26}}{v_{\text{P,in1}}} + p_{\text{P,shear}} & \text{if the product head is cropped.} \end{cases} \tag{A.52}$$

With the time

$$p_i^{\text{head},23,24} = \frac{d_{23,24}}{v_{\text{HB,out}}} \ , \tag{A.53}$$

the transit time is

$$p_i^{\text{thru},23,24} = p_i^{\text{head},23,24} - p_i^{\text{in},23,24} < 0 \ . \tag{A.54}$$

A.6.3. Checkpoints 24–25

The entry time depends on whether the product head is cropped at the shear before checkpoint 25.

$$p_i^{\text{in},24,25} = \begin{cases} \frac{d_{24,25}}{v_{\text{HB,out}}} + \frac{d_{25,26}}{v_{\text{P,in}}} + \frac{l_{\text{im}}-d_{24,26}}{v_{\text{P,in1}}} & \text{if the head is not cropped,} \\ \frac{d_{24,25}}{v_{\text{HB,out}}} + \frac{d_{25,26}}{v_{\text{P,in}}} + \frac{l_{\text{im}}-d_{24,26}}{v_{\text{P,in1}}} + p_{\text{P,shear}} & \text{if the head is cropped.} \end{cases} \tag{A.55}$$

With the time

$$p_i^{\text{head},24,25} = \begin{cases} \frac{d_{24,25}}{v_{\text{HB,out}}} & \text{if the head is not cropped,} \\ \frac{d_{24,25}}{v_{\text{HB,out}}} + p_{\text{P,shear}} & \text{if the head is cropped,} \end{cases} \tag{A.56}$$

the transit time is

$$p_i^{\text{thru},24,25} = p_i^{\text{head},24,25} - p_i^{\text{in},24,25} < 0 \ . \tag{A.57}$$

A.7. Profile rolling line

A.7.1. Checkpoints 25–26

The product is transported with the velocity $v_{\text{P,in}}$ until the product head reaches mill stand 1 of the profile rolling line. The product is then fed into stand 1 with the velocity $v_{\text{P,in1}}$. Before the product tail passes checkpoint 25, it may be cropped at the shear. This however is done without stopping the product (flying shear), so that it does not affect process times. Also the reduction of the product length due to cropping can be neglected.

The process times are

$$p_i^{\text{in},25,26} = \frac{d_{25,26}}{v_{\text{P,in}}} + \frac{l_{\text{im}} - d_{25,26}}{v_{\text{P,in1}}} \;, \tag{A.58}$$

$$p_i^{\text{thru},25,26} = \frac{d_{25,26} - l_{\text{im}}}{v_{\text{P,in1}}} < 0 \;. \tag{A.59}$$

A.7.2. Checkpoints 26–27, . . . , 30–31, 29 → Exit 2

Before entering mill stand 1 of the profile rolling line, the product has the length l_{im} and the cross-sectional area A_{im}. After the product head has reached stand 1, the product is fed into stand 1 with the velocity $v_{\text{P,in1}}$. The cross-sectional area A_k of the product after each stand, i. e., after each checkpoint $k = 26, \ldots, 31$, is defined in the product specific pass plan.

The entry time

$$p_i^{\text{in},k,k+1} = \frac{l_{\text{im}}}{v_{\text{P,in1}}} \;, \qquad k = 26, \ldots, 30 \;, \tag{A.60}$$

is identical for the stands 1 to 6. The velocity after the checkpoint k is

$$v_k = \frac{A_{\text{im}}}{A_k} v_{\text{P,in1}} \;, \qquad k = 26, \ldots, 31 \;. \tag{A.61}$$

With the distance $d_{k,k+1}$ between the checkpoints k and $k+1$, the transit time is

$$p_i^{\text{thru},k,k+1} = \frac{d_{k,k+1}}{v_k} - p_i^{\text{in},k,k+1} < 0 \;, \qquad k = 26, \ldots, 30 \;. \tag{A.62}$$

If the product leaves the system at Exit 2 after checkpoint 29, the last required process time is $p_i^{\text{in},29,30}$.

A.7.3. Checkpoints 31–32

The product exits stand 6 (checkpoint 31) of the profile rolling line with the velocity v_{31} according to (A.61) and moves to the shear at checkpoint 32. At this shear, cropping does not affect process times. Moreover, a reduction of the product length due to cropping can be neglected.

The entry time $p_i^{\text{in},30,31}$ is equal to (A.60). The transit time is

$$p_i^{\text{thru},31,32} = \frac{d_{31,32}}{v_{31}} - p_i^{\text{in},30,31} < 0 \;. \tag{A.63}$$

A.7.4. Checkpoints 32–33, 32 → Exit 3

The product head moves from the shear at checkpoint 32 (cropping does not affect process times, a reduction of the product length due to cropping can be neglected) via the looper table to checkpoint 33 (position of a probe). Between the checkpoints 32 and 33 the head has to travel (before a loop has formed) the distance $d_{32,33}$ with the velocity v_{31} equal to (A.61). The cross-sectional area A_{31} of the product after stand 6 is specified in the product specific pass schedule.

The entry time of the section is equal to (A.60). If the product does not leave the system at Exit 3, the transit time $p_i^{\text{thru},32,33}$ is required. With the product length

$$l_{31} = \frac{A_{\text{im}}}{A_{31}} l_{\text{im}} \tag{A.64}$$

after stand 6, the transit time follows in the form

$$p_i^{\text{thru},32,33} = \frac{d_{32,33} - l_{31}}{v_{31}} < 0 \ . \tag{A.65}$$

A.7.5. Checkpoints 33–34

The product head moves with the velocity v_{31} according to (A.61) to stand 7 of the profile rolling line. After the product head reaches stand 7 at checkpoint 34, the product is fed into stand 7 with the velocity $v_{\text{P,in7}}$. If $v_{\text{P,in7}} < v_{31}$, the product is processed with so-called *separated speeds* using the looper. This implies that continuity equation $A_k v_k = \text{const.}$ is not applicable between the stands 6 and 7.

With the product length l_{31} according to (A.64) and the distance $d_{33,34}$ between the checkpoints 33 and 34, the entry time is

$$p_i^{\text{in},33,34} = \frac{d_{33,34}}{v_{31}} + \frac{l_{31} - d_{33,34}}{v_{\text{P,in7}}} \ . \tag{A.66}$$

The transit time is

$$p_i^{\text{thru},33,34} = \frac{d_{33,34} - l_{31}}{v_{\text{P,in7}}} < 0 \ . \tag{A.67}$$

A.7.6. Checkpoint 34 → Exit 4

The product with the length l_{31} according to (A.64) is fed into stand 7 with the velocity $v_{\text{P,in7}}$. The entry time is thus

$$p_i^{\text{in},34,\text{out}} = \frac{l_{31}}{v_{\text{P,in7}}} \ . \tag{A.68}$$

When the product tail has passed the checkpoint 19, the product leaves the system at Exit 4. Therefore, a transit time does not exist.

A.8. Evaluation

To validate the process time models, calculated process times are compared to actual values obtained by the material tracking system of the plant. Timestamps are recorded for some of the defined checkpoints. Either the passage time of the head $t_i^{\mathrm{H},k}$ or the passage time of the tail $t_i^{\mathrm{T},k}$ of a product σ_i is recorded. The difference between the times at two checkpoints is compared with the time resulting from the corresponding process time models. The following sections are evaluated:

$t_i^{\mathrm{H},13} - t_i^{\mathrm{H},2}$ is the time in which the product is ejected from the WBF and tranported to the reversing rolling stand.

$t_i^{\mathrm{T},20} - t_i^{\mathrm{H},13}$ is the time in which the product is rolled out at the reversing rolling stand and subsequently transported in the direction of the WBF.

$t_i^{\mathrm{H},15} - t_i^{\mathrm{H},13}$ is the time in which the product is rolled out at the reversing rolling stand and subsequently transported towards the flat rolling line.

$t_i^{\mathrm{T},18} - t_i^{\mathrm{H},15}$ is the time in which the product is rolled out at the flat rolling line.

$t_i^{\mathrm{H},25} - t_i^{\mathrm{T},20}$ is the time in which the product is transported towards and over the cross conveyor and subsequently through the heating box.

$t_i^{\mathrm{H},33} - t_i^{\mathrm{H},25}$ is the time in which the product head passes the first six stands of the profile rolling line and reaches a probe before the seventh stand.

$t_i^{\mathrm{T},33} - t_i^{\mathrm{H},33}$ is the time in which the product is rolled out at the seventh stand of the profile rolling line.

The resulting differences of the actual minus the calculated processing times are shown in a boxplot in Figure A.3. For each box, the central red line represents the median. The bottom and top edges of the boxes show the 25th and 75th percentiles, respectively. The whiskers of a maximum length of 1.5 times the interquartile range extend to the most extreme data points that are not outliers. Outliers are marked individually with a + symbol [56].

The largest deviation of actual and the calculated times (median equals $-2.4\,\mathrm{s}$) occurs at the section $t_i^{\mathrm{H},33} - t_i^{\mathrm{H},25}$ which represents the first 6 stands of the profile rolling line. Larger fluctuations occur in section $t_i^{\mathrm{H},25} - t_i^{\mathrm{T},20}$ which includes the heating box. These deviations are a consequence of products that have to wait in the heating box if they cannot be immediately processed further because of delays of the previous product at the profile rolling line. Analyses also showed that in this plant section the recorded time stamps sometimes do not match the actual passing times, resulting in deviations which are usually recognizable as outliers.

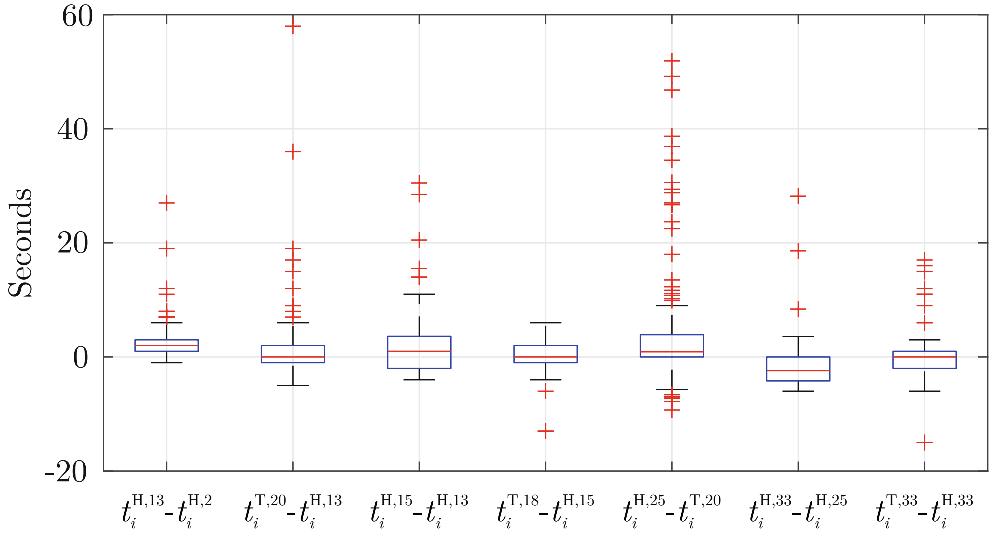

Figure A.3.: Boxplot of differences of actual and computed process times of individual products σ_i.

Bibliography

[1] E. Aarts and J. K. Lenstra, eds. *Local search in combinatorial optimization.* 1st ed. Chichester: Wiley, 1997.

[2] M. Abdolrazzagh-Nezhad and S. Abdullah. "Job shop scheduling: Classification, constraints and objective functions". In: *International Journal of Computer, Electrical, Automation, Control and Information Engineering* 11.4 (2017), pp. 368–391. DOI: doi.org/10.5281/zenodo.1129630.

[3] M. T. Adham and P. J. Bentley. "Artificial ecosystem algorithm applied to multi-line steel scheduling". In: *2019 IEEE Congress on Evolutionary Computation (CEC).* 2019 IEEE Congress on Evolutionary Computation (CEC). 2019, pp. 982–989. DOI: 10.1109/CEC.2019.8790320.

[4] J. F. Allen. "Maintaining knowledge about temporal intervals". In: *Communications of the ACM* 26.11 (1983), pp. 832–843. DOI: 10.1145/182.358434.

[5] A. Alqurashi, A. Etemadi, and A. Khodaei. "Model predictive control to two-stage stochastic dynamic economic dispatch problem". In: *Control Engineering Practice* 69 (2017), pp. 112–121. DOI: 10.1016/j.conengprac.2017.09.012.

[6] E. Angelelli, C. Bazgan, M. G. Speranza, and Z. Tuza. "Complexity and approximation for Traveling Salesman Problems with profits". In: *Theoretical Computer Science* 531 (2014), pp. 54–65. ISSN: 0304-3975. DOI: https://doi.org/10.1016/j.tcs.2014.02.046.

[7] S. Anily and A. Federgruen. "Simulated annealing methods with general acceptance probabilities". In: *Journal of Applied Probability* 24.3 (1987), pp. 657–667. DOI: 10.2307/3214097.

[8] A. Aschauer, F. Roetzer, A. Steinboeck, and A. Kugi. "An Efficient Algorithm for Scheduling a Flexible Job Shop with Blocking and No-Wait Constraints". In: *IFAC-PapersOnLine.* 20th IFAC World Congress 50.1 (2017), pp. 12490–12495. DOI: 10.1016/j.ifacol.2017.08.2056.

[9] A. Aschauer, F. Roetzer, A. Steinboeck, and A. Kugi. “Scheduling of a Flexible Job Shop with Multiple Constraints”. In: *IFAC-PapersOnLine*. 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018 51.11 (2018), pp. 1293–1298. DOI: 10.1016/j.ifacol.2018.08.354.

[10] A. Aschauer, F. Roetzer, A. Steinboeck, and A. Kugi. “Efficient scheduling of a stochastic no-wait job shop with controllable processing times”. In: *Expert Systems with Applications* 162 (2020), p. 113879. DOI: 10.1016/j.eswa.2020.113879.

[11] K. R. Baker and D. Trietsch. *Principles of sequencing and scheduling*. 1st ed. Hoboken, NJ: Wiley, 2009.

[12] D. P. Bertsekas. *Nonlinear programming*. Belmont: Athena Scientific, 2008.

[13] J. Błażewicz. *Scheduling computer and manufacturing processes*. 2nd ed. Berlin: Springer, 2001.

[14] C. Blum and G. R. Raidl. *Hybrid metaheuristics: powerful tools for optimization*. Cham: Springer, 2016.

[15] V. Černý. “Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm”. In: *Journal of Optimization Theory and Applications* 45.1 (1985), pp. 41–51. DOI: 10.1007/BF00940812.

[16] A. I. Chen, G. K. Yang, and Z. M. Wu. “Production scheduling optimization algorithm for the hot rolling processes”. In: *International Journal of Production Research* 46.7 (2008), pp. 1955–1973. DOI: 10.1080/00207540600988048.

[17] M. Chica, A. A. Juan Pérez, O. Cordon, and D. Kelton. “Why Simheuristics? Benefits, Limitations, and Best Practices When Combining Metaheuristics with Simulation”. In: *SSRN Electronic Journal* (2017). DOI: 10.2139/ssrn.2919208.

[18] G. B. Dantzig, D. R. Fulkerson, and S. M. Johnson. *Solution of a Large-Scale Traveling-Salesman Problem*. RAND Corporation, Apr. 12, 1954.

[19] S. Elmaghraby. “On the optimal release time of jobs with random processing times, with extensions to other criteria”. In: *International Journal of Production Economics* 74.1-3 (2001), pp. 103–113. DOI: 10.1016/S0925-5273(01)00111-6.

[20] D. Feillet, P. Dejax, and M. Gendreau. “Traveling salesman problems with profits”. In: *Transportation Science* 39.2 (2005), pp. 188–205. DOI: 10.1287/trsc.1030.0079.

[21] J. M. Framinan, R. Leisten, and R. Ruiz García. *Manufacturing scheduling systems: an integrated view on models, methods and tools*. London: Springer, 2014.

[22] M. Gendreau and J.-Y. Potvin, eds. *Handbook of Metaheuristics*. 2nd ed. Vol. 146. International Series in Operations Research & Management Science. New York: Springer, 2009.

[23] A. Gunawan, H. C. Lau, and P. Vansteenwegen. "Orienteering Problem: a survey of recent variants, solution approaches and applications". In: *European Journal of Operational Research* 255.2 (2016), pp. 315–332. DOI: 10.1016/j.ejor.2016.04.059.

[24] J. N. D. Gupta and E. F. Stafford. "Flowshop scheduling research after five decades". In: *European Journal of Operational Research* 169.3 (2006), pp. 699–711. DOI: 10.1016/j.ejor.2005.02.001.

[25] G. Gutin and A. Punnen, eds. *The traveling salesman problem and its variations*. Dordrecht: Kluwer, 2002.

[26] S. Hatami, L. Calvet, V. Fernández-Viagas, J. M. Framiñán, and A. A. Juan. "A simheuristic algorithm to set up starting times in the stochastic parallel flowshop problem". In: *Simulation Modelling Practice and Theory* 86 (2018), pp. 55–71. DOI: 10.1016/j.simpat.2018.04.005.

[27] F. Jin, S. Song, and C. Wu. "A simulated annealing algorithm for single machine scheduling problems with family setups". In: *Computers & Operations Research* 36 (2009), pp. 2133–2138. DOI: 10.1016/j.cor.2008.08.001.

[28] N. L. Johnson, A. W. Kemp, and S. Kotz. *Univariate Discrete Distributions*. Hoboken: Wiley, 1995.

[29] A. Juan and M. Rabe. "Combining simulation with heuristics to solve stochastic routing and scheduling problems". In: Oct. 2013, pp. 641–649.

[30] A. A. Juan, J. Faulin, S. E. Grasman, M. Rabe, and G. Figueira. "A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems". In: *Operations Research Perspectives* 2 (2015), pp. 62–72. ISSN: 2214-7160. DOI: 10.1016/j.orp.2015.03.001.

[31] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. "Optimization by Simulated Annealing". In: *Science* 220.4598 (1983), pp. 671–680. DOI: 10.1126/science.220.4598.671.

[32] M. Kowalski, A. Kugi, and A. Steinboeck. "Job Scheduling for a Multi-Line Steel Hot Rolling Mill With Selectable Furnaces". In: *IFAC-PapersOnLine* 56.2 (2023). 22nd IFAC World Congress, pp. 5376–5381. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2023.10.184.

[33] M. Kowalski, A. Steinboeck, A. Aschauer, and A. Kugi. “Optimal Start Times for a Flow Shop with Blocking Constraints, No-Wait Constraints, and Stochastic Processing Times”. In: *IFAC-PapersOnLine* 54.1 (2021), pp. 659–664. DOI: 10.1016/j.ifacol.2021.08.176.

[34] M. Kowalski, A. Steinboeck, and A. Kugi. “Scheduling Multiple Groups of Jobs for a Multi-Line Steel Hot Rolling Mill”. In: *IFAC-PapersOnLine* 55.21 (2022), pp. 168–173. DOI: 10.1016/j.ifacol.2022.09.262.

[35] L. Kroon, G. Maróti, M. Helmrich, M. Vromans, and R. Dekker. “Stochastic improvement of cyclic railway timetables”. In: *Transportation Research Part B: Methodological* 42.6 (2008), pp. 553–570. DOI: 10.1016/j.trb.2007.11.002.

[36] W.-Y. Ku and J. C. Beck. “Mixed Integer Programming models for job shop scheduling: A computational analysis”. In: *Computers & Operations Research* 73 (2016), pp. 165–173. DOI: 10.1016/j.cor.2016.04.006.

[37] P. J. M. v. Laarhoven and E. H. L. Aarts. *Simulated annealing: theory and applications*. Mathematics and its applications. Dordrecht: Kluwer Academic Publishers, 1992.

[38] L. Landau and E. Lifshitz. *Statistical Physics*. 3rd ed. Vol. 5. Course of Theoretical Physics. Elsevier Science, 1980.

[39] E. Langford. “Quartiles in Elementary Statistics”. In: *Journal of Statistics Education* 14.3 (Jan. 1, 2006). DOI: 10.1080/10691898.2006.11910589.

[40] Y.-C. Liang, S. Kulturel-Konak, and A. E. Smith. “Meta heuristics for the orienteering problem”. In: Proceedings of the 2002 Congress on Evolutionary Computation. Vol. 1. 2002, pp. 384–389. DOI: 10.1109/CEC.2002.1006265.

[41] L. Lopez, M. Carter, and M. Gendreau. “The hot strip mill production scheduling problem: A tabu search approach”. In: *European Journal of Operational Research* 106.2 (1998), pp. 317–335. ISSN: 0377-2217. DOI: 10.1016/S0377-2217(97)00277-4.

[42] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. “Equation of State Calculations by Fast Computing Machines”. In: *The Journal of Chemical Physics* 21.6 (1953), pp. 1087–1092. DOI: 10.1063/1.1699114.

[43] Z. Michalewicz and D. B. Fogel. *How to solve it: modern heuristics*. 2nd ed. Heidelberg: Springer, 2010.

[44] C. E. Miller, A. W. Tucker, and R. A. Zemlin. “Integer Programming Formulation of Traveling Salesman Problems”. In: *Journal of the ACM* 7 (1960), pp. 326–329. DOI: 10.1145/321043.321046.

[45] K. Nailwal, D. Gupta, and K. Jeet. “Heuristics for no-wait flow shop scheduling problem”. In: *International Journal of Industrial Engineering Computations* 7 (2016), pp. 671–680. DOI: 10.5267/j.ijiec.2016.2.005.

[46] R. Ojstersek, M. Brezocnik, and B. Buchmeister. “Multi-objective optimization of production scheduling with evolutionary computation: A review”. In: *International Journal of Industrial Engineering Computations* 11 (2020), pp. 359–376. DOI: 10.5267/j.ijiec.2020.1.003.

[47] A. Özgür, Y. Uygun, and M.-T. Hütt. “A review of planning and scheduling methods for hot rolling mills in steel production”. In: *Computers & Industrial Engineering* 151 (2021), p. 106606. DOI: 10.1016/j.cie.2020.106606.

[48] M. Pincus. “A Monte Carlo Method for the Approximate Solution of Certain Types of Constrained Optimization Problems”. In: *Operations Research* 18.6 (1970), pp. 1225–1228. DOI: 10.1287/opre.18.6.1225.

[49] M. Pinedo. *Scheduling: Theory, Algorithms, and Systems*. Cham: Springer, 2016.

[50] M. N. Rosenbluth. “Genesis of the Monte Carlo Algorithm for Statistical Mechanics”. In: *AIP Conference Proceedings* 690.1 (2003), pp. 22–30. DOI: 10.1063/1.1632112.

[51] B. Silverman. *Density Estimation for Statistics and Data Analysis*. Vol. 26. Monographs on Statistics and Applied Probability. London, New York: Chapman and Hall, 1986. ISBN: 978-1-351-45617-3.

[52] K. Steger-Jensen, H.-H. Hvolby, P. Nielsen, and I. Nielsen. “Advanced planning and scheduling technology”. In: *Production Planning & Control* 22.8 (2011), pp. 800–808. DOI: 10.1080/09537287.2010.543563.

[53] E.-G. Talbi. *Metaheuristics: From Design to Implementation*. 1st ed. Hoboken: Wiley, 2009.

[54] The MathWorks Inc. *MATLAB version: 9.13.0 (R2022b)*. Natick, Massachusetts, United States, 2022.

[55] The MathWorks Inc. *Optimization Toolbox version: 9.4 (R2022b)*. Natick, Massachusetts, United States, 2022.

[56] The MathWorks Inc. *Statistics and Machine Learning Toolbox version: 12.4 (R2022b)*. Natick, Massachusetts, United States, 2022.

[57] P. Toth and D. Vigo. *Vehicle Routing*. Ed. by D. Vigo and P. Toth. 2nd ed. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2014.

[58] P. Vansteenwegen and A. Gunawan. *Orienteering problems: models and algorithms for vehicle routing problems with profits*. EURO Advanced Tutorials on Operational Research. Springer, 2019.

[59] P. M. Verderame, J. A. Elia, J. Li, and C. A. Floudas. "Planning and Scheduling under Uncertainty: A Review Across Multiple Sectors". In: *Industrial & Engineering Chemistry Research* 49.9 (2010), pp. 3993–4017. DOI: `10.1021/ie902009k`.