TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Produktion und Supply Chain Management

Optimization-based approaches for product platform design, production planning, and capacity modeling in semiconductor supply chains

Phillip Oliver Kriett

Vollständiger Abdruck der von der Fakultät für Wirtschaftswissenschaften der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Wirtschaftswissenschaften (Dr. rer. pol.)

genehmigten Dissertation.

Vorsitzender:	Prof. Dr. Stefan Minner
Erster Prüfer:	Prof. Dr. Martin Grunow
Zweiter Prüfer:	Prof. Dr. Hubert Missbauer Universität Innsbruck

Die Dissertation wurde am 01.03.2019 bei der Technischen Universität München eingereicht und durch die Fakultät für Wirtschaftswissenschaften am 15.08.2020 angenommen.

Acknowledgments

First and foremost, I am very grateful to my supervisor Prof. Dr. Martin Grunow for his invaluable support, advice, and encouragement throughout the course of my PhD studies. He has always been extremely passionate about my research and supportive of my career, and offered numerous opportunities for me to mature as a researcher and as a person. I learned from you so many things about research, teaching, and life. Thank you for making these years the happiest and most growing time in my life.

My sincere thanks go to Prof. Dr. Hubert Missbauer and Prof. Dr. Stefan Minner for their insightful comments on my research as well as for taking the role of the second examiner and the chairman of the examination committee. I would like to thank all current and former members of the Department of Operations and Supply Chain Management for creating such a friendly and collaborative working environment. Outside of our department, I would like to thank Prof. Dr. Peter Gritzmann and his team at the Department of Mathematics, Prof. Dr. Renzo Akkerman at the Wageningen University & Research, as well as Prof. Dr. Argon Chen and Prof. Dr. Shi-Chung Chang at the Institute of Industrial Engineering of National Taiwan University for many fruitful discussions. Thanks are due to the external partners of this research in the semiconductor industry for numerous joint projects. In particular, I would like to thank Hans Ehm and his colleagues at Infineon Technologies as well as Ryan Rhoades and his former team at Siltronic. I also would like to thank the anonymous reviewers of the IISE Transactions and the International Journal of Production Research for their sharp feedback and constructive criticism.

Lastly, I would like to give my everlasting thanks to my parents Dorothee and Peter as well as my sisters Laura and Leonie. I could have never made this journey without their unconditional love and support. A special thanks for her love and patience goes to Esther: My partner in crime and the mother of our beautiful son Theodore.

Abstract

Semiconductor supply chains are challenged by an intense international competition, technological complexity, and a high innovation rate, which is typical for the semiconductor industry. This dissertation addresses these challenges by proposing optimization-based approaches for three different decision problems in the areas product platform design and hierarchical production planning. The proposed approaches are applied to cases from the semiconductor industry and extensive numerical experiments are conducted for validation and evaluation. First, stochastic optimization is introduced for product platform design in silicon wafer manufacturing. Numerical results show that taking the uncertainty about future demand explicitly into account helps to design product platforms optimally — also to the requirements of future customer orders — and thus reduce future design workload and costs. Second, a novel cycle time-oriented mid-term production planning model is applied to wafer fabrication. Tightly integrated with production control, the model ensures an optimal response to machine failures and unforeseen demand changes. Compared to conventional work-in-process-oriented planning, cycle time-oriented planning delivers higher service levels, shorter cycle times, and it generates simpler production plans. Third, a low-dimensional capacity model is suggested for company-wide production planning. It hides the detailed capacity allocation decisions, which are usually made for parallel machines, and thus reduces the complexity of the planning process. Depending on the number of modeled machines and products, an exact or a heuristic procedure is used for the generation of specific capacity constraints. An aggregation step, which exploits certain attributes of machines and products, reduces the problem size and computation time. Compared to existing methods, the proposed procedures deliver a more accurate representation of throughput limitations, in particular for parallel machines. The presented optimization-based approaches improve the resource efficiency and the service level of semiconductor supply chains and thus strengthen the resilience of manufacturers against future challenges.

Zusammenfassung

Die Herausforderungen, denen sich Halbleiterlieferketten stellen, ergeben sich aus dem starken internationalen Wettbewerb, der technologischen Komplexität, und der hohen Innovationsrate, die typisch für die Halbleiterindustrie ist. Die vorliegende Dissertation begegnet diesen Herausforderungen mit Optimierungsmodellen für drei unterschiedliche Entscheidungsprobleme aus den Bereichen Produktplattformentwicklung und hierarchische Produktionsplanung. Die präsentierten Modelle werden auf Fälle aus der Halbleiterindustrie angewendet und in umfangreichen numerischen Experimenten validiert und bewertet. Zuerst wird stochastische Optimierung zur Entwicklung von Produktplattformen in der Substratherstellung vorgeschlagen. Die numerischen Ergebnisse zeigen, dass die explizite Berücksichtigung der Nachfrageunsicherheit dabei helfen kann, Produktplattformen optimal zu entwickeln — auch gemäß den Anforderungen zukünftiger Bestellungen — und dadurch zukünftigen Entwicklungsaufwand und Kosten zu reduzieren. Zweitens wird eine neuartige Fertigungszeit-orientierte mittelfristige Produktionsplanung für die Fertigung von integrierten Schaltkreisen eingeführt. Eng verbunden mit der Produktionssteuerung, sorgt das Planungsmodell für eine optimale Reaktion auf Maschinenausfälle und unvorhergesehene Änderungen der Nachfrage. Im Vergleich zur konventionellen Umlaufbestand-orientierten Produktionsplanung erhöht Fertigungszeit-orientierte Planung den Lieferbereitschaftsgrad, reduziert die Fertigungszeit, und generiert einfachere Produktionspläne. Drittens wird ein niedrigdimensionales Kapazitätsmodell für die unternehmensweite Produktionsplanung empfohlen. Es verdeckt die detaillierten Kapazitätsallokationsentscheidungen, welche normalerweise für parallele Maschinen getroffen werden müssen, und reduziert so die Komplexität des Planungsprozesses. Abhängig von der Anzahl der modellierten Maschinen und Produkte wird ein exaktes oder ein heuristisches Verfahren zur Generierung spezifischer Kapazitätsbeschränkungen eingesetzt. Ein Aggregationsschritt, welcher bestimmte Eigenschaften von Maschinen und Produkten ausnutzt, reduziert die Problemgröße und damit die Berechnungszeit. Im Vergleich zu existierenden Methoden liefern die vorgeschlagenen Verfahren ein akkurateres Abbild der Durchsatzbeschränkung, insbesondere für parallele Maschinen. Die beschriebenen Optimierungsmodelle unterstützen die Ressourceneffizienz und den Lieferbereitschaftsgrad in Halbleiterlieferketten und verbessern so die Widerstandsfähigkeit der Hersteller gegenüber zukünftigen Herausforderungen.

Contents

List of Tables

List of Figures

List of Abbreviations

ANOVA	Analysis of variance
CT	Cycle time
CT-LP	Cycle time-oriented linear programming formulation
DPF	Direct product mix formulation
ETO	Engineer-to-order
fab	Wafer fabrication facility
FIFO	First in first out
FWS	Finished wafer stock
HSD	Honestly significant difference
IC	Integrated circuit
IoT	Internet of things
LP	Linear programming
MILP	Mixed integer-linear programming
MTPPS	Mid-term production planning and scheduling
NP	Nondeterministic polynomial time
ODD	Operation due date
PAR	Production achievement ratio
PCS	Partition-based constraint search
SSF	Step-separated formulation
SUPM	System of unrelated parallel machines
TPPDP	Tactical product platform design problem
UPM	Unrelated parallel machines
WIP	Work in process
WIP-LP	Work in process-oriented linear programming formulation

Chapter 1

Introduction

1.1 Semiconductors

Semiconductors are one of the most pervasive and powerful inventions in human history and have affected almost every aspect of human life. With a world population of 7.55 billion in 2017, there were globally about 18 billion semiconductor devices connected to IP networks (see Cisco, 2018). A continuation of growth from 2.4 network devices per capita in 2017 to 3.6 by 2022 is in particular expected from Internet of Things (IoT) applications, i.e., "systems of interconnected people, physical objects, and IT platforms, as well as any technology to better build, operate, and manage the physical world via pervasive data collection, smart networking, predictive analytics, and deep optimization" (see IEEE-SA, 2015). The value of global semiconductor production was $412 billion in 2017. In the same year, European semiconductor manufacturers generated $38 billion of revenue, to which the German semiconductor industry contributed around $14.7 billion (see WSTS, 2018; ZVEI, 2017). Electronics, which includes semiconductors, is the second largest manufacturing industry in Germany and employs 880 thousand people in 2018 (see ZVEI, 2018).

The most important type of semiconductor is the integrated circuit (IC), which includes analog, micro, logic, and memory chips. It accounts for 84 % of the world-wide semiconductor market (see WSTS, 2018). Other types of semiconductors are discrete semiconductors, optoelectronics, and sensor devices. ICs are created on the surface of a semiconductor material, which is commonly a thin slice of mono-crystalline silicon (also denoted as wafer or substrate). Silicon wafers are typically manufactured by growing a single-crystal rod (also called ingot) from a silicon melt to a cylindrical shape of several meters in length and up to 450 mm in diameter. These ingots are sliced into wafers, which undergo surface treatments. The creation of ICs on the surface of a wafer is called wafer fabrication. In the production stages subsequent to wafer fabrication, a wafer is cut apart into dies, dies are sorted and assembled to semiconductor devices, which are again tested (see Figure 1.1). Dies are small

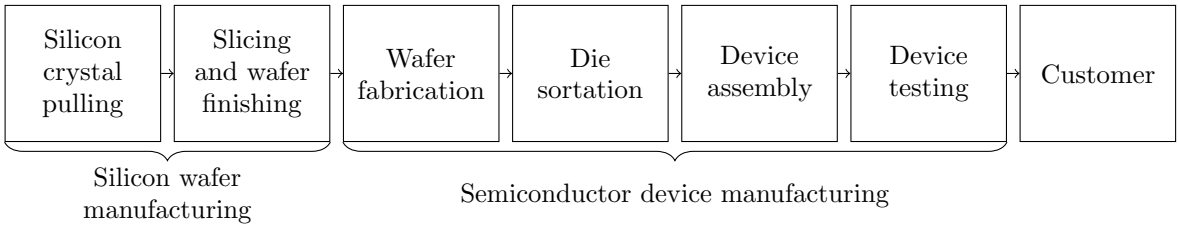


Figure 1.1: Production stages in a typical semiconductor supply chain.

fragments of the wafer, each holding a functional IC. Depending on the size of both wafer and IC, hundreds or thousands of dies are fabricated on a wafer. For more information about semiconductors and wafer fabrication technology, see for example Geng (2005) and Turley and Turley (2003).

1.2 Challenges in semiconductor supply chains

Semiconductor supply chains are shaped by distinct attributes that are characteristic for the semiconductor industry. Commonalities among semiconductor manufacturers are the exposure to a competitive international market, the mastering of technologically complex production processes, and a high innovation rate. These challenges form the root causes of many specific decision problems in the semiconductor industry.

Semi-finished and finished semiconductor products have high value-to-volume ratios. The manufacturing network of semiconductor device manufacturers, such as Infineon, prove that it is economical to ship semi-finished and finished products across continents in order to exploit the competitive advantage of different locations for different stages of production (see, e.g., Ehm et al., 2011). Geographical distance is likewise not a barrier for market entry, which means that companies from across the globe often compete for the same group of customers. This ensures low consumer prices and it forces manufacturers and suppliers to continuously improve the cost, speed, and quality of their operations.

Silicon wafer manufacturing and wafer fabrication have high requirements on cleanliness and precision. The radius of a silicon atom is in the range of 0.1 nanometer. IC features, which are created in wafer fabrication, are as small as a few nanometers. An important step in wafer fabrication is photo lithography, which transfers a two-dimensional pattern from a photo mask to the surface of a with light-sensitive chemicals covered wafer. After exposure, the physiochemical attributes of the patterned wafer surface is altered by applying techniques, such as etching, deposition, and oxidation. Up to 40 of such layers, which have to be perfectly aligned, are iteratively created and build the desired IC (see, e.g., Mönch et al., 2013). The required level of precision and purity is achieved with substantial investments in technology, which push the total cost of a new wafer fabrication facility up to several billion Dollars (see, e.g., Intel, 2017; Samsung, 2017; TSMC, 2018).

Semiconductor devices are high-tech products and technological progress is ongoing. The most popular observation that illustrates the pace of development in the semiconductor industry is probably Moore's Law, which has correctly predicted the doubling of the number of transistors per IC every 12 to 24 months, beginning with one transistor in 1959 (see Mack, 2011; Moore, 1965). As a result of this efficiency gain, every newly developed device experiences a price decline once it is introduced. Declining product prices have to be considered at all planning levels since inventories can pose a financial risk. Declining product prices also incentivize investments in research and development in order to secure future business opportunities.

1.3 Research objectives

The objective of this dissertation is to contribute to the knowledge about relevant decision making problems in semiconductor supply chains. We identify three research questions based on cases that were discovered in joint projects with partners from the local semiconductor industry. These research questions are linked to each other by the general challenges in semiconductor supply chains and address three different decision making problems, which cover different supply chain segments (see Figure 1.1) and belong to different levels of hierarchical production planning and control (see, e.g., Stadtler, 2015). Due to the complementary nature of the studied topics, this dissertation provides a broad picture of supply chain management-related issues in the semiconductor industry. Every research question is answered with a quantitative approach and involves some form of optimization. We demonstrate all proposed models in extensive numerical experiments with both industry and randomly generated problem instances. Although these models are tailored to the specifics of the semiconductor industry, we provide generic problem formulations and model descriptions that allow an easy application to similar settings in other industries. In the following, we outline the three cases and formulate the three research questions that are addressed in this dissertation.

The first case refers to product design in silicon wafer manufacturing. The high innovation rate in the semiconductor industry, which manifests itself in the frequent development of new IC generations, also affects the suppliers of silicon wafers. A new generation of ICs often requires a different substrate than previous generations, e.g., with different impurity levels. Designing a silicon wafer is expensive. It involves an iterative process of manufacturing sample ingots and testing the resulting sample wafers on the wafer fabrication equipment of the customer until the physiochemical requirements are satisfied. Several business functions including sales, product design, engineering, and production contribute to this process. If customer orders for new wafer types could be served with existing ingots instead of designing new ones, this would reduce the product development workload and enable a shorter time to market for the device that the customer intends to manufacture. The customer formulates many of the feature value requirements that the wafer has to satisfy as tolerance intervals.

Pending orders, ongoing negotiations, and technological trends can be used by the silicon wafer manufacturer to forecast future feature value requirements. An ingot can be designed not only to the requirements of a customer order on hand but also the forecasted requirements of future orders. However, in order to be feasible for a given set of customer orders, an ingot has to meet in every feature the most demanding requirement. This increases the expected production cost of all served orders. In the search for the optimal ingot design, this production cost increase has to be traded off against the expected reduction of design costs. The objective is to design an ingot that serves as a product platform for a range of customer orders in the present and in the future such that total costs are minimized. In this context, the first research question is:

Research question 1. *In an engineer-to-order manufacturing environment with a rapidly evolving product portfolio and uncertain future demand, what is the optimal product platform design?*

The second case is about mid-term production planning in wafer fabrication. Wafer fabrication is the creation of ICs on the surface of silicon wafers. It is a technologically sophisticated production stage of semiconductor device manufacturing and requires expensive tools and equipment. The technological complexity translates into operational complexity as equipment costs make a job shop with recirculating process flows the only financially viable form of organizing production. ICs are fabricated layer by layer, which means that wafers pass a series of process steps several times. A process flow describes the complete sequence of process steps, which determine the resulting type of IC. A wafer fabrication facility (short wafer fab) often fabricates several types of ICs in parallel. As a result, equipment is shared by wafer lots that differ in both process flow and level of completion. For instance, a medium-sized wafer fab can release 11,000 wafers per week distributed across 25 different recirculating process flows. Every process flow can count 400 to 800 process steps visiting 160 work centers, which are defined by grouping 1000 individual machines according to their capabilities. A wafer cycles 30 to 80 days through the fab before it is finished. Both, uncertain yield and uncertain customer demand lead to uncertain output targets for the wafer fab. Output targets that are raised within the cycle time of 30 to 80 days can only be met by prioritizing wafer lots that are in process. Mid-term production planning determines release quantities and lot priorities for both new releases and work in process (WIP) such that multiple objectives are optimized: High fixed costs make high throughput rates and therefore the avoidance of machine idling necessary. The frequent development of new ICs, uncertain output targets, and the deflation of product prices demand short cycle times, low WIP levels, and thin inventories. A high on-time delivery performance is a prerequisite for maintaining excellent customer relationships. The second research question is hence:

Research question 2. *Which mid-term production planning method meets best the conflicting objectives in wafer fabrication and how does it interact with production control?*

The third case addresses production planning (also called master production scheduling) in semiconductor device manufacturing. Production planning coordinates the flow of material and the allocation of bottleneck resources at the company-level such that costs are minimized and promised deliveries are fulfilled on time. The production network of a semiconductor device manufacturer typically includes the stages wafer fabrication, sortation, assembly, and test. Traditionally, there is a stock of dies located between sort and assembly and a stock of semiconductor devices held after final test. But there can be more buffer inventories distributed across the network. Production planning ensures that the right quantity of the right type of semi-finished product leaves a buffer inventory and seizes a bottleneck resource in the right time bucket. It also ensures that sufficient capacity of bottleneck resources is reserved for product development. An accurate model of the throughput limitations of bottleneck work centers is a prerequisite for production planning, which intends to prevent excessive queues as well as machine idling. Work centers often consist of several machines of different age, whose performance can differ in speed and capability. The traditional way of modeling the capacity of such a collection of parallel machines is to capture the capacity allocation of individual machine types to individual product types in detail. While the accurate representation of bottleneck work center capacity is crucial, the scale and scope of semiconductor device manufacturing make it undesirable to model detailed resource allocation decisions in the production planning problem. Instead, planners prefer capacity constraints that limit the total production rates of product types without surfacing the complexity of how workload is distributed across parallel machines. The third research questions is therefore:

Research question 3. *How can we accurately capture the capacity of parallel machines such that the total production rates of product types are the only variables of the model?*

1.4 Outline

This dissertation is organized as a collection of three research papers, which can be read as individual contributions. Every paper is dedicated to one of the three research questions and forms a separate chapter. In Chapter 2, we propose and evaluate a two-stage stochastic optimization model to solve the product platform design problem for both customer orders on hand and expected future customer orders. This chapter aims at answering research question 1. In Chapter 3, we propose a novel mid-term production planning approach that is cycle time-oriented. We demonstrate its interaction with production control as well as its superiority compared to popular WIP-oriented production planning. This chapter aims at answering research question 2. In Chapter 4, we propose and evaluate a procedure that generates low-dimensional linear capacity constraints for parallel machines. These constraints contain one decision variable per product type, modeling the total production quantity of every product type. This chapter aims at answering research question 3. Every chapter contains a

review of the respective literature, which is not limited to semiconductors but delimits our contributions from existing work in general. In addition to the conclusions that are drawn at the end of every chapter, we finish this dissertation in Chapter 5 with a high-level summary of the generated insights and an expansion on our discussion of future research opportunities.

Chapter 2

Tactical platform design in supply chains with rapidly evolving product portfolios

Abstract

Engineer-to-order companies design products to the requirements of individual customer orders. The design of new products occupies a highly specialized and expensive workforce, which experiences an increasing workload due to shortening product life cycles. An opportunity to reduce the cost and workload of product design is to introduce product platforms that serve both present and future customer orders. However, future customer orders are uncertain and designing a product platform to cover the requirements of the most demanding expected customer order increases the manufacturing costs of all product variants served by the platform. We propose a two-stage stochastic program with recourse that determines the optimal number and designs of product platforms. It trades a reduction of design costs off against an increase in manufacturing costs taking present and expected future orders into account. We test our approach in settings typical for the silicon wafer manufacturing industry. Compared to existing modelling approaches derived for the deterministic platform design problem, our efficient model requires fewer binary decision variables, which reduces computation times significantly. Numerical experiments also show the greatest benefit of tactical product platform design in situations with high design costs, low growth and variability of feature value requirements, and low growth of order quantities.

2.1 Introduction

Form postponement describes the delay of product differentiation to down-stream processing steps. It typically involves the manufacture of components to stock, from which customization steps pull. Product platforms enable form postponement as they delay product differentiation (see, e.g., Jiao et al., 2007; Su et al., 2005). The key benefits of form postponement with product platforms are the reduction of safety stock levels due to risk-pooling and the reduction of complexity in the manufacturing system (see, e.g., Hillier, 2000; Lee and Tang, 1997). In addition, design and engineering costs as well as product lead times are reduced (see, e.g., Fisher et al., 1999; Jans et al., 2008; Perera et al., 1999), which is of particular relevance for engineer-to-order (ETO) environments. The main drawback is the increase in manufacturing costs because of platform designs that exceed the feature value requirement of some of the served product variants (also denoted as over-costs, see Briant and Naddef (2004)).

Product platform design selects common parts and an underlying core technology that are to be implemented across a range of product variants. A product platform is often designed by combining readily specified modular components (see, e.g., Ben-Arieh et al., 2009; Swaminathan and Tayur, 1998). Customized product variants are then derived by removing and adding components, which has been proposed, for example, for the manufacture of power tools and other electronic devices. A more general approach to product platform design is to determine all the feature values that specify a platform (see, e.g., Boysen and Scholl, 2009; Fujita and Yoshida, 2004; Menezes et al., 2016; Thonemann and Brandeau, 2000). Customized product variants are then derived from a platform unit in downstream processes. Typical examples can be found in manufacturing industries and also in the process industry (see, e.g., Kilic et al., 2013). This paper extends the literature on the latter, more general approach.

The benefit of a product platform depends on its design. Designing product platforms is in particular challenging in supply chains with rapidly evolving product portfolios. While old product variants phase out, new product variants with different requirements are introduced to keep up with technological progress. Moreover, legacy product variants are often tied to their product platform because the costs of change are prohibitive. A reconfiguration of the supply chain as well as product approvals by authorities and the customer are time-consuming and expensive. As a result, product platform designs are a differentiating factor from the customer's perspective.

In order to take full advantage of product platforms in changing environments, platform design has to take present and future expansions of the product portfolio into account. We define the tactical product platform design problem (TPPDP), which is to minimize the costs at which product platforms are designed and manufactured to serve product variants that are ordered in the present and in the future. Both the feature value requirement and the order quantity of future customer orders are usually uncertain but insights from pending orders, ongoing negotiations, and technological trends can be leveraged to specify expectations. The

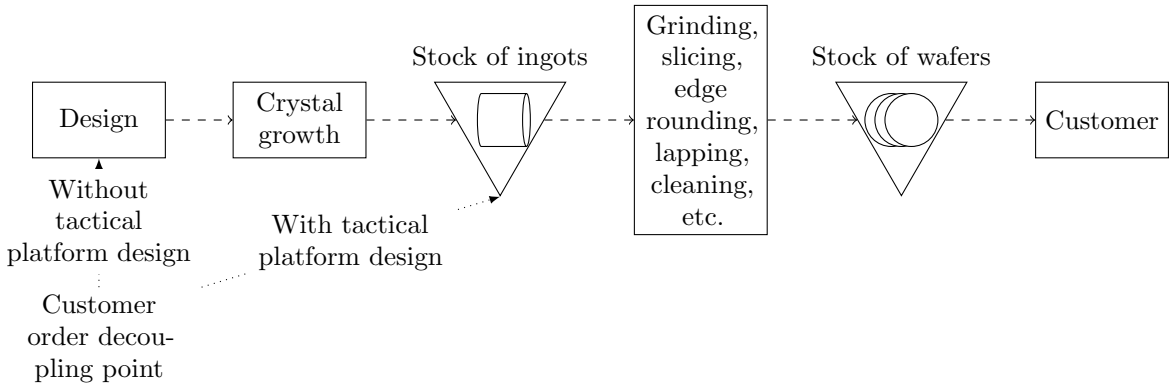


Figure 2.1: Postponement of the customer order decoupling point in silicon wafer manufacturing.

TPPDP is initialized when customer orders on hand require the design of a new product platform. This is the case when none of the existing platforms meets the feature value requirements. The question is then if it is optimal to design a product platform not only to the requirements of today's but also to the requirements of expected future orders. This tactical decision can reduce the number of future customer orders that enter the platform design phase. In order to minimize total costs, the TPPDP formulation trades off the reduction of platform design costs against the increase of manufacturing costs.

This paper is motivated by a case in the semiconductor silicon wafer industry, in which silicon wafers are produced by an ETO company. Subsequent to the design phase, cylindric silicon monocrystals (also called ingots) are manufactured in a crystal growth process, such as Czochralski pulling. These ingots are characterized by several features including the grow material type, silicon grade, dopant material, crystal orientation, electrical resistivity, the number of oxidation-induced stacking faults, and the concentration of carbon, oxygen, and iron. Ingots are sliced into silicon wafers, which undergo finishing steps, such as edge rounding, lapping, cleaning, etching, coating, polishing, and epitaxy. Especially in the semiconductor industry, customer preferences are pushing for short product lead times, short product life cycles, and a large product variety. As a result, ETO companies are experiencing decreasing order quantities and an increasing order frequency, which is a common observation in ETO supply chains (see, e.g., Kumar and Wellbrock, 2009). We investigate the opportunity to postpone the customer order decoupling point from the design phase to the stock of ingots with tactical product platform design (see Figure 2.1).

The scientific contributions of this paper include:

- a formulation of the product platform design problem that
  - accounts for an evolving product portfolio in a two-stage stochastic programming formulation;

  - supports both discrete and continuous feature values and efficiently models the quadratic relationship between platform design and platform assignment decisions;

- an application of the proposed model to a setting that is typical for the silicon wafer industry and numerical results that demonstrate its efficiency and effectiveness;
- the identification of situations in which tactical product platform design is most beneficial.

In the following Section 2.2, we review the relevant literature. In Section 2.3, we first define the TPPDP and then present a two-stage stochastic programming formulation with recourse. In Section 2.4, we discuss the assumptions that underlie the generation of random problem instances and present the numerical results of experiments that reveal the effectiveness and the efficiency of the proposed formulation. We finish the paper with our conclusive remarks in Section 2.5.

2.2 Related work

A product platform can be either broadly defined as a collection of assets, i.e., components, processes, knowledge, people, and relationships, or more narrowly as a set of physical components, modules, and parts (Jiao et al., 2007). This paper is concerned with the design of physical product platforms, which is related to the component commonality problem. Comprehensive reviews of models and methods that have been proposed for the design of common components can be found in Wazed et al. (2010), Fixson (2007), and Labro (2004).

Simpson (2004) and Jiao et al. (2007) present a classification and summary of approaches that facilitate platform-based product development and the optimization of product platforms. The objective is to simultaneously design multiple products for achieving higher optimality – beyond design methods that consider a single product at a time. Product platforms help accomplishing product variety at lower costs compared to every product is made from a unique semi-finished product. Economies of scale are realized by reducing the number of different semi-finished products, while the valuable variety of final product variants remains unrestricted.

If the specification of product platforms can be reduced to a single feature, platform design is an assortment problem. One-way substitution (i.e., feature values can exceed the requirement but must not fall below) allows the grouping of customer orders so that for every group, a platform is designed to the requirement of the most demanding variant within the group. Jans et al. (2008) formulate this deterministic, single-stage optimization problem as a MILP, which maximizes profit and balances platform development costs, manufacturing costs, and selling prices. In order to solve large-scale instances of this problem, Briant and Naddef (2004) and Menezes et al. (2016) propose a Lagrangian relaxation and a steepest descent greedy algorithm, respectively. They assume that the required number of product platforms is given as an input and focus on minimizing manufacturing costs.

Thonemann and Brandeau (2000) propose an integer programming formulation of a deterministic, single-stage component commonality problem that minimizes the total costs of production, setups, holding inventory, and variability in the manufacturing system. They solve the problem with a branch-and-bound algorithm and simulated annealing. The solution defines the optimal number and specification of components as well as the assignment of components to product variants. While Thonemann and Brandeau assume that components are characterized by features that can be either turned on or off, Boysen and Scholl (2009) extend every feature by a finite set of feasible feature values that allow one-way substitution. Boysen and Scholl formulate the problem as a MILP and present a two-stage graph-based heuristic to solve the problem.

Swaminathan and Tayur (1998) and Ben-Arieh et al. (2009) define a product platform as a composition of modular, fully specified components. Product variants are assembled based on a platform by adding and removing components. The problem is to specify platform configurations and define production quantities so that the total costs of mass-producing product platforms, assembling product variants, holding inventory, and experiencing stock-outs are minimal for a given demand. Swaminathan and Tayur (1998) also consider that future demand is stochastic and propose a two-stage stochastic programming formulation, in which platform configurations and production quantities are decided in the first stage, while potential platform modifications and the additional production of missing parts are modelled as recourse decisions. The model does, however, not address the uncertainty of the feature value requirements of future customer orders.

In summary, there exist optimization-based methods to solve the deterministic single-stage product platform design problem. Few contributions consider the uncertainty of future customer orders. There is a gap in the scientific literature as there has not been presented a model that accomplishes tactical product platform design for an evolving product portfolio with uncertainty about both order quantities and feature value requirements of future customer orders.

2.3 Tactical product platform design

2.3.1 Problem statement

ETO companies design product variants to the requirements of customer orders. Customer orders specify feature value requirements with tolerance intervals. It is assumed that a semi-finished product can serve as product platform and be the basis for more than one type of product variant. The extent of such a product platform depends on its compliance with the feature value requirements of customer orders.

A set of customer orders arrives in the present. These orders define the feature value requirements and order quantities of product variants that have to be designed and manufactured in the present. Reorders in the future of the same product variants are possible but order quantities are uncertain. In addition, a set of different customer orders is expected to arrive in the future. Both the feature value requirements and the order quantities of these orders are uncertain.

Product platform design incurs fixed costs, which represent design costs and the costs of adding complexity to the production system. Manufacturing a product platform incurs variable costs, which depend on the realized feature values and the production volume. Every product variant has to be served by one product platform and such assignments made in the present cannot be modified in the future. The TPPDP is to determine the optimal product platform designs in order to meet customer orders on hand and, potentially, expected future customer orders. The objective is to minimize the total costs of designing and manufacturing product platforms in the present and in the future, which requires the balancing of certain and uncertain fixed costs with certain and uncertain manufacturing costs along the time line.

2.3.2 Two-stage stochastic program with recourse

We formulate the TPPDP as a two-stage stochastic program with recourse. Similar to the core component-commonality problem presented by Boysen and Scholl (2009), the proposed formulation determines how many different product platforms have to be designed, what is the specification of these product platforms, and which product variant is served by which product platform. In contrast to Boysen and Scholl (2009), our formulation answers these questions for a product portfolio that evolves over time. Moreover, our novel modelling approach allows the use of continuous instead of discrete decision variables to model platform design decisions, i.e., the specification of the feature values that are going to be realized in product platforms.

Modelling approach and assumptions

We assume that every customer order specifies the order quantity and the feature value requirement of exactly one product variant. Customer orders arrive in both stages of the two-stage stochastic program. The first stage models the present. Order quantities and feature value requirements of customer orders that arrive in the present are certain. The second stage models the future. Order quantities and feature value requirements of customer orders that arrive in the future are uncertain, which is modelled by scenarios in the second stage. Reorders in the future of product variants that have been ordered in the present are uncertain too and are therefore included in the second stage.

First-stage decision variables design product platforms in the present. Every product variant that is ordered in the present has to be served by one product platform that is designed in the present. Such platforms can be designed to serve product variants that are expected to be ordered in the future. This often increases platform manufacturing costs as in every feature, the most demanding of the served product variants dictates the specification of the platform. In case an expected future customer order is not served by a first-stage design, the recourse decision remains to design additional product platforms in the second stage.

We assume that two different values of the same feature have a different cost contribution, i.e., the cost contribution is strictly increasing over the by cost contribution sorted set of feature values. Due to this strict monotonicity, the feature values of a minimum-cost product platform design have to attain the lower bound of the tolerance interval of at least one of the served product variants. If the optimal platform design includes a feature value that is not the lower bound of a tolerance interval of one of the served product variants, then the manufacturing costs of the design could be reduced by replacing this feature value with the minimum requirement of the most demanding of the served product variants and, hence, cannot be optimal.

The strict monotonicity implies that for every feature, the mapping between feature value and cost contribution is unambiguous. We can therefore use the cost contribution per feature as a proxy for feature values in the modelling of platform design decisions. Furthermore, it is sufficient to consider only the lower bounds and upper bounds of tolerance intervals in the TPPDP formulation. Optimal product platform specifications consist only of lower bounds. Upper bounds are needed to define the feasibility area of feature values.

The cost contributions of features are assumed to be independent and additive. Neither the specification of other features nor the served product variant has an effect on the cost contribution of the feature value that is realized in a product platform. This is in particular a valid assumption for the semiconductor silicon wafer industry as wafers have homogeneous dimensions. Wafers of different diameter would define separate product platform design problems and the production of one wafer always requires one slice of an ingot.

We divide the set of features into the subsets F^{D} and F^{C}. Design decisions of the features in F^{D} are modelled by discrete decision variables. Design decisions of the features in F^{C} are modelled by continuous decision variables. A feature belongs to F^{D} if the feature value requirement of at least customer order defines two or more intervals on the by cost contribution sorted union set of feasible feature values. Otherwise, every feature value requirement defines exactly one interval on the by cost contribution sorted union set of feasible feature values and the feature belongs to F^{C}.

In the following, we present the MILP formulation of the TPPDP. Order quantities in the first stage are assumed to be positive. Second stage demand is assumed to be zero or greater than zero. Given the limited time horizon for which information about potential future customer orders is available, we neglect the time-value of money.

TPPDP formulation

We first introduce the notation that is used in the MILP formulation of the TPPDP and begin with the definition of sets. For the sake of brevity, we refer to product platform as platform and to product variant as variant:

R^1 Set of variant types that customers order in the first stage.

R^2 Set of variant types that customers order for the first time in the second stage.

P^1 Set of platform types that are designed in the first stage.

P^2 Set of platform types that are designed in the second stage.

F^{C} Set of platform features that customer orders specify with a single interval on the by cost contribution sorted union set of feasible feature values.

F^{D} Set of platform features that customer orders specify with one or more intervals on the by cost contribution sorted union set of feasible feature values.

F Union set of all platform features ($F := F^{\mathrm{C}} \cup F^{\mathrm{D}}$).

V_f Set of values of feature $f \in F^{\mathrm{D}}$ that form the lower bound of any of the intervals defined by any of the customer orders.

V^1_{rf} Subset of V_f containing values that lie within the tolerance intervals defined by the customer order for variant $r \in R^1$.

V^2_{rfs} Subset of V_f containing values that lie within the tolerance intervals defined by the customer order for variant $r \in R^2$ in second-stage scenario $s \in S$.

S Set of second-stage scenarios.

There cannot be more platform designs than customer orders, which makes $|P^1| := |R^1|$ and $|P^2| := |R^2|$ sufficient. Next, we define parameters:

D^1_r Number of platform units that are needed to meet the demand of variant $r \in R^1$ in the first stage.

D^2_{rs} Number of platform units that are needed to meet the demand of variant $r \in R^1 \cup R^2$ in scenario $s \in S$.

FC Fixed costs per platform type.

VC_{fv} Cost per platform unit of feature value $v \in V_f$.

UB^1_{rf} Cost per platform unit of feature value v^{UB1}_{rf}, which is the upper bound of the tolerance interval defined for feature $f \in F^{\mathrm{C}}$ of variant $r \in R^1$.

UB^2_{rfs} Cost per platform unit of feature value v^{UB2}_{rfs}, which is the upper bound of the tolerance interval defined for feature $f \in F^{\mathrm{C}}$ of variant $r \in R^2$ in scenario $s \in S$.

LB^1_{rf} Cost per platform unit of feature value v^{LB1}_{rf}, which is the lower bound of the tolerance interval defined for feature $f \in F^{\mathrm{C}}$ of variant $r \in R^1$.

LB^2_{rfs} Cost per platform unit of feature value v^{LB2}_{rfs}, which is the lower bound of the tolerance interval defined for feature $f \in F^{\mathrm{C}}$ of variant $r \in R^2$ in scenario $s \in S$.

Pr_s Probability of scenario $s \in S$.

M Sufficiently large value.

The formulation requires five types of decision variables, which are divided into first-stage and second-stage variables:

x^1_{rp} 1 if platform $p \in P^1$ serves variant $r \in R^1$ and 0 otherwise.

x^2_{rps} 1 if platform $p \in P^1 \cup P^2$ serves variant $r \in R^2$ in scenario $s \in S$ and 0 otherwise.

y^1_p 1 if platform $p \in P^1$ is active and 0 otherwise.

y^2_{ps} 1 if platform $p \in P^2$ in scenario $s \in S$ is active and 0 otherwise.

c^1_{pf} Cost of feature $f \in F$ per unit of platform $p \in P^1$.

c^2_{pfs} Cost of feature $f \in F$ per unit of platform $p \in P^2$ in scenario $s \in S$.

k^1_{rf} Cost of feature $f \in F^{\mathrm{C}}$ per unit of the platform that serves variant $r \in R^1$.

k^2_{rfs} Cost of feature $f \in F^{\mathrm{C}}$ per unit of the platform that serves variant $r \in R^2$ in scenario $s \in S$.

z^1_{rfv} 1 if the platform that serves variant $r \in R^1$ attains value $v \in V^1_{rf}$ in feature $f \in F^{\mathrm{D}}$ and 0 otherwise.

z^2_{rfvs} 1 if the platform that serves variant $r \in R^2$ in scenario $s \in S$ attains value $v \in V^2_{rfs}$ in feature $f \in F^{\mathrm{D}}$ and 0 otherwise.

$$\zeta = \min \sum_{r \in R^1} \sum_{f \in F} D^1_r k^1_{rf} + \sum_{p \in P^1} FC y^1_p + \sum_{s \in S} Pr_s Q(c^1_{pf}, k^1_{rf}, y^1_p, s) \tag{2.1}$$

$$\sum_{p \in P^1} x^1_{rp} \geq 1 \qquad r \in R^1 \tag{2.2}$$

$$x^1_{rp} \leq y^1_p \qquad r \in R^1, p \in P^1 \tag{2.3}$$

$$UB^1_{rf} + M(1 - x^1_{rp}) \geq c^1_{pf} \qquad r \in R^1, p \in P^1, f \in F^{\mathrm{C}} \qquad (2.4)$$

$$LB^1_{rf} x^1_{rp} \leq c^1_{pf} \qquad r \in R^1, p \in P^1, f \in F^{\mathrm{C}} \qquad (2.5)$$

$$k^1_{rf} + M\left(1 - x^1_{rp}\right) \geq c^1_{pf} \qquad r \in R^1, p \in P^1, f \in F^{\mathrm{C}} \qquad (2.6)$$

$$\sum_{v \in V^1_{rf}} z^1_{rfv} = 1 \qquad r \in R^1, f \in F^{\mathrm{D}} \qquad (2.7)$$

$$\sum_{v \in V^1_{rf}} VC_{fv} z^1_{rfv} + M\left(1 - x^1_{rp}\right) \geq c^1_{pf} \qquad r \in R^1, p \in P^1, f \in F^{\mathrm{D}} \qquad (2.8)$$

$$\sum_{v \in V^1_{rf}} VC_{fv} z^1_{rfv} - M\left(1 - x^1_{rp}\right) \leq c^1_{pf} \qquad r \in R^1, p \in P^1, f \in F^{\mathrm{D}} \qquad (2.9)$$

$$\sum_{v \in V^1_{rf}} VC_{fv} z^1_{rfv} \leq k^1_{rf} \qquad r \in R^1, f \in F^{\mathrm{D}} \qquad (2.10)$$

$$c^1_{pf}, k^1_{rf}, y^1_p \in \mathbb{R}_{\geq 0}$$

$$x^1_{rp}, z^1_{rfv} \in \{0, 1\}$$

The objective function (2.1) of the TPPDP formulation minimizes the total of manufacturing costs and fixed costs of platforms that are designed in both the first stage and the second stage. The latter is modelled by the sum of recourse functions Q multiplied with scenario probabilities Pr_s. The optimal objective function value is denoted as ζ. Constraint (2.2) ensures that every variant that is ordered in the first stage is also served with a platform that is designed in the first stage. With constraint (2.3), every platform that is designed in the first stage will also incur fixed costs in the first stage. Constraints (2.4) and (2.5) make sure that the specification of the platform complies with the requirements of the served variants for the features in F^{C}. Constraint (2.6) propagates the manufacturing costs per feature and per platform unit to the served variants so that the total manufacturing costs can be calculated in the objective function by multiplication with demand. Constraint (2.7) ensures that the platform features in F^{D} are specified by enforcing the selection of one of the feasible values in V^1_{rf}. Constraints (2.8) and (2.9) make sure that the specification of the platform complies with the requirements of the served variants for the features in F^{D}. Constraint (2.10) propagates the manufacturing costs of the features in F^{D} to the served variants, which allows the calculation of total manufacturing costs in the objective function.

$$Q(c^1_{pf}, k^1_{rf}, y^1_p, s) = \sum_{r \in R^1} \sum_{f \in F} D^2_{rs} k^1_{rf} + \sum_{r \in R^2} \sum_{f \in F} D^2_{rs} k^2_{rfs} + \sum_{p \in P^2} FC y^2_{ps} \qquad (2.11)$$

$$\sum_{p \in P^1 \cup P^2} x^2_{rps} \geq 1 \qquad \forall r \in \{r : r \in R^2 \wedge D^2_{rs} > 0\} \qquad (2.12)$$

$$x^2_{rps} \leq \begin{cases} y^1_p & \forall r \in R^2, p \in P^1 \\ y^2_{ps} & \forall r \in R^2, p \in P^2 \end{cases} \qquad (2.13)$$

$$UB^2_{rfs} + M(1 - x^2_{rps}) \geq \begin{cases} c^1_{pf} & \forall r \in R^2, p \in P^1, f \in F^{\mathrm{C}} \\ c^2_{pfs} & \forall r \in R^2, p \in P^2, f \in F^{\mathrm{C}} \end{cases} \tag{2.14}$$

$$LB^2_{rfs} x^2_{rps} \leq \begin{cases} c^1_{pf} & \forall r \in R^2, p \in P^1, f \in F^{\mathrm{C}} \\ c^2_{pfs} & \forall r \in R^2, p \in P^2, f \in F^{\mathrm{C}} \end{cases} \tag{2.15}$$

$$k^2_{rfs} + M\left(1 - x^2_{rps}\right) \geq \begin{cases} c^1_{pf} & \forall r \in R^2, p \in P^1, f \in F^{\mathrm{C}} \\ c^2_{pfs} & \forall r \in R^2, p \in P^2, f \in F^{\mathrm{C}} \end{cases} \tag{2.16}$$

$$\sum_{v \in V^2_{rfs}} z^2_{rfvs} = 1 \qquad \forall r \in R^2, f \in F^{\mathrm{D}} \tag{2.17}$$

$$\sum_{v \in V^2_{rfs}} VC_{fv} z^2_{rfvs} + M\left(1 - x^2_{rps}\right) \geq \begin{cases} c^1_{pf} & \forall r \in R^2, p \in P^1, f \in F^{\mathrm{D}} \\ c^2_{pfs} & \forall r \in R^2, p \in P^2, f \in F^{\mathrm{D}} \end{cases} \tag{2.18}$$

$$\sum_{v \in V^2_{rfs}} VC_{fv} z^2_{rfvs} - M\left(1 - x^2_{rps}\right) \leq \begin{cases} c^1_{pf} & \forall r \in R^2, p \in P^1, f \in F^{\mathrm{D}} \\ c^2_{pfs} & \forall r \in R^2, p \in P^2, f \in F^{\mathrm{D}} \end{cases} \tag{2.19}$$

$$\sum_{v \in V^2_{rfs}} VC_{fv} z^2_{rfvs} \leq k^2_{rfs} \qquad \forall r \in R^2, f \in F^{\mathrm{D}} \tag{2.20}$$

$$c^2_{pfs}, k^2_{rfs}, y^2_{ps} \in \mathbb{R}_{\geq 0}$$

$$x^2_{rps}, z^2_{rfvs} \in \{0, 1\}$$

The recourse function (2.11) models the second-stage response in scenario s to the decisions made in the first stage. In addition to the manufacturing costs incurred by reorders of platforms that have been designed in the first stage, the recourse function also includes the manufacturing costs and fixed costs of platforms that are designed in the second stage. The second-stage constraints (2.12) to (2.20) have a equivalent meaning as their first-stage counterparts (2.2) to (2.10). The difference is that they are defined for variants that are ordered in a second-stage scenario. These variants can be served with a platform that is either designed in the first stage ($p \in P^1$) or in the second-stage scenario in which the variant is ordered ($p \in P^2$). Constraint (2.12) is not defined in case second-stage demand is zero in order to avoid that fixed costs are incorrectly added to the objective function.

The efficiency gain of the presented model compared to Boysen and Scholl (2009) is due to the features in F^{C}. Design decisions for the features in F^{C} are modelled by the continuous decision variables c^1_{pf} and c^2_{pfs}. In contrast, values of the features in F^{D} are determined by the binary decision variables z^1_{rfv} and z^2_{rfvs}, which is similar to the modelling approach proposed by Boysen and Scholl. The number of the binary design variables z^1_{rfv} and z^2_{rfvs} is bounded by $n(R)n(F)n(V)n(S)$ where $n(R) \coloneqq |R^1 \cup R^2|$ is the number of ordered variants, $n(F) \coloneqq |F|$ is the number of features, $n(V) \coloneqq \max_{f \in F}\{|V_f|\}$ is the maximum number of feature values, and $n(S) \coloneqq |S|$ is the number of second-stage scenarios. A platform feature either belongs to F^{C} or to F^{D}. The more features belong to F^{C}, the greater is the efficiency gain compared to Boysen and Scholl as z^1_{rfv} and z^2_{rfvs} are not used for these features. Most of the features

that specify silicon ingots (e.g., electrical resistivity, the number of oxidation-induced stacking faults, the carrier diffusion length and lifetime, and the concentration of carbon, oxygen, and iron) have indeed single-interval tolerance intervals and belong to F^{C}. The features grow material type and silicon grade have tree-like feature value substitution rules, which requires discrete feature value modelling and lets these features belong to F^{D}.

2.4 Numerical experiments

Tactical product platform design can reduce the total costs of designing and manufacturing product platforms in the present and in the future. Several factors, including the feature value requirement and the order quantity of customer orders, are expected to influence the effectiveness of the proposed TPPDP formulation. We discover the factors that affect tactical product platform design in experiments that are inspired by settings found in silicon wafer manufacturing. The tactical design of ingots as product platforms is intended to postpone the customer order decoupling point (see Figure 2.1). In Section 2.4.1, we first describe the assumptions that are made in problem instance generation. In Section 2.4.2, we present both the experimental design and the numerical results that relate effectiveness to certain factor level combinations. In Section 2.4.3, we discuss an additional experiment that is conducted in order to measure the effect of the proposed modelling approach on computation time.

2.4.1 Generation of problem instances

The generated problem instances describe settings, in which one variant type is ordered in the first stage and a few other variant types are expected to be ordered in the second stage. This is a realistic setting because the TPPDP will be initialized every time an ordered variant requires the design of a new platform. Extreme differences between the feature value requirements of variants push over-costs and make a common platform financially unfavourable. That is why from the set of all variants that are expected to be ordered in the near future, only those that are similar enough to the variant that is ordered in the present have to be included in the second stage of the TPPDP formulation.

We assume that the list of features can be reduced to the features that contribute the most to manufacturing costs. The remaining features are assumed to be inexpensive so that the platform can be designed to any feature value requirement at low over-costs. In our experiments, five product platform features are modelled and every feature has 40 possible feature values including the null value, which models that the feature is not realized. The cost per platform unit of the null value is zero. The cost per platform unit of any other feature value v is defined as the vth partial sum $\sum_{i=1}^{v} a_i$, with discrete random variable a_i, which has three equally likely realizations: \$0.1, \$0.5, and \$0.9. The cost contribution of any feature to the cost per platform unit is thus a strictly increasing function of the feature values

$v = 0, 1, \ldots, 39$. Note that a platform unit represents a slice of an ingot and an ingot can yield more than 2000 slices, depending on ingot length and final wafer thickness. Such a slice is processed to a wafer in downstream processing steps, including edge rounding, lapping, and cleaning (see Figure 2.1), in order to become one of the ordered wafers, i.e., product variants.

For simplicity, we assume that feature value requirements only define lower bounds for feature values and upper bounds are assumed to be infinite. Note that the proposed formulation is ready to consume upper bounds. The feature value requirement of the variant $r \in R^1$ that is ordered in the first stage is $v_{rf}^{\mathrm{LB1}} = 16$ across all features f and its first-stage demand is $D_r^1 = 1000$ platform units. The feature value requirements and the demand of second-stage orders are uncertain. It is assumed that pending orders, ongoing negotiations, and technological trends can be used to define scenarios that represent probable future realizations. In our experiments, the feature value requirement v_{rfs}^{LB2} of variant $r \in R^2$ in second-stage scenario $s \in S$ is modelled as a random number that is drawn from the normal distribution $\mathcal{N}(\mu_r^{\mathrm{FV}}, \sigma^{\mathrm{FV}})$ and rounded to the nearest integer. The mean feature value requirement of variant $r \in R^2$, μ_r^{FV}, is also modelled as a normally distributed random number following $\mathcal{N}(\mu^{\mathrm{FM}}, \sigma^{\mathrm{FM}})$. Second-stage order quantities D_{rs}^2 are likewise assumed to be normally distributed following $\mathcal{N}(\mu^{\mathrm{D}}, \sigma^{\mathrm{D}})$. Distributional assumptions are used to generate $|S|$ equiprobable scenarios that allow us to solve the TPPDP using sample average approximation.

2.4.2 Effectiveness of the TPPDP formulation

Design of experiment

An experiment is conducted in order to discover the effect of six factors on the ability of the TPPDP formulation to reduce costs. The considered factors are the growth, the variability, and the uncertainty of the feature value requirement of second-stage orders, the growth and the uncertainty of second-stage order quantities, and the costs of designing a product platform. The cost of designing a product platform is assumed to be a fixed cost and is modelled by parameter FC. The growth and the uncertainty of second-stage demand are modelled by parameters μ^{D} and σ^{D}. The growth, the variability, and the uncertainty of the feature value requirement of second-stage orders are modelled by parameters μ^{FM}, σ^{FM}, and σ^{FV}, respectively.

We define two levels per factor (see Table 2.1). The fixed costs assumption is based on our observation that the design of a new silicon wafer, which includes the design of an ingot, is an iterative process that can take a few months and involves multiple business functions including sales, product design, engineering, and production. The demand per variant in the second stage is either the same as the first-stage demand, which is 1000 platform units, or

Table 2.1: Factor levels.

Factor	Description	Unit	Factor level 1	Factor level 2
FC	Fixed costs	[\$]	70,000	110,000
μ^{D}	Demand growth	[platform units]	1,000	2,000
σ^{D}	Demand uncertainty	[platform units]	100	200
μ^{FM}	Feature value growth	[feature value]	16	20
σ^{FM}	Feature value variability	[feature value]	1.6	3.2
σ^{FV}	Feature value uncertainty	[feature value]	1.6	3.2

twice as much. The factor levels for the distribution parameters μ^{FM}, σ^{FM}, and σ^{FV} result in an average cost per platform unit of around \$46.2 (before solving the TPPDP), which is a realistic cost per wafer depending on size and quality. Standard deviations are defined as either 10 % or 20 % of the corresponding level 1 mean.

We observe three response variables: The expected number of second-stage orders that are served by the first-stage platform design η, the relative value of two-stage modelling $relVTSM$, and the relative value of the stochastic solution $relVSS$. η is formally defined as:

$$\eta := \sum_{r \in R^2} \sum_{p \in P^1} \sum_{s \in S} Pr_s x_{rps}^2 \tag{2.21}$$

$relVTSM$ represents the expected cost increase if, instead of the proposed two-stage model, a single-stage model would optimize the two decision stages of the TPPDP sequentially. $relVTSM$ is computed by first optimizing the first stage of the TPPDP independently of second-stage scenarios. The resulting first-stage solution is then hard-coded in the original TPPDP formulation, allowing second-stage decisions to be chosen optimally. We denote the optimal objective function value as SEQ and define $relVTSM$ as:

$$relVTSM := \frac{SEQ - \zeta}{\zeta} \tag{2.22}$$

with ζ standing for the objective function value of the optimal solution of the TPPDP formulation presented in Section 2.3.2.

The value of stochastic solution quantifies the expected cost increase if expected values would be used instead of weighted scenarios in order to represent uncertainty in the second stage of the TPPDP formulation. Let EEV be obtained by solving the deterministic program, in which the random numbers are replaced by their expected values μ^{D} and μ^{FM}. The resulting first-stage solution is then hard-coded in the original TPPDP formulation, allowing second-stage decisions to be chosen optimally. We define $relVSS$ in accordance with Birge and Louveaux (2011) as:

$$relVSS := \frac{EVV - \zeta}{\zeta} \tag{2.23}$$

Every problem instance contains $|S|= 25$ randomly generated second-stage scenarios. Every scenario assumes that three variant types are ordered in the second stage and describes a random realization of both demand and feature value requirement of these three variants. All features are assumed to belong to F^{C} in order to reduce computation times. The experiment is a 2^6 full factorial design. Every treatment combination is replicated 45 times, which results in 2,880 independent problem instances. Proper variation and coordination of random number streams are implemented.

Results

An analysis of variance (ANOVA) is conducted for η, *relVTSM*, and *relVSS*. We have verified that the assumptions of error term normality and homogeneity of variance are not violated in any ANOVA. The results are that all main effects, except for the effect of demand uncertainty, and several two-way interaction effects are significant at the 99 % confidence level (see Appendix A.1). The quality of optimal solutions is measured for every treatment combination by the approximate 95.0 % confidence interval for the true optimum of total costs ζ^*, which is on average ± 1.2 % of treatment mean $\overline{\zeta}$ (see Appendix A.2). It shows that $|S|= 25$ can be considered as a sufficient sample size in this experiment.

Table 2.2 presents the grand means of η, *relVTSM*, and *relVSS*. On average 1.58 out of three second-stage customer orders are expected to be served by the first-stage platform design. The grand means of the relative value of two-stage modelling and of the relative value of stochastic solution are 3.39 % and 6.91 %, respectively. Since the grand mean of total costs is \$503,992 (see Appendix A.2), this corresponds to on average \$17,072 and \$34,820. In comparison to sequential optimization, the TPPDP formulation generates a substantial expected saving as it prevents second-stage customer orders from entering the platform design phase. The value of stochastic solution is even greater because designing first-stage platforms to the expected values of second-stage feature value requirements does incur over-costs, whereas platform design costs are not avoided as effectively.

A closer look at the main effects in Table 2.2 provides additional insights. The TPPDP formulation performs better than sequential problem solving because it reduces the expected number of platforms that have to be designed in the second stage. The greater the fixed costs (FC), the greater is the expected saving of the TPPDP formulation, which explains the increase of both $\overline{relVTSM}$ and $\overline{relVSS}$. Greater fixed costs also compensate greater over-costs, which explains the increase of the mean number of second-stage orders served by the first-stage platform design ($\overline{\eta}$). Low demand growth (μ^{D}) and a low feature value growth (μ^{FM}) reduce the over-costs incurred by the first-stage platform design and thus increase the net saving of tactical product platform design. High feature value variability (σ^{FM}) and high feature value uncertainty (σ^{FV}) result in fewer opportunities to design product platforms for future orders. This is because in every feature, the requirement of the most demanding of the served variants dictates the specification of the platform. Greater variability and

Table 2.2: Means by factor level and grand means.

Factor	Level	$\overline{\eta}$	$\overline{relVTSM}$ [%]	$\overline{relVSS}$ [%]
FC	70,000	1.11*	1.35*	4.93*
	110,000	2.04	5.43	8.89
μ^{D}	1,000	2.15*	5.72*	9.59*
	2,000	1.00	1.05	4.23
σ^{D}	100	1.58	3.38	6.90
	200	1.58	3.39	6.91
μ^{FM}	16	1.91*	4.61*	6.53*
	20	1.25	2.16	7.29
σ^{FM}	1.6	1.67*	3.93*	7.19*
	3.2	1.48	2.84	6.63
σ^{FV}	1.6	2.09*	5.13*	8.63*
	3.2	1.06	1.64	5.18
Grand mean		1.58	3.39	6.91

* significant main effect (P-value < 0.0001).

uncertainty therefore increase the chance that customer orders exist that push the feature value requirement and thus over-costs to a prohibitive level. An analysis of the interaction effects on *relVTSM* supports these results because significant interaction effects are ordinal and involve a mutual amplification of main effects (see Appendix A.3).

2.4.3 Computational efficiency of the TPPDP formulation

A second experiment is conducted in order to measure the effect of both problem size and feature type on computation time. Problem size is controlled by varying the number of second-stage scenarios $|S|$ between 2 and 16. The effect of feature type is explored by changing the cardinality of F^{D}. Since there exist two features in silicon wafer manufacturing (i.e., grow material type and silicon grade) that usually follow tree-like feature value substitution rules, we vary the cardinality of F^{D} between zero and two. The total number of features is, as in the previous experiment, five and features that are not in F^{D} are in F^{C}. In order to reduce the number of treatment combinations, we set each of the factors presented in Table 2.1 to the average of the two factor levels. To make the problems computationally harder, the second stage models five instead of three new orders for product variants.

We observe the computation time of both the proposed TPPDP formulation and a reference model, which implements the modelling approach of Boysen and Scholl (2009) (see Appendix A.4). All computations are performed by the MILP solver of IBM ILOG CPLEX 12.71 on an Intel Xeon CPU E3-1220 V2 at 3.1 GHz clock speed with four cores and 32 GB memory while no other application is running. The experiment is replicated 45 times. The resulting mean computation times of solving problem instances to a MIP gap of 1.5 % are presented in Table 2.3.

Table 2.3: Mean computation times in seconds.

	TPPDP formulation			
$\|S\|$	$\|F^{\mathrm{D}}\|=0$	$\|F^{\mathrm{D}}\|=1$	$\|F^{\mathrm{D}}\|=2$	Reference model
2	0.2	0.4	0.9	113.0
4	0.4	1.7	3.1	>25,000.0
8	4.7	16.7	33.9	>25,000.0
12	125.7	194.3	457.3	>25,000.0
16	1,173.3	1,394.2	1,941.5	>25,000.0
Grand mean	260.9	321.4	487.4	

Increasing the cardinality of F^{D} in the TPPDP formulation from zero to one and from one to two increases the computation time on average by 23.23 % and 51.61 %, respectively. The computation time of the TPPDP formulation grows exponentially in $|S|$. This is because the TPPDP is NP-hard. (Note that the uncapacitated facility location problem, which is NP-complete (see, e.g., the min-sum multicenter problem in Garey and Johnson, 1979), is reducible to the TPPDP as facility assignment decisions can be transformed to platform assignment decisions.) Nevertheless, the proposed TPPDP formulation provides a significant computation time reduction compared to the reference model, which adapts Boysen and Scholl's approach for modelling deterministic platform design decisions to the stochastic situation (see Appendix A.4). The reference model can only be solved for problem instances with two second-stage scenarios. The TPPDP formulation in contrast can be solved for all instances and for the instances with two second-stage scenarios in less than 1 % of the time that is needed by the reference model.

2.5 Conclusion

We propose a two-stage stochastic programming formulation to solve the tactical product platform design problem. The formulation designs product platforms to the requirements of present and expected future customer orders by trading off a decrease of design costs against an increase of manufacturing costs. The explicit consideration of expectations about future customer orders increases the probability that product platforms that are designed in the present will be suitable to serve future customer orders too. This reduces platform design workload and effectively postpones the customer order decoupling point.

In numerical experiments with randomly generated problem instances that reflect settings in the semiconductor silicon wafer industry, 52.5 % of the expected future customer orders are on average served by the product platform that is designed in the present. The total costs of the traditional platform design approach, which involves the sequential solution of two single-stage platform design problems, is on average 3.39 % more expensive than tactical

product platform design. The average relative value of stochastic solution is even greater, i.e., on average 6.91 % of the total costs of tactical product platform design. This shows that using stochastic optimization for tactical product platform design is an effective means to reduce costs.

The proposed formulation avoids binary decision variables in the modelling of design decisions if feature value requirements are specified by a tolerance interval. As a result, the proposed formulation requires fewer decision variables than a reference approach developed for the deterministic problem. This leads to a decrease of computation times of several orders of magnitude, permitting an efficient solution also of our stochastic and hence computationally demanding model formulation. However, the TPPDP is NP-hard and computation times grow exponentially in problem size. An opportunity for future research is therefore to solve the tactical product platform design problem with tailored solution approaches such as branch-and-price.

Tactical product platform design is financially beneficial because it reduces the product design workload. Numerical experiments confirm the greatest benefit in environments with high design costs and low manufacturing over-costs caused by over-design. Low order quantities, a low feature value requirement growth, homogeneous customer orders, and little uncertainty about future feature value requirements favour low over-costs. This makes tactical product platform design in particular interesting for ETO companies that have customers in mature high-tech industries, such as semiconductor device manufacturing for the automotive industry. Steady incremental innovation, product proliferation and standardization limit over-costs, while a labour shortage makes highly specialized product design skills rare and thus product platform design expensive. This study shows that tactical product platform design can reduce total costs in such environments significantly.

Chapter 3

Cycle time-oriented mid-term production planning for semiconductor wafer fabrication

Published as:
Kriett, P. O., Eirich, S., and Grunow, M. (2017). Cycle time-oriented mid-term production planning for semiconductor wafer fabrication. *International Journal of Production Research*, 55(16):4662–4679

Abstract

Wafers are produced in an environment with uncertain demand and failure-prone machines. Production planners have to react to changes of both machine availability and target output, and revise plans appropriately. The scientific community mostly proposes WIP-oriented mid-term production planning to solve this problem. In such approaches, production is planned by defining targets for throughput rates and buffer levels of selected operations. In industrial practice, however, cycle time-oriented planning is often preferred over WIP-oriented planning. We therefore propose a new linear programming formulation, which facilitates cycle time-oriented mid-term production planning in wafer fabrication. This approach plans production by defining release quantities and target cycle times up to selected operations. It allows a seamless integration with the subordinate scheduling level. Here, least slack first scheduling translates target cycle times into lot priorities. We evaluate our new methodology

in a comprehensive simulation study. The results suggest that cycle time-oriented mid-term production planning can both increase service level and reduce cycle time compared to WIP-oriented planning. Further, it requires less modelling effort and generates plans, which are easier to comprehend by human planners.

3.1 Introduction

Semiconductor wafer fabrication is the creation of electronic integrated circuits (ICs) through layer-by-layer treatment of the surface of a circular slice of monocrystalline silicon (wafer). High volume wafer fabrication facilities (fabs) can accommodate hundreds of machines, which fabricate ICs according to distinct process flows. A process flow describes a sequence of up to 800 process steps, also called operations. It starts with the release of bare wafers into the fab and ends with the arrival of finished wafers at the finished wafer stock (FWS). Wafer fabrication is recognised as a complex endeavour, which is in particular because it defines a job shop environment with recirculating process flows and various process constraints, both production and development on the same equipment, and both stochastic equipment downtime and yield (Atherton and Atherton, 1995; Uzsoy et al., 1992).

A high throughput is necessary to achieve competitive unit costs and amortize expensive wafer fabrication equipment. At the same time, customers evaluate their IC suppliers based on lead time and on-time delivery. However, an accumulation of substantial inventories, which would enable such customer response, must be avoided because of the resulting risk of obsolescence that is due to short product life cycles, strong competition, and the high volatility of the semiconductor market. As a result, short cycle times, i.e., short sojourn times of wafers between release and arrival at the FWS, are key in this industry. They allow short customer lead times, an effective response to demand changes, and a quick development and ramp-up of new ICs. To avoid contractual penalties or a loss of customer goodwill, it is necessary to plan and schedule wafer fabrication in such a way that late deliveries are minimised while achieving the desired throughput and while keeping cycle times short.

3.1.1 Hierarchical production planning and scheduling in wafer fabrication

Hierarchical production planning and scheduling is a widely used concept (see, e.g., Hopp and Spearman (2011) and Missbauer and Uzsoy (2011)). Hierarchically related decision levels differ in objective, scope, and level of aggregation. The solution of an upper level decision problem is integrated into lower level decision problems through constraints. In wafer fabrication, several studies suggest the use of an upper tactical, a lower tactical, and a scheduling level (Bard et al., 2010; Cai et al., 2011; Govind et al., 2008; Hwang and Chang, 2003; Leachman et al., 2002; Sawik, 2006).

The upper tactical decision level corresponds to company-wide master production planning. The resulting master production schedule usually defines weekly production targets for each plant (including the wafer fab) over a planning horizon of up to 52 weeks. For a given demand, the objective of master production planning is to offer as early delivery dates as possible and to guarantee that promised deliveries are fulfilled on time. Constraints reflect the promised delivery dates of committed orders, expected plant capacity, expected plant cycle time, initial WIP, and the initial inventory including FWS.

The lower tactical decision level is called mid-term production planning (Bard et al., 2010; Hwang and Chang, 2003). The objective of mid-term production planning is to minimise the deviation of actual fab output from the master production schedule and to keep cycle time short. Constraints reflect the master production schedule, the expected capacity of bottleneck machines, the expected cycle time between bottleneck operations, initial WIP, and initial FWS.

The scheduling level maximises throughput and fulfils the mid-term production plan. In practice, there is usually a main scheduling policy in place, such as least-slack first. Depending on the complexity of process constraints and the scarcity of a particular resource, the main scheduling policy either dispatches lots directly or provides priorities to machine-specific scheduling algorithms.

3.1.2 Mid-term production planning in wafer fabrication

Mid-term production planning takes advantage of plant-wide WIP tracking and optimisation capabilities to provide local schedulers with tactical priority changes. The scope of scheduling problems in wafer fabrication is usually restricted to a single or a small number of work centres. That means scheduling problems are myopic and solved in a decentralised way. Restricting the scope makes scheduling problems solvable within an acceptable time but neglects information that is necessary to make dispatch decisions leading to better results on the fab level. Through prioritisation and deprioritisation, mid-term production planning aligns decentralised scheduling with fab objectives and hence with company objectives. The specific planning decisions that effectuate priority changes depend on the mid-term production planning approach that is used.

Leachman et al. (2002) point out that the planning of wafer fabrication either follows the WIP-management paradigm, i.e., is WIP-oriented, or the lot-dispatching paradigm, i.e., is cycle time-oriented (CT-oriented). Following the WIP-management paradigm means that throughput and WIP are decisions, while cycle time is a result. Following the lot-dispatching paradigm means that throughput and cycle time are decisions, while WIP is a result.

The origin of WIP-oriented planning lies in material requirements planning (see, e.g., Vollmann et al., 2005). General purpose linear programming (LP) formulations for WIP-oriented production planning have been proposed for example by Billington et al. (1983) and Hackman and Leachman (1989). A number of WIP-oriented mid-term production planning

models have been developed specifically for semiconductor manufacturing. Leachman and Carmon (1992), Kim and Leachman (1994) and Cai et al. (2011) suggest linear programming formulations. Hwang and Chang (2003) present an integer programming formulation and a solution method based on Lagrangian relaxation. For complexity reasons, many studies propose heuristic approaches, such as the decomposition of large-scale LP formulations into sub-problems (Bard et al., 2010), a work centre-based decomposition according to the shifting bottleneck procedure (Barua et al., 2005; Sourirajan and Uzsoy, 2007), and heuristic scheduling algorithms (Jula and Leachman, 2008; Kim and Leachman, 1994; Leachman et al., 2002).

The planning of wafer fabrication in fabs of Infineon Technologies, e.g., at Dresden (Germany), is CT-oriented and not WIP-oriented. Every lot of wafers receives a set of operation due dates (ODDs) at the point of release. The set of ODDs is defined by the release date of the lot plus target cycle times up to bottleneck operations along the process flow. Cycle time is controlled effectively with ODDs because schedulers generally dispatch the lot with the earliest ODD first. This balances lateness across the lots. Not only fab operators but also the highly influential study of Lu et al. (1994) suggests simplicity and good cycle time performance as the key advantages of an ODD-based least slack scheduling policy.

Despite these industry requirements, all of the published mid-term production planning models for wafer fabrication that have been proposed in scientific literature follow the WIP-management paradigm. A CT-oriented mid-term production planning model has not yet been suggested.

3.1.3 Problem statement and scientific contribution

We consider a production environment in which wafer fabrication is driven by ODDs, such as it is practice in industry. The fab has to fulfil a master production schedule on time, while the master production schedule as well as the machine availability are subject to uncertainty. The addressed planning problem is to determine how many lots have to be released during the next planning period and which target cycle times, i.e., which set of ODDs, have to be assigned to each lot (including both new releases and initial WIP) such that both cycle time and the deviation of fab output from the master production schedule are minimised.

The scientific contributions of this study include

- the identification of a gap between scientific literature and industry requirements in terms of the underlying planning paradigms for mid-term production planning,
- a new optimisation-based methodology to accomplish CT-oriented mid-term production planning and its integration with hierarchical planning and scheduling, and
- an experimental proof of the superiority of CT-oriented planning over WIP-oriented planning for a reference case from semiconductor manufacturing.

In Section 3.2, we present the new LP formulation of the stated CT-oriented mid-term production planning problem and discuss its integration with the scheduling level. To evaluate our approach, we benchmark it against the commonly used WIP-oriented mid-term production planning, for which we present an LP formulation in Section 3.3. We evaluate both formulations in a rolling horizon framework that includes a discrete event simulation model of a reference wafer fab, which is described in Section 3.4. The design of experiments and the numerical results are presented in Section 3.5. The paper ends with our concluding remarks in Section 3.6.

3.2 CT-oriented mid-term production planning

We propose a new linear programming formulation, denoted as CT-LP, of the CT-oriented mid-term production planning problem. The objective is to minimize both the deviation of fab output from the output requirement and the amount of WIP, which directly affects mean cycle time. The CT-LP defines release quantities by product type and divides the lots of every product type in every segment into different classes of target cycle times that are denoted as priorities. Different classes of target cycle times are equivalent to different priorities and translate into different sets of ODDs. For example, a higher priority leads to tighter ODDs. The CT-LP does not provide WIP level targets to the scheduling level but these are a result of the management of cycle time via ODDs. The target cycle times depend on several factors, such as fab utilization, product mix, lot priority, and the number of prioritised lots. It is assumed that target cycle times are determined for example by simulation as it has been done by Asmundsson et al. (2006). A shortage of fab output is assumed to create a backlog instead of lost sales. Recall that finished wafers are consumed by the next manufacturing stage and not by the customer. After describing the CT-LP in Section 3.2.1, the integration of planning results into the scheduling level is explained in Section 3.2.2.

3.2.1 LP formulation

The indices and sets used for the CT-LP are:

$t \in \{1, 2, \ldots, T + L_{\max}\}$	Time period representing the time interval $(t-1, t]$.
$i \in \{1, 2, \ldots, I\}$	Product.
$j \in F_i = \{1, \ldots, J_i, J_i + 1\}$	Process flow of product i. $1, \ldots, J_i$ is the sequence of bottleneck operations. $J_i + 1$ models the FWS.
$k \in \{1, 2, \ldots, K\}$	Bottleneck work centre. A work centre is a group of identical machines running in parallel.
$p \in \{1, 2, \ldots, P\}$	Priority of a lot (low p means high priority).

We define the following parameters:

T	Length of planning horizon.
T_{WIP}	Maximum number of time periods by which WIP can be delayed.
L_i	Maximum cycle time of product i from its release until its arrival at the FWS given in time periods (smallest following integer). $L_{\max} := \max_i L_i$.
I	Total number of products.
J_i	Total number of bottleneck operations along the process flow of product i. Operation $J_i + 1$ represents the FWS.
K	Total number of bottleneck work centres.
P	Total number of priority classes.
$E_{i,j',\tau,p}$	Probability that a lot of product i with priority p initiates bottleneck operation $j' \in F_i$ in the τth period after the period in which it is released. Note that $\tau \in \{0, 1, \ldots, T + L_{\max}\}$ and $\sum_{\tau=0}^{T+L_{\max}} E_{i,j',\tau,p} = 1$.
$E_{i,j,j',\tau,p}$	Probability that a lot of product i with priority p initially in segment $[j-1, j)$ initiates bottleneck operation j' (with $j' \geq j$) in the τth period after the period in which it continues moving. Note that $j \in F_i$, $\tau \in \{0, 1, \ldots, T + L_{\max}\}$ and $\sum_{\tau=0}^{T+L_{\max}} E_{i,j,j',\tau,p} = 1$.
$WIP_{i,j}$	Number of lots of product i that are located in segment $[j-1, j)$ at the beginning of $t = 1$ with $j \in F_i$. $WIP_{i,j}$ models the initial WIP level between the bottleneck operations $j - 1$ (included) and j (excluded). The raw wafer stock is not considered.
$A_{k,i,j}$	Number of machine hours that bottleneck work centre k is occupied when it performs bottleneck operation $j \in \{1, 2, \ldots, J_i\}$ on a lot of product i.
$C_{k,t}$	Total machine hours that bottleneck work centre k is effectively available in period t.
$D_{i,t}$	Demand of product i in period $t \in \{1, 2, \ldots, T\}$, i.e., the fab output required by the master production schedule.
$\bar{D}_i$	Mean demand of product i.
V^{B}	Cost per period and per lot of backlog.
V^{D}	Cost per period and per lot of delayed WIP.

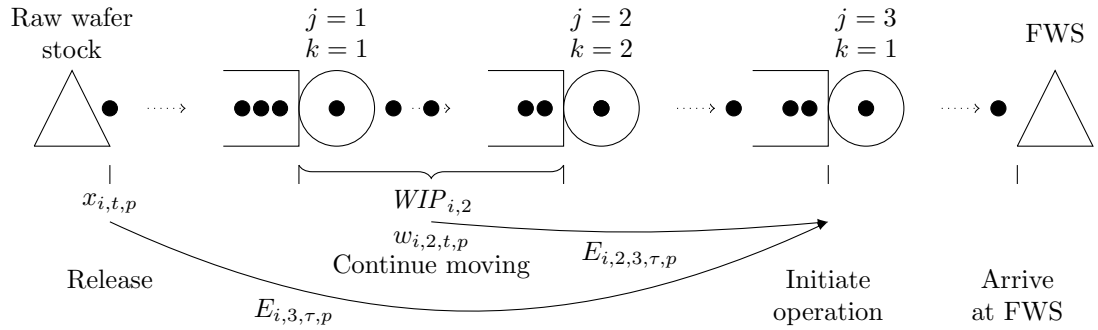


Figure 3.1: Modelling of WIP and cycle time in the CT-LP formulation.

V^{H} Cost per period and per lot of FWS.

V_p^{W} Cost per period and per lot of WIP with priority p.

UB_p Upper bound for the fraction of WIP with priorities $1, 2, \ldots, p$ for $p \in \{1, 2, \ldots, P-1\}$.

The following decision variables are continuous and greater than or equal to zero, i.e., in $\mathbb{R}_{\geq 0}$:

$x_{i,t,p}$ Number of lots of product i with priority p that are released into the fab in period t.

$w_{i,j,t,p}$ Number of lots of product i that are initially located in segment $[j-1, j)$ with $j \in F_i$ and that continue moving in period $t \in \{1, 2, \ldots, T_{\mathrm{WIP}}\}$ with priority p.

$h_{i,t}$ Number of lots of product i that are located in the FWS at the end of period t. The initial FWS level must be given as parameter $h_{i,0}$.

$b_{i,t}$ Backlog of lots of product i at the end of period t. The initial backlog must be given as parameter $b_{i,0}$.

$o_{i,t}^x, o_{i,t}^w$ Number of lots of product i that arrive at the FWS in period t. Arrivals that originate from releases are modelled by $o_{i,t}^x$. Arrivals that originate from initial WIP are modelled by $o_{i,t}^w$.

$c_{k,t}^x, c_{k,t}^w$ Number of machine hours that work centre k is occupied in period t. The capacity consumption of releases is modelled by $c_{k,t}^x$. The capacity consumption of initial WIP is modelled by $c_{k,t}^w$.

We assume that each process flow fabricates a single (representative) product. Figure 3.1 illustrates an example of a recirculating process flow, which passes two bottleneck work centres — one of them twice ($k = 1$). Process flows are divided into segments $[j-1, j)$. A segment is defined by the operations on two consecutive bottleneck work centres along the process flow.

A segment begins with the setup of the upstream bottleneck machine and ends at the setup of the downstream bottleneck machine, i.e., it includes the processing time of the upstream but not of the downstream operation. This allows us to define ODDs that are comparable even if different lots have different processing times on the same work centre.

The parameters $E_{i,j',\tau,p}$ and $E_{i,j,j',\tau,p}$ model the time needed for activities in one or more consecutive segments, which includes setup, loading, processing, unloading, travelling, and queuing time at both bottleneck and non-bottleneck work centres. $E_{i,j',\tau,p}$ and $E_{i,j,j',\tau,p}$ define probability mass functions of segment cycle time, which is measured in number of time periods. If a lot of product i with priority p is released in period $(t-1,t]$ or continues moving in segment $[j-1,j)$ at the beginning of this period, then $E_{i,j',\tau,p}$ and $E_{i,j,j',\tau,p}$, respectively, provide the probability with which the lot initiates operation j' in period $(t+\tau-1,t+\tau]$. This way of modelling cycle time is an extension of Kim and Kim (2001) and Missbauer (2002).

Figure 3.1 illustrates that the initial WIP of product i in segment $[j-1,j)$ is modelled by parameter $WIP_{i,j}$. $WIP_{i,1}$ models the number of lots between the release and the first bottleneck operation. WIP_{i,J_i+1} models the number of lots between the last bottleneck operation J_i and the arrival at the FWS. The decision variable $w_{i,j,t,p}$ decides whether initial WIP continues moving immediately or in a later period and with which priority. Decision variable $x_{i,t,p}$ models the number, type, and priority of released lots.

CT-LP formulation

$$\min \sum_{i=1}^{I}\sum_{t=1}^{T+L_{\max}}\sum_{p=1}^{P} V_p^{\mathrm{W}} x_{i,t,p} + \sum_{i=1}^{I}\sum_{j=1}^{J_i+1}\sum_{t=1}^{T_{\mathrm{WIP}}}\sum_{p=1}^{P} V_p^{\mathrm{W}} w_{i,j,t,p} + \sum_{i=1}^{I}\sum_{j=1}^{J_i+1}\sum_{t=2}^{T_{\mathrm{WIP}}}\sum_{p=1}^{P} tV^{\mathrm{D}} w_{i,j,t,p} + \sum_{i=1}^{I}\sum_{t=1}^{T+L_{\max}} V^{\mathrm{H}} h_{i,t} + \sum_{i=1}^{I}\sum_{t=1}^{T+L_{\max}} V^{\mathrm{B}} b_{i,t} \quad (3.1)$$

$$o_{i,t}^{x} + o_{i,t}^{w} + h_{i,t-1} + b_{i,t} = \begin{cases} D_{i,t} + h_{i,t} + b_{i,t-1} & \forall i, \forall t \le T \\ \bar{D}_i + h_{i,t} + b_{i,t-1} & \forall i, \forall t > T \end{cases} \quad (3.2)$$

$$x_{i,t,P} = \bar{D}_i \qquad \forall i, \forall t > T - L_i \quad (3.3)$$

$$c_{k,t}^{x} + c_{k,t}^{w} \le C_{k,t} \qquad \forall k, \forall t \quad (3.4)$$

The objective function (3.1) minimises total costs including priority-dependent WIP cost, penalty cost of delaying WIP, FWS holding cost, and backlog cost. Both releases and initial WIP incur WIP costs. Equation (3.2) ensures mass conservation at the FWS. At the end of any time period, the sum of arrivals at the FWS, initial FWS, and new backlog must equal

the sum of demand (i.e., the requirement resulting from master production scheduling), new FWS, and initial backlog. Yield is not considered but could be easily included (see Leachman, 2002). The mean demand $\bar{D}_i$ in (3.2) and (3.3) aims at preventing any kind of end-of-horizon effect (cp. Leachman, 2002). Capacity constraint (3.4) limits the machine hours required by both new releases and initial WIP to the available machine hours.

$$WIP_{i,j} = \sum_{t=1}^{T_{\mathrm{WIP}}} \sum_{p=1}^{P} w_{i,j,t,p} \qquad \forall i, \forall j \in F_i \tag{3.5}$$

$WIP_{i,j}$ models the initial number of lots in segment $[j-1, j)$ of the process flow of product i at the beginning of period $t = 1$. Equation (3.5) allocates $WIP_{i,j}$ to $w_{i,j,t,p}$ over P priority classes and T_{WIP} time periods. Variable $w_{i,j,t,p}$ defines the fraction of initial WIP that is planned to continue moving at the beginning of t with priority p. Note that if $w_{i,j,t,p} > 0$ for some $t > 1$, wafer lots are planned to be delayed in segment j for $t-1$ periods. This can be necessary in case the projected capacity consumption at downstream operations exceeds the available capacity. If we do not allow to delay lots in a segment, the CT-LP formulation can become infeasible. In order to prevent that planned WIP evolve into a (WIP-oriented) means of lot prioritisation, $w_{i,j,t,p} > 0$ is penalised in the objective function for $t > 1$.

Equations (3.6) and (3.7) model the projected arrivals of product i at the FWS in period t that originate from releases and initial WIP, respectively. Equations (3.8) and (3.9) model the workload in machine hours of work centre k in period t projected from releases and initial WIP, respectively.

$$o^x_{i,t} = \sum_{\tau=0}^{min(t-1, L_{\max})} \sum_{p=1}^{P} E_{i,J_i+1,\tau,p} x_{i,t-\tau,p} \qquad \forall i, \forall t \tag{3.6}$$

$$o^w_{i,t} = \sum_{j=1}^{J_i+1} \sum_{\tau=1}^{min(t, T_{\mathrm{WIP}})} \sum_{p=1}^{P} E_{i,j,J_i+1,t-\tau,p} w_{i,j,\tau,p} \qquad \forall i, \forall t \tag{3.7}$$

$$c^x_{k,t} = \sum_{i=1}^{I} \sum_{j'=1}^{J_i} \sum_{\tau=0}^{min(t-1, L_{\max})} \sum_{p=1}^{P} A_{k,i,j'} E_{i,j',\tau,p} x_{i,t-\tau,p} \qquad \forall k, \forall t \tag{3.8}$$

$$c^w_{k,t} = \sum_{i=1}^{I} \sum_{j=1}^{J_i} \sum_{j'=j}^{J_i} \sum_{\tau=1}^{min(t, T_{\mathrm{WIP}})} \sum_{p=1}^{P} A_{k,i,j'} E_{i,j,j',t-\tau,p} w_{i,j,\tau,p} \qquad \forall k, \forall t \tag{3.9}$$

The higher the fraction of prioritised lots on the shop floor, the lower is the effect of prioritisation on cycle time. Equation (3.10) hence limits the number of lots of product i with priorities $1, 2, \ldots, p$ that are expected to finish in t to the fraction UB_p of the demand of product i in period t. There is no constraint on the number of lots that have the lowest priority P.

$$\sum_{\tau=0}^{min(t-1,L_{\max})} \sum_{p'=1}^{p} E_{i,J_i+1,\tau,p'} x_{i,t-\tau,p'} + \sum_{j=1}^{J_i+1} \sum_{\tau=1}^{min(t,T_{\mathrm{WIP}})} \sum_{p'=1}^{p} E_{i,j,J_i+1,t-\tau,p'} w_{i,j,\tau,p'} \leq UB_p D_{i,t}$$
$$\forall i, \forall t, \forall p \in \{1, 2, \ldots, P-1\} \quad (3.10)$$

3.2.2 Integration with scheduling level

Wafer fabrication is characterized by recirculating process flows sharing the same equipment. The queue of a work centre can therefore hold lots from different process flows and different lots of the same process flow can wait for different operations. Least slack first scheduling dispatches the lot with the earliest ODD first. The priority ranking of lots that are waiting in the same queue is thus defined by their ODD for the operation that is pending at the respective work centre. The idea is to translate a higher priority into a tighter ODD so that the corresponding lot moves up in the ranking.

The priority of WIP lots, i.e., the set of ODDs assigned to each lot, is aligned with the plan at the beginning of every time period $(t-1, t]$. The target number of lots of product i in segment $[j-1, j)$ with priority p is defined by $\lceil \sum_{p'=1}^{p} w_{i,j,t,p'} \rceil - \lceil \sum_{p'=1}^{p-1} w_{i,j,t,p'} \rceil$. On the shop floor, priority p is reassigned first to lots of the respective product type in the respective segment that did have the same priority p in the previous time period. If there are more lots to prioritise, lots of the next lower priority class are selected. Here, the lots with the earliest ODD are selected first. The $x_{i,t,p}$ lots of product i that have to be released in period t with priority $p \in \{1, 2, \ldots, P\}$ are placed in descending order of priority in the release sequence.

Every lot has an ODD for its next operation on a bottleneck work centre. Assuming that l is a lot of product i with priority p and release date RD_l, then its ODD for operation j, i.e. its due date for the initiation of operation j, is

$$ODD_{l,i,j,p} = RD_l + \sum_{s=1}^{j} TCT_{i,s,p}, \quad (3.11)$$

where $TCT_{i,j,p}$ is the target cycle time of a lot of product i with priority p between the initiation of operation $j-1$ and the initiation of operation j. The target cycle times of all segments are assumed to be known for all priorities.

In case the mid-term production plan requires a change of the priority of lot l from p' to p, the release date must be revised before a new set of ODDs can be calculated. Given that lot l is located in segment $[j-1, j)$ at the beginning of time period t, the revised release date is

$$RD_l = t - \sum_{s=1}^{j-1} TCT_{i,s,p} - \frac{1}{2} TCT_{i,j,p}. \quad (3.12)$$

The calculation of RD_l in (3.12) implements the assumption that the lot did have priority p since its release and that it has halfway passed segment $[j-1, j)$. The set of target cycle times $TCT_{i,j,p'}$ is then used to calculate the new set of ODDs with equation (3.11). In case the mid-term production plan requires no change of priority, lots keep their old set of ODDs.

3.3 WIP-oriented mid-term production planning

To show how CT-oriented mid-term production planning differs from WIP-oriented mid-term production planning, we define a benchmark LP formulation, denoted as WIP-LP, in Section 3.3.1. The WIP-LP is based on the WIP-oriented models proposed by Hwang and Chang (2003), Bard et al. (2010), and Cai et al. (2011). The objective of the WIP-LP is the same as the objective of the CT-LP, i.e., minimizing both the deviation of fab output from output requirement and the amount of WIP. Despite this similarity, the WIP-LP effectuates the objective differently. It provides targets for the throughput and the buffer level of every bottleneck operation. These targets are fulfilled by the scheduling level (see Section 3.3.2). The key difference to CT-oriented mid-term production planning lies in the concept of flow control. While the CT-LP accelerates or decelerates a lot primarily by changing its target cycle times, the WIP-LP delays a lot in a buffer or makes it move by changing the targets of throughput and buffer level. In WIP-oriented planning, cycle times are not planned directly but are a result of the management of throughput and buffer level. The assumptions regarding cycle time and backlog are the same as for the CT-LP.

3.3.1 LP formulation

The notation introduced in Section 3.2.1 is mostly reused. Additional parameters are:

$E_{i,j,\tau}$ Probability that a lot of product i arrives at the queue of bottleneck operation $j \in F_i$ in the τth period after the period in which it initiates bottleneck operation $j-1$ with $\tau \in \{0, 1, \ldots, L_{\max}\}$. $E_{i,1,\tau}$ refers to the flow from the release to the arrival at the queue of the first operation. $E_{i,J_i+1,\tau}$ refers to the flow from the initiation of the last operation to the arrival at the FWS. Note that $\sum_{\tau=0}^{L_{\max}} E_{i,j,\tau} = 1$.

$E'_{i,j,\tau}$ Probability that a lot of product i initially in segment $[j-1, j)$ arrives at the queue of bottleneck operation j in period τ with $j \in F_i$ and $\tau \in \{1, 2, \ldots, T + L_{\max}\}$. Unlike as in the CT-LP, the modelled segment $[j-1, j)$ does not include the downstream queue (cp. Figure 3.2). Note that $\sum_{\tau=1}^{T+L_{\max}} E'_{i,j,\tau} = 1$.

WIP_{ij} Number of lots of product i that are located in segment $[j-1, j)$ (excluding the queue of bottleneck operation j) at the beginning of $t = 1$, with $j \in F_i$. Hence, WIP_{ij} represents the initial WIP in segment $[j-1, j)$ that is not waiting for operation j.

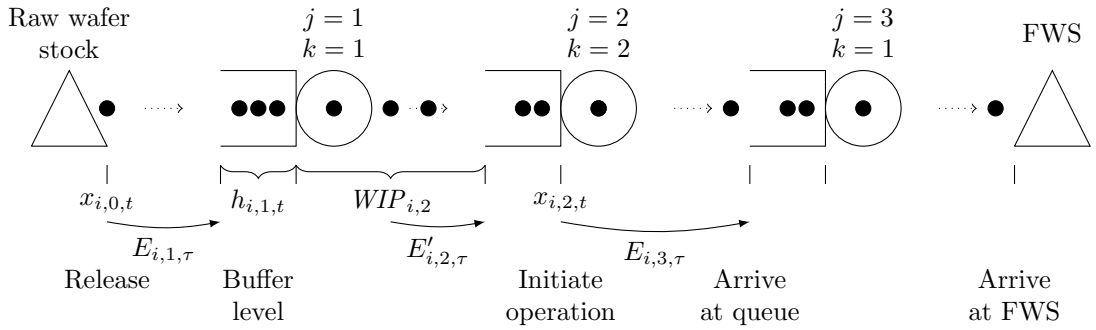


Figure 3.2: Modelling of WIP and cycle time in the WIP-LP formulation.

V^{W} Cost per period and per lot of WIP.

Additional variables are:

$x_{i,j,t}$ Number of lots of product i that initiate bottleneck operation $j \in \{0\} \cup F_i$ in period t. $x_{i,0,t}$ models the number of releases into the fab. $x_{i,J_i+1,t}$ models the number of lots that leave the FWS to meet demand.

$h_{i,j,t}$ Number of lots of product i that wait in queue for bottleneck operation $j \in F_i$ at the end of period t, i.e., the buffer size of j. $h_{i,J_i+1,t}$ models number of lots of product i in the FWS. The initial buffer size must be given as parameter $h_{i,j,0}$.

Figure 3.2 illustrates the modelling of both WIP and cycle time in the WIP-LP. Lots that wait in queue for bottleneck operation j are called the buffer of j. For every operation j of each product i and for every time period t, the WIP-LP provides both $x_{i,j,t}$ and $h_{i,j,t}$. $E_{i,j,\tau}$ defines the probability mass function of the cycle time between the initiation of operation $j-1$ and the arrival at the queue of operation j. $E'_{i,j,\tau}$ defines the probability mass function of the cycle time of initial WIP in segment $[j-1, j)$ until its arrival at the queue of operation j. WIP_{ij}, i.e., the initial WIP of segment $[j-1, j)$, covers all lots in segment $[j-1, j)$ except the buffer of j.

WIP-LP formulation

$$\min \sum_{i=1}^{I} \sum_{j=1}^{J_i} \sum_{t=1}^{T+L_{\max}} V^{\mathrm{W}} h_{i,j,t} + \sum_{i=1}^{I} \sum_{t=1}^{T+L_{\max}} V^{\mathrm{H}} h_{i,J_i+1,t} + \sum_{i=1}^{I} \sum_{t=1}^{T+L_{\max}} V^{\mathrm{B}} b_{i,t} \tag{3.13}$$

$$\sum_{\tau=0}^{min(t-1,L_{\max})} E_{i,j,\tau} x_{i,j-1,t-\tau} + E'_{i,j,t} WIP_{i,j} + h_{i,j,t-1} = x_{i,j,t} + h_{i,j,t} \quad \forall i, \forall j \in F_i, \forall t \tag{3.14}$$

$$x_{i,J_i+1,t} + b_{i,t} = \begin{cases} D_{i,t} + b_{i,t-1} & \forall i, \forall t \leq T \\ \bar{D}_i + b_{i,t-1} & \forall i, \forall t > T \end{cases} \tag{3.15}$$

$$x_{i,0,t} = \bar{D}_i \qquad \forall i, \forall t > T - L_i \tag{3.16}$$

$$\sum_{i=1}^{I} \sum_{j=1}^{J_i} A_{k,i,j} x_{i,j,t} \leq C_{k,t} \qquad \forall k, \forall t \tag{3.17}$$

The objective function (3.13) minimises WIP costs, FWS holding costs, and backlog costs. Note that WIP costs incur per lot and per period at every buffer where the lot is planned to be delayed for one or more periods. The equations (3.14) and (3.15) ensure mass conservation at each WIP buffer and at the FWS. Equation (3.17) is the capacity constraint. Equation (3.16) in combination with the case $t > T$ of equation (3.15) prevents end-of-horizon effects.

3.3.2 Integration with scheduling level

In WIP-oriented planning, lots do not directly receive priorities. However, a high throughput target $x_{i,j,t}$ prioritises lots of product i that wait for operation j in the sense that in general more lots of product type i will be processed than of other types also waiting in the same queue. This is achieved with the smallest production achievement ratio (PAR) first scheduling policy.

Let $x'_{i,j}$ be the counter of lots of product i that have initiated operation j since the beginning of the current time period $(t-1, t]$. Further, let $IJ(k)$ be the set of all product-operation combinations (i, j) that require processing from work centre k. The smallest PAR first scheduling policy dispatches that lot from the queue of work centre k first that minimises the PAR $x'_{i,j}/x_{i,j,t}$ over all $(i, j) \in IJ(k)$. Smallest PAR first scheduling balances the relative backlog across the different types of jobs at a work centre. Dispatching lots of product i for operation j (with $(i, j) \in IJ(k)$) stops at work centre k if $x'_{i,j}/x_{i,j,1} \geq 1$. Without this stopping rule, the allocation of capacity and hence the WIP-oriented mid-term production plan would be ineffective and meaningless. Note that in case yield is 100 %, as it is assumed in this study, the fulfilment of all throughput targets $x_{i,j,1}$ implies that all buffer level targets $h_{i,j,1}$ are fulfilled as well. Therefore, we do not use the buffer level targets on the scheduling level.

3.4 Rolling-horizon framework for performance evaluation

We embed the LP formulations into a rolling horizon framework, which is illustrated in Figure 3.3. The framework allows us to evaluate the different mid-term production planning approaches in a dynamic environment under identical conditions. The framework includes a wafer fab configuration, which is based on the widely used MIMAC data set 1 (Fowler and

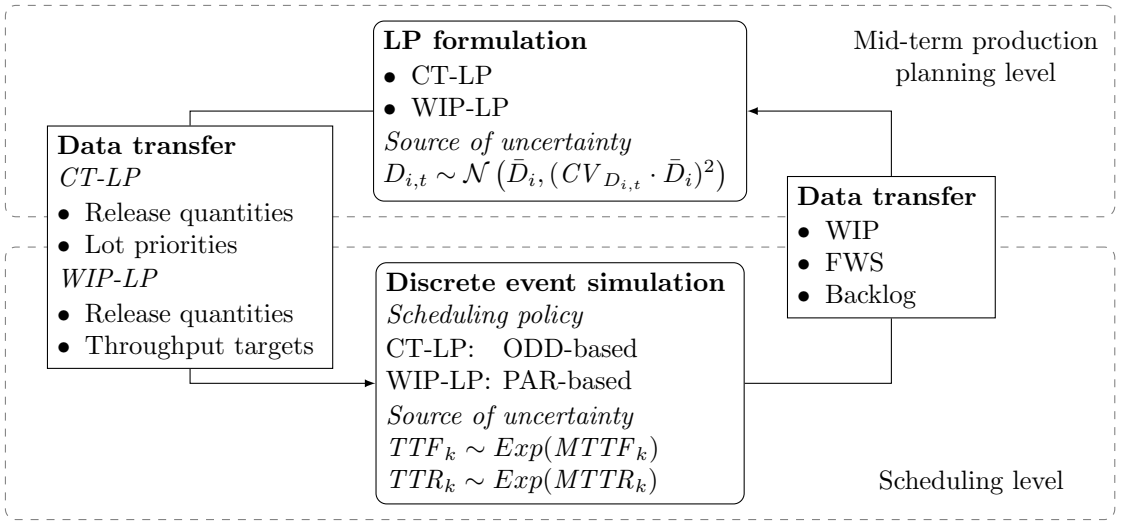


Figure 3.3: Rolling horizon framework.

Robinson, 1995). On the mid-term production planning level, one of the LP formulations solves the production planning problem for the wafer fab at the beginning of the week. The first week of the plan is deployed in a discrete event simulation model of the fab. Either release quantities and lot priorities (CT-LP) or release quantities and throughput targets (WIP-LP) are forwarded to the scheduling level. A scheduling policy, which depends on whether the CT-LP or the WIP-LP is used, implements the plan. At the end of the week, the current WIP, FWS, and backlog are reported from the scheduling level back to the planning level in order to initialize a revised mid-term production planning problem. Then the cycle repeats. The framework includes two sources of uncertainty: the output requirement of the fab and machine availability.

3.4.1 Reference case

The MIMAC data set 1 specifies a fab that fabricates memory chips on two process flows (see Table 3.1). The fab includes a total of 83 work centres, of which each is composed of 1 to 18 identical machines, running in parallel. Setup time, load time, processing time, unload time, and wafer travel time are modelled by static values. 16 work centres operate in batch mode, for which batch size limits and feasible batch compositions are given. Two work centres have sequence dependent setup times. The average machine availability is 91.0 %. Time to failure and time to repair are exponentially distributed and the means ($MTTF_k$ and $MTTR_k$) are given for each work centre k. We scale all $MTTF_k$ and $MTTR_k$ with 0.5 based on the assumption that high-impact low-probability machine failures, such as those with $MTTF_k > 1$ week, lead to manual adjustments of $C_{k,t}$ in a failure event and should not reduce the effective capacity upfront. As a result of the scaling, the number of work centres with $MTTF_k > 1$ week decreases from 16 to two.

Table 3.1: Two process flows specified by the MIMAC data set 1.

	Process flow 1	Process flow 2
Product mix	2/3	1/3
Number of operations	210	245
Number of mask layers	14	16
Raw processing time [week]	1.75	2.01

The discrete event simulation model of the fab uses the Java library SSJ 2.5 (L'Ecuyer et al., 2002). Lots of 48 wafers are released with inter-arrival times that are constant within each time period. A cyclic release sequence ensures that both process flows are evenly loaded. Each work centre has a single queue. Queues of bottleneck work centres are sorted according to the scheduling policy. All other queues are ordered in a FIFO manner. Batch operations run in a greedy 'load and go' fashion, i.e., processing starts as soon as the queue holds enough lots to form a feasible batch. Operations with sequence dependent setup times follow a setup avoidance rule. In the event of a machine failure, items in process are finished before the machine goes off-line. Neither rework nor operators are considered and yield is 100 %. The maximum steady-state throughput rate with machine failures is 78.82 lots per week and without machine failures 83.15 lots per week.

3.4.2 Implementation of LPs

CT-LP and WIP-LP are implemented in Java using the IBM ILOG Concert Technology. Cost parameters are initialized such that $V^{\mathrm{B}} = 10$, $V^{\mathrm{D}} = 50$, $V^{\mathrm{H}} = 2$, $V^{\mathrm{W}} = 4$, $V_1^{\mathrm{W}} = 5$, and $V_2^{\mathrm{W}} = 4$, which is motivated by settings in related studies (cp. Cai et al., 2011; Irdem et al., 2010; Kim and Leachman, 1994). Two priority classes are considered, i.e., $P = \{1, 2\}$ with $p = 1$ and $p = 2$ representing hot lots and regular lots, respectively. Following the recommendation of Fronckowiak et al. (1996), the upper bound for hot lots UB_1 is 20 %.

Empirical parameters are initialized based on observations in pre-simulation runs. This includes capacity consumption factors and probability mass functions of segment cycle times. Priority-dependent segment cycle times are estimated based on pre-simulation runs with priority-FIFO dispatching and a share of 20 % prioritised lots. All pre-simulation runs have the same fab utilization and machine failure rate configuration as the experimental runs, which they precede.

The set of bottleneck work centres includes the $K = 5$ most utilised work centres of the fab. Analyses show that these work centres serve about 30 % of all operations, i.e., $J_1 = 61$ and $J_2 = 75$. This is consistent with the idea that mid-term production planning is conducted at an intermediate level of detail between master production planning and scheduling.

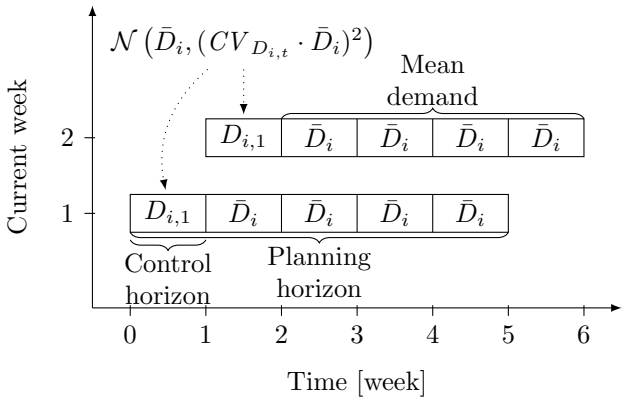


Figure 3.4: Demand uncertainty in the rolling horizon framework.

Mid-term production plans are calculated for a planning horizon 20 weeks, which is approximately five times the mean cycle time at 95.16 % fab utilization with machine failures. The first week of the planning horizon, i.e., the control horizon, is deployed on the shop floor. While plans are based on mean machine availability, the actual machine availability is determined by exponentially distributed random variables in the course of simulation. Likewise the actual demand, i.e., the master production schedule requirement, is assumed to be uncertain. Every time the planning horizon 'rolls' one week ahead, the demand $D_{i,1}$, which has to be met during the control horizon, is drawn from a normal distribution. This is illustrated in Figure 3.4 for a one-week control horizon that is part of a five-week planning horizon. For each product i, the normal distribution is truncated to $0 \leq D_{i,1} \leq 2 \cdot \bar{D}_i$ and has the standard deviation $CV_{D_{i,t}} \cdot \bar{D}_i$ with $CV_{D_{i,t}} = \sigma / \bar{D}_i$ being the coefficient of variation. Beyond the control horizon, planning assumes a static demand of $\bar{D}_i$ for product i, which is the mean of the normal distribution.

The two LPs must use different time granularities such that actual segment cycle times are longer than a modelled time period. If a segment cycle time is shorter than a modelled time period, a lot could be at the start and the end of a segment at the same point in time because it arrives in the same time period as it starts. The CT-LP uses a granularity of weeks, i.e., $T = 20$. This is sufficient to approximate the total cycle time to downstream bottleneck operations, which is generally in the order of weeks. In accordance with Hwang and Chang (2003) and Bard et al. (2010), the WIP-LP uses a granularity of days ($T = 140$). This is necessary because the WIP-LP models the total cycle time as the sum of segment cycle times, which are mostly shorter than a week. In order to translate the weekly master production schedule requirement into daily $D_{i,t}$ for the WIP-LP, the requirement is evenly distributed. The PAR scheduling policy is adapted such that it aims at fulfilling the daily targets cumulated over the days of a week.

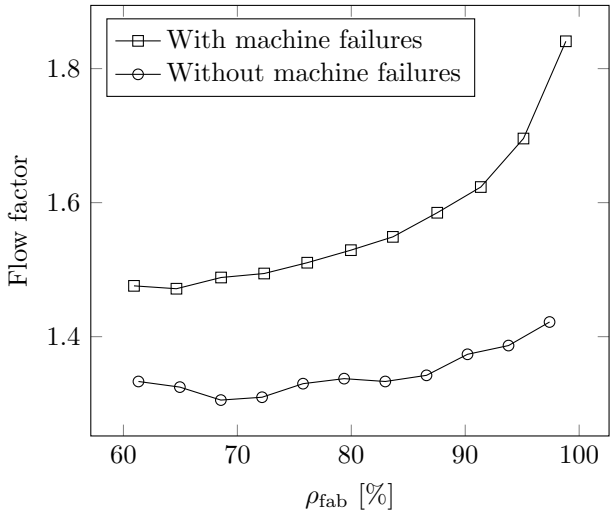


Figure 3.5: Operating curve of the MIMAC fab 1.

The number of variables of the WIP-LP is approximately seven times the number of variables of the CT-LP because of the difference in time granularity. Assuming that $L_{max} \leq T$, the number of variables of CT-LP and WIP-LP are bounded by $2IT(JP + P + 2)$ and $2IT(2J + 1)$, respectively. The number of constraints of CT-LP and WIP-LP are bounded by $2T(IP + 2I + K) + IJ$ and $2T(IJ + 2I + K)$. Both time granularity and segment-specific mass conservation constraints drive the high number of constraints in the WIP-LP.

3.4.3 Verification

Figure 3.5 shows two operating curves based on data that are collected in simulation runs with FIFO dispatching as scheduling policy. The flow factor is the mean observed cycle time divided by the mean total raw processing time of the process flows. The raw processing time includes load time, processing time, and unload time. The shapes of the operating curves meet the expectations about the effect of manufacturing variability and fab utilization on cycle time.

We verify the coordination between LP formulation and simulation model by comparing the number of lots that are planned to arrive at the FWS per week with the number of lots that actually arrive at the FWS in the simulation. In a configuration without machine failures and a static total demand of 78 lots per week ($\rho_{\text{fab}} = 93.81\,\%$), planning with the CT-LP achieves a mean absolute percentage deviation of observed output from planned output of 5.81 %. The WIP-LP reaches under the same conditions 5.0 %. This verifies that both LP formulations are capable of performing equally well in a scenario with no uncertainty.

3.5 Numerical experiments

3.5.1 Design of experiments

An experiment is conducted to study the effect of the mid-term production planning and scheduling (MTPPS) methods that are specified in Table 3.2 on fab performance. We compare four different methods. CT-oriented planning stands for CT-oriented mid-term production planning with two priority classes using the CT-LP. Release planning is CT-oriented planning but with only one priority class. It plans the release quantities of regular lots subject to demand but the ODDs of released lots cannot be modified. WIP-oriented planning stands for WIP-oriented mid-term production planning using the WIP-LP. The constant release rate method releases $\bar{D}_i$ lots per week regardless of the actual demand.

In addition to the MTPPS method, two other factors, which are likely to have an effect on the performance of MTPPS, are considered. These are supply uncertainty (with and without machine failures) and demand uncertainty ($CV_{D_{i,t}} = 0.0, 0.1, 0.2, 0.3$). Note that if $CV_{D_{i,t}} = 0.0$, the demand is constant throughout the planning horizon and has the value $\bar{D}_i$. The responses that we are going to analyse at first are number of lots in system $WIP + FWS$, i.e., the sum of work in process and finished wafer stock, and γ-service level. The γ-service level of a week t is defined as

$$\gamma_t = \max\left\{1 - \frac{b_t^{\text{obs}}}{D_t}, 0\right\} \tag{3.18}$$

with observed demand D_t and observed backlog b_t^{obs}.

The experimental design is a 4^2-×-2 full factorial design. Five replications are performed for each treatment combination, resulting in 160 independent simulation runs. A single simulation run simulates wafer fabrication for 300 weeks including a 70-week warm-up period. The steady-state fab utilization is with machine failures 95.16 % and without machine failures 93.81 %. Proper variation and coordination of random number streams are implemented.

Table 3.2: Mid-term production planning and scheduling (MTPPS) methods.

Method	Planning model	Scheduling policy	Priority classes
CT-oriented planning	CT-LP	ODD	2
Release planning	CT-LP	ODD	1
WIP-oriented planning	WIP-LP	PAR	n/a
Constant release rate	Release of $\bar{D}_i$ per week	FIFO	n/a

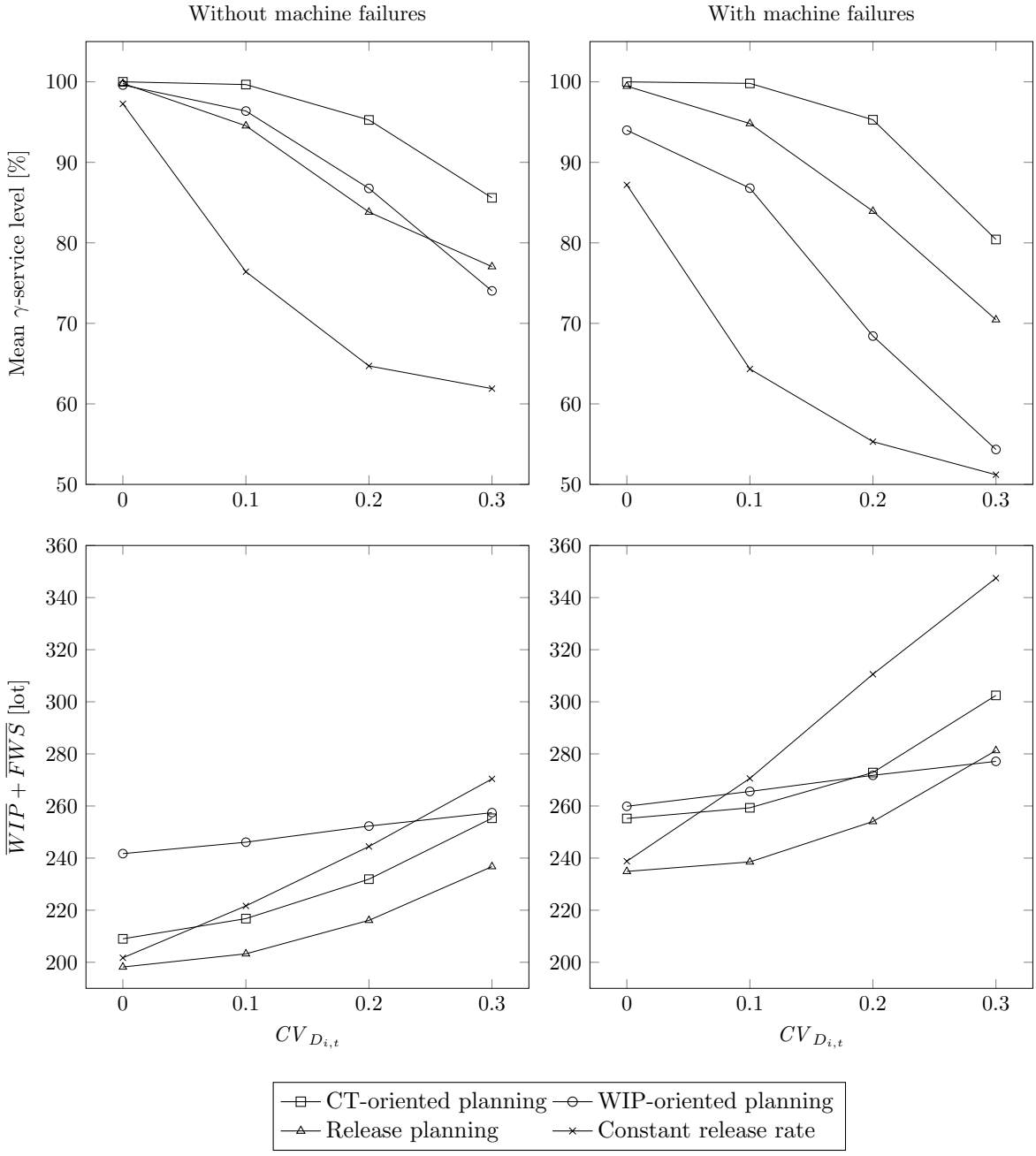


Figure 3.6: Plots of γ-service level and $\overline{WIP} + \overline{FWS}$ showing MTPPS method $\times$ demand uncertainty interaction for both levels of machine failures.

3.5.2 Results

Service level and number of lots in system

An analysis of variance (ANOVA) is conducted for service level and number of lots in system (see the Tables B.1 and B.2 in the appendix). For each ANOVA, we have verified that the assumptions of error term normality and equality of variance are not violated. The results are that all main effects and all two-way interaction effects are significant at the 95 % confidence level.

Specific focus of the experimentation lies on differences in the effectiveness of the MTPPS method. The conclusion of a significant interaction between MTPPS method and the two types of uncertainty implies that differences in MTPPS methods must be examined separately for the different levels of uncertainty. This is shown in the interaction plots in Figure 3.6.

CT-oriented planning outperforms WIP-oriented planning. At all levels of demand uncertainty and supply uncertainty, CT-oriented planning results in the highest service level. At the same time, it always leads to fewer lots in system than WIP-oriented planning, except for the one configuration with machine failures and $CV_{D_{i,t}} = 0.3$.

In configurations with machine failures, which is the more realistic factor level, CT-oriented planning provides on average, i.e., across the four demand uncertainty levels, a 17.99 percentage points higher service level than WIP-oriented planning, while the number of lots in system is on average 1.44 % higher. Without machine failures, CT-oriented planning produces on average a 5.94 percentage points higher service level than WIP-oriented planning, whereas the number of lots in system is on average 8.48 % lower.

A comparison of CT-oriented planning with release planning reveals the positive effect of the CT-LP performing lot prioritisation on service level while all other factors remain unchanged. CT-oriented planning achieves on average a 6.71 percentage points higher service level with 8.04 % more lots in system than release planning in configurations with machine failures. The numbers are similar without machine failures.

The constant release rate method shows the effect of undertaking no mid-term production planning in reaction to changes of machine availability and demand. With increasing uncertainty, the service level declines and the number of lots in system increases. As Figure 3.7 shows, the increase in number of lots in system is due to an increasing FWS caused by a mismatch between output and demand. On average, CT-oriented planning increases the service level compared to constant release rate by 29.37 percentage points and 20.06 percentage points in configurations with and without machine failures, respectively.

Tukey's honestly significant difference test is used to verify that the differences between all pairs of MTPPS methods at each treatment combination of demand uncertainty and supply uncertainty are significant (at an overall 95 % confidence level). Indeed, the service level of CT-oriented planning differs significantly from the service level of other MTPPS methods

except from release planning in two out of seven treatment combinations. $\overline{WIP} + \overline{FWS}$ of CT-oriented planning differs significantly from $\overline{WIP} + \overline{FWS}$ of other MTPPS methods except from WIP-oriented planning in one out of seven treatment combinations. The complete test results are presented in the Tables B.3 to B.6 in the appendix.

Work in process and finished wafer stock

We expand the analysis by disaggregating the number of lots in system into its summands WIP and FWS. The treatment means of WIP and FWS are presented in Figure 3.7. Since we collect statistics after the simulation has reached a steady state, Little's law can be applied. Hence, the results on WIP and FWS can also be used to draw conclusions on cycle time and FWS turns, respectively.

Compared to the WIP level of other MTPPS methods, the WIP level of WIP-oriented planning is elevated. This is consistent with experiences reported by Infineon fab operators. One reason for this phenomenon is a fundamental shortcoming of WIP-oriented planning. It relies on PAR-based scheduling, which will only dispatch until the targets are met and, consequently, will limit throughput to the targets. This increases the sojourn time of lots in buffers, WIP level, and eventually cycle time. CT-oriented planning and release planning, in contrast, rely on ODD-based scheduling, which is not dependent on throughput targets and will dispatch lots as long as queues are not empty. Compared to WIP-oriented planning, CT-oriented planning generally leads to lower WIP levels. This also means that CT-oriented planning achieves shorter cycle times. The WIP levels of CT-oriented planning increase significantly, however, with demand uncertainty.

Looking at averages over demand uncertainty, CT-oriented planning results in 7.37 % fewer lots in WIP and 575.38 % more lots in FWS than WIP-oriented planning in the case with machine failures. The superiority of CT-oriented planning is confirmed by the case without machine failures. Here, CT-oriented planning shows on average 13.70 % fewer lots in WIP and 148.10 % more lots in FWS than WIP-oriented planning.

Compared to release planning, CT-oriented planning holds on average 2.85 % and 2.26 % (with and without machine failures) more lots in WIP. Likewise, we see 96.77 % and 101.96 % more lots in the FWS. This shows that prioritising lots effectively moves lots out of the process flows into the FWS and, in doing so, improves service levels. However, the improvement of the service level through lot prioritisation comes at the expense of a marginal increase of variability in the production system and therefore of WIP and cycle time.

Comparing the different methods at the scheduling level, we find that at low levels of uncertainty, the ODD-based methods (CT-oriented planning, release planning) achieve shorter cycle times than the FIFO-based constant release rate. This matches the findings of Lu et al. (1994). At higher levels of uncertainty, planning activities add variability to the production system and the mean cycle time of the constant release rate method is quickly passed.

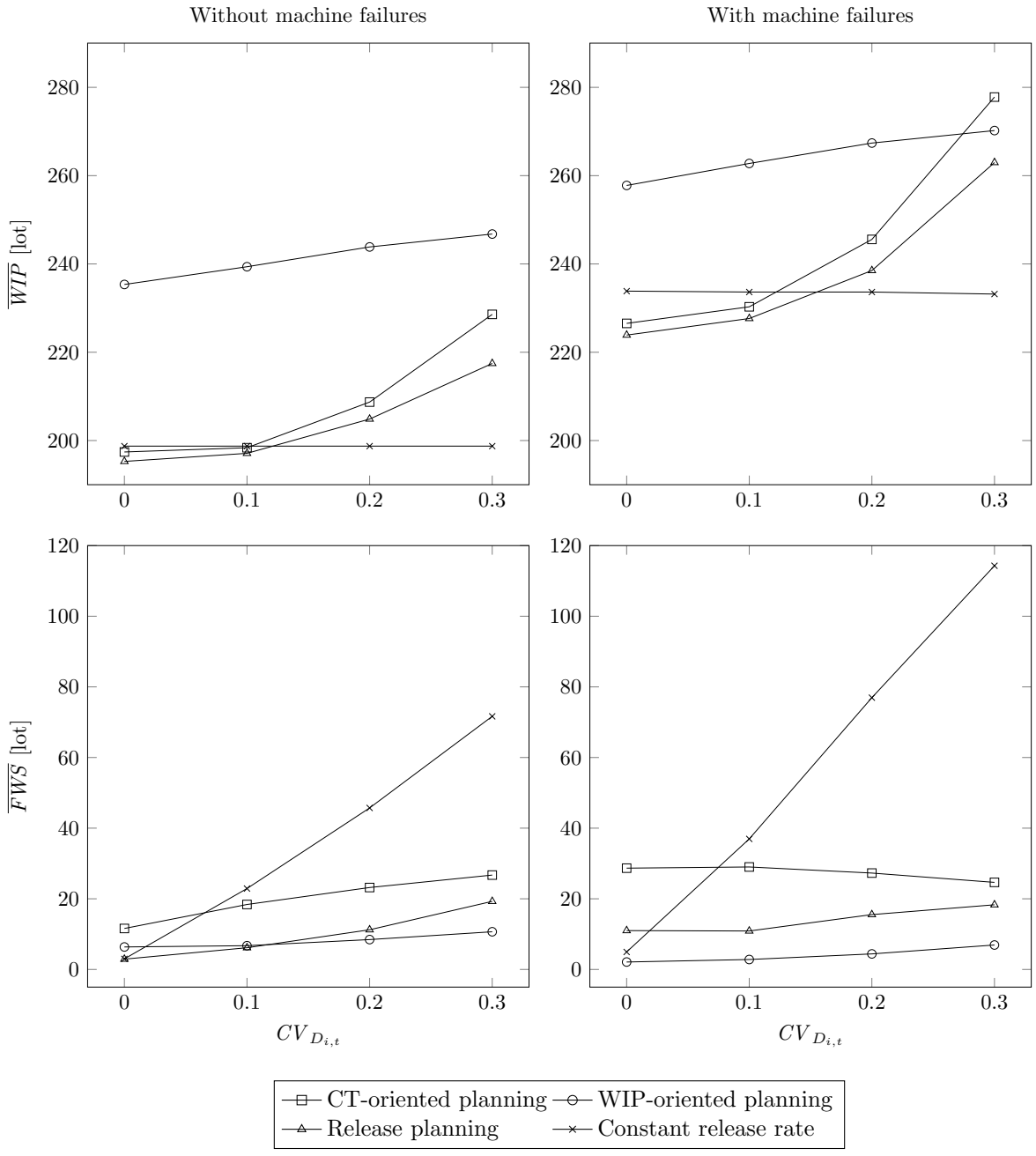


Figure 3.7: Plots of $\overline{WIP}$ and $\overline{WIP}$ showing MTPPS method × demand uncertainty interaction for both levels of machine failures.

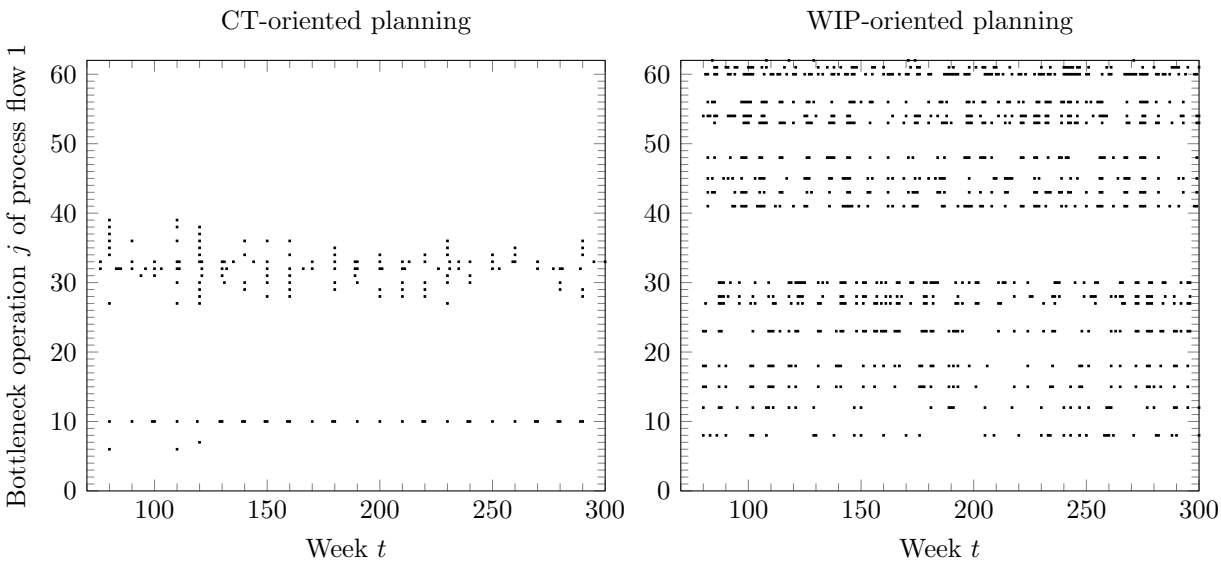


Figure 3.8: Lot prioritisation of CT-oriented planning (left) and lot buffering of WIP-oriented planning (right) in reaction to cyclic demand increases.

Comprehensibility and the size of mid-term production plans

Both mid-term production planning models, CT-oriented and WIP-oriented, are designed to make adjustments on the scheduling level in reaction to changes of the system state and demand. In this section, we compare the scope of these adjustments, i.e., the scope of priority adjustments and the scope of buffer-level target adjustments.

In a separate series of simulation runs without machine failures, the fab serves a static base demand that leads to a fab utilization of 93.81 % for eight weeks. During the following two weeks, demand increases to a level that is equivalent to a fab utilization of 104.63 %. This 10-week cycle repeats for 300 weeks. The two-week peak demand appears only in the control horizon. Beyond that, mid-term production planning assumes the base demand. In order to satisfy the peak demand on time, mid-term production planning has to prioritise lots because there is no opportunity to increase releases in a timely manner.

The outcome of this experiment is illustrated in Figure 3.8. In case CT-oriented planning increases the priority of at least one lot in segment $(j-1, j]$ in week t, then the left-hand side plot of Figure 3.8 shows a dot at these coordinates. In case WIP-oriented planning plans the buffer level of bottleneck operation j at the end of week t above a critical value, then the right-hand side plot of Figure 3.8 shows a dot at these coordinates. The critical value is the mean buffer level of j at 93.81 % fab utilization.

Lot prioritisation of the CT-LP creates a pattern in Figure 3.8 (left) that is easy to comprehend. Every ten weeks, some lots in segments $27 \leq j \leq 39$ are prioritised because they are likely to finish one week earlier than with regular priority. These lots are prioritised to fulfil the peak demand on time. Since prioritising lots in segments $27 \leq j \leq 39$ will create a backlog of scheduled completions at a later point in time, lots in segment $j = 10$ are prioritised to balance that. Buffer level changes of the WIP-LP in Figure 3.8 (right) do not follow such a clear pattern.

The CT-LP is much smaller than the WIP-LP. The WIP-LP counts on average 50,258 columns and 26,305 rows, while CT-LP has on average 1,552 columns and 778 rows. There is also a difference in the amount of data that is transferred between planning level and scheduling level. While CT-oriented planning usually generates priority changes for selected segments in selected time periods, WIP-oriented planning has to provide every bottleneck operation in every time period with a target.

3.6 Conclusion

This study addresses the mid-term production planning problem in wafer fabrication. The challenge is to guide large inventories through a failure-prone production system such that a time-varying demand is met and cycle time is short. The planning problem becomes more difficult as the required fab output can change unexpectedly during the production lead time.

In the scientific literature, this problem is commonly solved following the WIP-management paradigm by planning throughput rates and buffer levels of bottleneck operations. Based on our analysis of industry practices, we see the need for a new planning model that solves the problem in a CT-oriented way by planning throughput rates for the point of release and target cycle times up to bottleneck operations.

We propose a new LP formulation for CT-oriented mid-term production planning and compare it against an LP formulation for WIP-oriented mid-term production planning. The LP is embedded in a rolling horizon framework. Production plans are deployed in a discrete event simulation model of a reference wafer fab. Sources of uncertainty are demand changes that result from updated master production schedules and machine failures.

The results show that CT-oriented mid-term production planning in combination with a least slack scheduling policy outperforms WIP-oriented mid-term production planning in combination with a scheduling policy that fulfils throughput targets. In the examined case, CT-oriented planning provides on average a 17.99 percentage points higher service level with 7.37 % shorter cycle time and just 1.44 % more lots in system. The CT-oriented approach is also superior in settings with reduced uncertainty, as our results for a production system without machine failures show.

CT-oriented planning achieves short cycle times at low levels of variability. With increasing demand uncertainty, lot prioritisation, which aims at maintaining the service level, adds to the variability in the production system and therefore increases cycle time. In our experiments, lot prioritisation through CT-oriented mid-term production planning contributes on average 6.71 percentage points to the service level at the expense of a 2.85 % increase of cycle time.

WIP-oriented planning is characterized by prolonged cycle times. Since throughput targets are formulated subject to mean-based capacity assumptions, they are likely to constrain the performance of work centres and cause delays. CT-oriented planning generates more comprehensible, human-readable plans than WIP-oriented planning. We consider this as an important result since acceptance by operators is a prerequisite for effective planning.

Even though our comparison builds on an accepted reference case, future research could test the presented results also for other environments. A research question could be how well CT-oriented mid-term production planning performs when demand, product mix, and the share of prioritised lots cannot be assumed static. This could require a frequent revision of the distributional assumptions about cycle times. The relevance of this question may, however, be limited as master production planning generally ensures a high load in order to guarantee a high utilization of expensive equipment. Another research question could be how the dual price of constraint (3.10), which limits the number of prioritised lots, can be used to determine the optimal number of prioritised lots.

Hierarchical production planning and control has to be designed in a way that it fulfils the manufacturing objectives even if important parameters, such as future capacity and demand, cannot be predicted with certainty. CT-oriented planning meets this requirement. It improves the service level through the tactical prioritisation and deprioritisation of lots. In addition, the negative effect of inappropriate priority changes is limited since opportunistic non-delay scheduling will keep utilization high.

Chapter 4

Generation of low-dimensional capacity constraints for parallel machines

Published as:
Kriett, P. O. and Grunow, M. (2017). Generation of low-dimensional capacity constraints for parallel machines. *IISE Transactions*, 49(12):1189–1205

Abstract

A crucial input to production planning is a capacity model that accurately describes the amount of work that parallel machines can complete per planning period. This paper proposes a procedure that generates the irredundant set of low-dimensional, linear capacity constraints for unrelated parallel machines. Low-dimensional means that the constraints contain one decision variable per product type, modeling the total production quantity across all machines. The constraint generation procedure includes the Minkowski addition and the facet enumeration of convex polytopes. We discuss state-of-the-art algorithms and demonstrate their effectiveness in experiments with data from semiconductor manufacturing. Since the computational complexity of the procedure is critical, we show how uniformity among machines and products can be

used to reduce problem size. Further, we propose a heuristic based on graph partitioning that trades constraint accuracy off against computation time. A full factorial experiment with randomly generated problem instances shows that the heuristic provides more accurate capacity constraints than alternative low-dimensional capacity models.

4.1 Introduction

Production planning defines time-phased production quantities based on capacity, inventory, and demand. These production targets form the master production schedule and are typically determined for a wide range of product types, multiple production sites, and a planning horizon of several weeks. Production control translates these targets into scheduling and dispatching decisions (see, e.g., Hopp and Spearman, 2011; Vollmann et al., 1997). Production planning requires a representation of capacities at an aggregate level. Such aggregation must be done accurately. If capacity is overestimated, the targets can be infeasible, which increases cycle times and late deliveries. If capacity is underestimated, the production system can become under-utilized, which increases unit costs. Especially in capital-intensive industries, such as semiconductor manufacturing, accurate capacity modeling is crucial for the quality of production plans and eventually for the success of a business (see, e.g., Mönch et al., 2013; Uzsoy et al., 1992, 1994).

Production is usually planned subject to the capacity of bottleneck stages. A bottleneck stage often consists of unrelated parallel machines of different types (UPMs), i.e., there is no relationship among the processing times of the same job on different machine types. Machines of different age, with different capabilities, and from different manufacturers run in parallel. A good example for the prevalence of UPMs in manufacturing and therefore in production planning is the semiconductor industry. Bottleneck operations in wafer fabrication are performed by photolithography workstations, which have been modeled as UPMs in, e.g., Lee et al. (2002) and Chung et al. (2008). Bottlenecks of UPMs also exist in wafer testing (Centeno and Armacost, 2004), assembly, and final testing (Song et al., 2007). Examples of UPM bottlenecks in other industries are drilling operations in the manufacturing of printed wiring boards (Yu et al., 2002) and dicing operations in the fabrication of compound semiconductors (Kim et al., 2002). Even operations on space stations require the modeling of resources as UPMs (Logendran and Subur, 2004).

The common way of performing production planning is to use variables that model the total production quantity of every product type and, in addition, to also use allocation variables that model the distribution of these quantities among the different UPMs. However, production planning is primarily focused on obtaining production targets that are feasible with regard to the available capacities. Detailed capacity allocations are beyond the scope of production planning and are not part of the production targets determined on this planning level. Including them increases the complexity of both the planning model and the planning process, which includes data retrieval, plan generation, and plan review. For these reasons,

practitioners prefer to exclude detailed capacity allocation decisions from the production planning level. Hence, it is necessary to generate capacity constraints that accurately model the capacity of UPMs without the use of allocation variables. Not using allocation variables reduces the dimensionality of the planning problem to the number of products. We denote the resulting capacity constraints therefore as low-dimensional. Such low-dimensional capacity constraints are accurate if they define the same set of production plans as feasible as the higher-dimensional capacity constraints that are based on allocation variables.

The scientific contributions of this paper include:

- a two-step procedure, which generates accurate, low-dimensional capacity constraints for unrelated parallel machines by
  - first, exploiting partial uniformity to reduce the size of the constraint generation problem,
  - second, decomposing the problem so that standard solution procedures from computational geometry can be applied;
- a partition-based heuristic that divides a large constraint generation problem into smaller ones, trading accuracy off against computation time;
- numerical results based on problem instances from semiconductor manufacturing that show the effectiveness of the proposed procedures; and
- a factorial experiment with randomly generated problem instances that evaluates the accuracy of the constraints provided by the heuristic in comparison to alternative capacity models.

In the following Section 4.2, we define the problem formally. The related literature is reviewed in Section 4.3. In Section 4.4, we develop a two-step procedure that generates accurate, low-dimensional capacity constraints for UPMs. We discuss the critical computational complexity of this procedure in Section 4.4.4. In Section 4.5, we then propose a partition-based constraint generation heuristic. We conduct experiments with field data from the semiconductor industry and with randomly generated problem instances in order to evaluate the proposed methods in Section 4.6. We summarize and draw conclusions in Section 4.7.

4.2 Problem statement

For the sake of brevity, we refer to machine types as machines and to product types as products. Let $I = \{1, 2, \ldots, m\}$ be a set of m parallel machines and $J = \{1, 2, \ldots, n\}$ a set of n products. We assume that every product $j \in J$ requires a single operation. This operation can be an aggregate representation of several real operations that are performed by the same machine on a re-circulating process flow. The eligible machine set $I(j) \subseteq I$ defines the subset

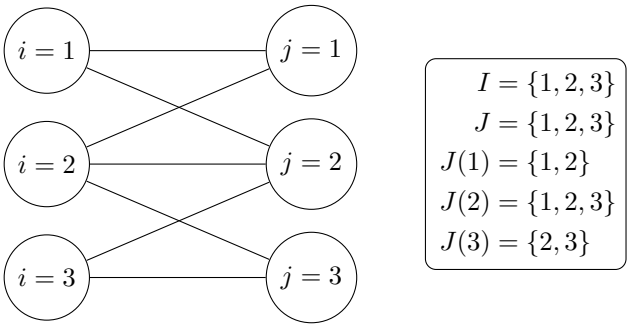


Figure 4.1: System of unrelated parallel machines with three machines and three products.

of parallel machines that are capable of performing the operation on product j. Such eligibility constraints are also referred to as capability constraints, assignment constraints, and process window constraints (Centeno and Armacost, 2004; Chung et al., 2008; Logendran and Subur, 2004). By duality, we know for every machine $i \in I$ the feasible product set $J(i) \subseteq J$ of products that can be processed by machine i. We assume that these sets are static, i.e., the allocation of the capacity of a machine has no effect on the feasible product set of any other machine.

Let $G = (V, E)$ be an undirected graph with the nodes $V = I \cup J$ and the edges E such that any two nodes $i \in I$ and $j \in J$ are connected by the edge ij if and only if machine i can process product j. A graph is connected if any two nodes are connected by a path. We say that I and J establish a system of unrelated parallel machines (SUPM) if I is a set of UPMs and graph G is connected. A specific instance is denoted as SUPM(I, J). In a SUPM(I, J), any product in J requires processing from at least one machine in I. Any two products in J compete with each other either directly or indirectly for capacity from machine set I. Figure 4.1 shows the connected graph of a SUPM with three machines and three products.

The processing time per item $p_{ij} \in \mathbb{R}_{>0}$ of product $j \in J(i)$ on machine $i \in I$ and the capacity per time unit $c_i \in \mathbb{R}_{>0}$ of machine $i \in I$ are known. Both p_{ij} and c_i are given in machine hours. The research question is how to generate the finite set of some l inequalities of the form

$$\sum_{j \in J} a_{hj} x_j \leq b_h \qquad \forall h \in \{1, 2, \ldots, l\} \tag{4.1}$$

that accurately model the capacity of a SUPM(I, J). The variable x_j in (4.1) stands for the total quantity of product j that is produced by the SUPM(I, J) per time unit. The problem is to determine the number of constraints l and the parameters $a_{hj}, b_h \in \mathbb{R}_{\geq 0}$ for all $j \in J$ and all $h \in \{1, 2, \ldots, l\}$ such that the constraints describe the set of feasible production quantities per time unit of the SUPM(I, J).

4.3 Related work

Linear programming formulations that model the capacity of parallel resources have been known for long time (see, e.g., Johnson and Montgomery, 1974). They became highly relevant when linear programming formulations were introduced for production planning with time lags (see, e.g., Hackman and Leachman, 1989). New concepts of capacity modeling, such as clearing functions for release planning, capture the nonlinear relation between throughput and WIP in the system (see, e.g., Selçuk et al., 2008). Linear capacity models remain however widely used in production planning due to their simplicity (see, e.g., Missbauer and Uzsoy, 2011). Leachman and Carmon (1992) denote such a model that is based on allocation variables as "step-separated formulation" (SSF). In the SSF, the allocation variable w_{ij} models the machine hours that machine i allocates to the processing of product j (see equation (4.2)). Inequality (4.3) ensures that the total workload does not exceed the capacity c_i of machine i.

$$x_j = \sum_{i \in I(j)} \frac{w_{ij}}{p_{ij}} \qquad \forall j \in J \tag{4.2}$$

$$\sum_{j \in J(i)} w_{ij} \leq c_i \qquad \forall i \in I \tag{4.3}$$

Bermon and Hood (1999) propose a variation of this formulation. In (4.4), the allocation variable w_{ij} represents the utilization of machine i attributable to the processing of product j. The capacity constraint (4.5) ensures that 100 % utilization is the upper bound.

$$x_j = \sum_{i \in I(j)} \frac{c_i}{p_{ij}} w_{ij} \qquad \forall j \in J \tag{4.4}$$

$$\sum_{j \in J(i)} w_{ij} \leq 1 \qquad \forall i \in I \tag{4.5}$$

We are interested in generating capacity constraints that do not require other decision variables than those that model the total production quantity per time unit of a product in J. Leachman and Carmon (1992) study this problem as well. The authors propose a capacity set generation procedure to generate the "direct product mix formulation" (DPF). The DPF fulfills the form requirement defined in (4.1). Yet the procedure fails to solve our problem because it generates accurate capacity constraints only for uniform parallel machines. Parallel machines are uniform if for every machine $i \in I$ and every product $j \in J(i)$, the equation $p_{ij} = \frac{p_j}{\mu_i}$ is true. Here, μ_i is the speed of machine i, which is defined relative to a reference machine. p_j is the processing time per item of product j on the reference machine. Note that unrelated parallel machines satisfy the equation $p_{ij} = \frac{p_j}{\mu_{ij}}$ for all $i \in I$ and all $j \in J(i)$ with the speed μ_{ij} being a function of both machine and product (Pinedo, 2012). Uniform parallel machines are therefore a special case of unrelated parallel machines.

Leachman and Carmon's capacity set generation procedure defines some l capacity sets $S_h \subseteq I$ with $h = 1, 2, \ldots, l$. These sets define l capacity constraints:

$$\sum_{j:I(j)\subseteq S_h} \alpha_i p_{ij} x_j \leq \sum_{i \in S_h} \alpha_i c_i \qquad \forall h \in \{1, 2, \ldots, l\}. \tag{4.6}$$

Here, $j : I(j) \subseteq S_h$ represents all products whose eligible machine sets are a subset of the capacity set S_h and α_i is a machine-specific scaling factor. The scaling factor α_i is chosen such that the scaled speeds of any pair of uniform parallel machines i, i' become identical, i.e., $\frac{\alpha_i}{\mu_i} = \frac{\alpha_{i'}}{\mu_{i'}}$. As a result, the processing times $p_{ij}, p_{i'j}$ of product $j \in J(i) \cap J(i')$ are identical.

In order to demonstrate Leachman and Carmon's capacity set generation procedure, we assume that the system illustrated in Figure 4.1 has the processing times and capacities presented in (4.7). Processing time $p_{ij} = \infty$ means that machine i cannot process product j and $j \notin J(i)$. Equation (4.7) shows that machine 2 takes twice as long as machine 1 for any product that can be processed on both machines. Likewise machine 3 takes twice as long as machine 2. Therefore, (4.7) satisfies the uniformity condition.

$$\begin{array}{cccc|cccc} p_{11} & p_{12} & p_{13} & c_1 & 1 & 2 & \infty & 20 \\ p_{21} & p_{22} & p_{23} & c_2 = & 2 & 4 & 6 & 35 \\ p_{31} & p_{32} & p_{33} & c_3 & \infty & 8 & 12 & 124 \end{array} \tag{4.7}$$

Applied to the system of uniform parallel machines in (4.7), Leachman and Carmon's capacity set generation procedure defines the three capacity sets $S_1 = \{1, 2\}$, $S_2 = \{2, 3\}$, and $S_3 = \{1, 2, 3\}$. The scaling factors are, for example, $\alpha_1 = 2$, $\alpha_2 = 1$, and $\alpha_3 = \frac{1}{2}$. All p_{ij} and all c_i are scaled as indicated in Step 1 in (4.8). In Step 2, the DPF is generated according to the definition in (4.6) for the capacity sets S_1, S_2, and S_3. The first inequality limits the capacity consumption of product 1 to the joint capacity of machines 1 and 2. The second inequality limits the capacity consumption of product 3 to the joint capacity of machines 2 and 3. The capacity consumption of all products is limited to the joint capacity of all machines in the third inequality.

$$\begin{array}{cccc} 1 & 2 & \infty & 20 \\ 2 & 4 & 6 & 35 \\ \infty & 8 & 12 & 124 \end{array} \xrightarrow[\alpha_1,\alpha_2,\alpha_3]{\text{Step 1}} \begin{array}{cccc} 2 & 4 & \infty & 40 \\ 2 & 4 & 6 & 35 \\ \infty & 4 & 6 & 31 \end{array} \xrightarrow[S_1,S_2,S_3]{\text{Step 2}} \begin{array}{cccc} 2x_1 & & & \leq 75 \\ & & 6x_3 & \leq 66 \\ 2x_1 & +4x_2 & +6x_3 & \leq 106 \end{array} \tag{4.8}$$

Hung and Cheng (2002) refer to Leachman and Carmon (1992) and acknowledge that the DPF does not require allocation variables. Hung and Cheng also point out that the uniformity condition is not satisfied in many industrial applications. To overcome this drawback and to make the procedure applicable to UPMs (that are only partially uniform), the authors propose an extension of Leachman and Carmon's capacity set generation procedure. This extension requires however allocation variables.

Hung and Cheng partition the set of products into blocks such that each block together with its sets of eligible machines defines a system of uniform parallel machines. If a machine is shared by two or more blocks, the allocation variable $w_{i\beta}$ is defined to allocate $w_{i\beta}$ machine hours from machine i to the processing of products in block β.

$$\begin{array}{cccc} p_{11} & p_{12} & p_{13} & c_1 \\ p_{21} & p_{22} & p_{23} & c_2 \\ p_{31} & p_{32} & p_{33} & c_3 \end{array} = \begin{array}{cccc} 1 & 2 & \infty & 20 \\ 2 & 4 & 6 & 35 \\ \infty & 4 & 12 & 124 \end{array} \tag{4.9}$$

Equation (4.9) defines a system of partially uniform parallel machines. Comparing the machines 2 and 3 reveals non-uniformity. If applied to (4.9), Hung and Cheng's extension divides the product set $J = \{1, 2, 3\}$ into the blocks $\{1, 2\}$ and $\{3\}$. The resulting systems of uniform parallel machines are presented on the left-hand side of (4.10) and (4.11). The capacity set generation procedure of Leachman and Carmon (1992) is applied to each block separately. The resulting inequalities include the scaled allocation variable $\alpha_{i\beta} w_{i\beta}$ if machine i is shared among two or more blocks and can process at least one product in block β. The constraints (4.12) and (4.13) ensure that the total workload on the shared machines 2 and 3 does not exceed capacity.

$$\text{Block 1}: \begin{array}{cccc} 1 & 2 & \infty & 20 \\ 2 & 4 & \infty & 35 \\ \infty & 4 & \infty & 124 \end{array} \xrightarrow[\alpha_{11},\alpha_{21},\alpha_{31}]{\text{Step 1}} \begin{array}{cccc} 2 & 4 & \infty & 40 \\ 2 & 4 & \infty & 35 \\ \infty & 4 & \infty & 124 \end{array} \xrightarrow[S_{11},S_{21}]{\text{Step 2}} \begin{array}{ll} 2x_1 & \leq 40 + w_{21} \\ 2x_1 + 4x_2 & \leq 40 + w_{21} + w_{31} \end{array} \tag{4.10}$$

$$\text{Block 2}: \begin{array}{cccc} \infty & \infty & 6 & 35 \\ \infty & \infty & 12 & 124 \end{array} \xrightarrow[\alpha_{12},\alpha_{22}]{\text{Step 1}} \begin{array}{cccc} \infty & \infty & 6 & 35 \\ \infty & \infty & 6 & 62 \end{array} \xrightarrow[S_{12}]{\text{Step 2}} 6x_3 \leq w_{22} + \tfrac{1}{2} w_{32} \tag{4.11}$$

$$w_{21} + w_{22} \leq 35 \tag{4.12}$$

$$w_{31} + w_{32} \leq 124 \tag{4.13}$$

At the time of writing, Liberopoulos (2002) is the only paper that examines the capacity modeling of SUPMs with low-dimensional capacity constraints. The author expresses the set of feasible production quantities as a convex hull of extreme points. Each facet of the convex hull corresponds to a capacity constraint. Liberopoulos provides binomial coefficients to quantify the number of extreme points as well as the number of facets in case every machine in I is perfectly flexible and can process each of the products in J, i.e., $J(i) = J$ for all $i \in I$. The study is, however, restricted to a special case of UPMs. It is assumed that for any two products, the ratio of processing times is different on any pair of machines. A constraint

generation procedure that specifies the parameters of the capacity constraints is not presented. Later, Fukuda and Weibel (2009) propose an ouptut-sensitive polynomial algorithm that solves the facet enumeration problem for polytopes relatively in general position, i.e., under the very assumption that Liberopoulos makes about processing times.

In summary, there exists a procedure to generate accurate, low-dimensional capacity constraints of uniform parallel machines, presented by Leachman and Carmon (1992). There is a gap in the scientific literature as there has not been presented a procedure that generates low-dimensional capacity constraints for UPMs in the general case, i.e., without any restrictions on processing times. This paper aims at filling this gap.

4.4 Generation of low-dimensional capacity constraints

4.4.1 Problem size reduction by aggregating uniform machines

Both Hung and Cheng (2002) and our analyses of field data from the semiconductor industry suggest that the assumption of uniform parallel machines does not hold for industrial applications. However, this does not mean that uniform parallel machines do not exist. In semiconductor manufacturing, for example, a younger machine often shows a process speed improvement compared to an older machine by a factor that is uniform across all products that can be processed by both machines. Given that a SUPM(I, J) includes some uniform machines, i.e., it is partially uniform, problem size can be reduced by aggregating uniform machines. Before we introduce the capacity constraint generation procedure, we therefore first discuss methods for problem size reduction.

The size of the constraint generation problem of a SUPM(I, J) grows with the cardinality of the sets I and J. Problem size reduction by aggregating uniform machines reduces the size of I, while the resulting capacity constraints remain accurate. Let $I_u \subseteq I$ to be a set of uniform machines. That means $J(i) = J(i')$ holds for all $i, i' \in I_u$ and $p_{ij} = \frac{p_j}{\mu_i}$ is true for all $i \in I_u$ and all $j \in J(i)$. We say that the machine set I_u can be aggregated to a representative machine i^* so that $I_u = \{i^*\}$. Let some $i^* \in I_u$ be the representative machine, which has the speed μ_{i^*}. Both the feasible product set $J(i^*)$ and the processing time parameters p_{i^*j} of the representative machine are not changed by the aggregation. The capacity of the representative machine is defined as the sum of scaled capacities of the represented machines. That means $c_{i^*} = \sum_{i \in I_u} \frac{\mu_i}{\mu_{i^*}} c_i$. The value $\frac{\mu_i}{\mu_{i^*}}$ describes the machine speed of i given in percent of the speed of representative machine i^*.

Aggregating uniform parallel machines with identical feasible product sets reduces the size of the machine set I. To give an example, let the left-hand side of (4.14) specify a SUPM(I, J) of four machines and four products. The subset $I_u = \{1, 2\}$ has identical feasible product sets since $J(1) = J(2) = \{1, 2, 3\}$ and the machines are uniform because the processing times of machine 2 are twice as long as the processing times of machine 1 across the feasible product set. We aggregate I_u, which results in $|I|= 3$, i.e., the size of the machine set I decreases by one.

$$\begin{array}{ccccc|c} p_{11} & p_{12} & p_{13} & p_{14} & c_1 & 1 & 3 & 2 & \infty & 15 \\ p_{21} & p_{22} & p_{23} & p_{24} & c_2 & 2 & 6 & 4 & \infty & 10 \\ p_{31} & p_{32} & p_{33} & p_{34} & c_3 & 2 & 6 & 4 & 6 & 35 \\ p_{41} & p_{42} & p_{43} & p_{44} & c_4 & \infty & \infty & 4 & 12 & 124 \end{array} \xrightarrow[\mu_{1^*}=1, \mu_2=\frac{1}{2}]{\text{Machine aggregation}} \begin{array}{ccccc} 1 & 3 & 2 & \infty & 20 \\ 2 & 6 & 4 & 6 & 35 \\ \infty & \infty & 4 & 12 & 124 \end{array} \tag{4.14}$$

4.4.2 Problem size reduction by aggregating uniform products

Problem size reduction by aggregating uniform products reduces the size of J, while the resulting capacity constraints remain accurate. We define $J_u \subseteq J$ to be a set of uniform products if and only if they have identical eligible machine sets and the eligible machines are uniform with respect to the products in J_u. That means $I(j) = I(j')$ holds for all $j, j' \in J_u$ and $p_{ij} = \frac{p_j}{\mu_i}$ is true for all $j \in J_u$ and all $i \in I(j)$.

Let any $j^* \in J_u$ be the representative product. Product set J_u can be aggregated to the representative product j^* so that $J_u = \{j^*\}$. As a result, the size of J decreases. Both the eligible machine set $I(j^*)$ and the processing times p_{i,j^*} do not change. We demonstrate the aggregation of uniform products with the outcome of aggregating uniform machines in (4.14). The aggregation of the uniform subset of products $J_u = \{1, 2\}$ is shown in (4.15) so that after aggregation $|J|= 3$.

$$\begin{array}{ccccc} 1 & 3 & 2 & \infty & 20 \\ 2 & 6 & 4 & 6 & 35 \\ \infty & \infty & 4 & 12 & 124 \end{array} \xrightarrow[p_{1^*}=1, p_2=3]{\text{Product aggregation}} \begin{array}{cccc} 1 & 2 & \infty & 20 \\ 2 & 4 & 6 & 35 \\ \infty & 4 & 12 & 124 \end{array} \tag{4.15}$$

Once the set of low-dimensional capacity constraints of the form (4.1) has been generated for a SUPM with aggregated products, the representative product has to be disaggregated again. Disaggregating provides the capacity constraints that include all the products of the original SUPM. The representative product j^* has the same eligible machine set and the same processing times (apart from uniform scaling) as the represented products. We disaggregate each of the generated capacity constraints (apart from the non-negativity constraints) that includes the representative product j^* by replacing the decision variable x_{j^*} with the linear combination $\sum_{j \in J_u} \frac{p_j}{p_{j^*}} x_j$. Since this linear combination is defined based on J_u, it includes the representative product j^* and the represented products. The value $\frac{p_j}{p_{j^*}}$ is the processing time of j given in percent of the processing time of j^*. Finally, one non-negativity constraint

is added for each x_j that has been added by disaggregation. This disaggregation step is demonstrated in Section 4.4.3 for the SUPM(I, J) that is defined on the right-hand side of (4.15). Note that in contrast to aggregated products, aggregated machines do not require disaggregation in order to ensure constraint accuracy.

4.4.3 Capacity constraint generation procedure

We propose a new procedure to generate the low-dimensional, irredundant, and accurate capacity constraints of a SUPM(I, J). Accurate means that the constraints define the same set of feasible production plans as the constraints according to the SSF from Leachman and Carmon (1992). Uniform machines and uniform products may have been aggregated but this is not necessary for the procedure to be functional.

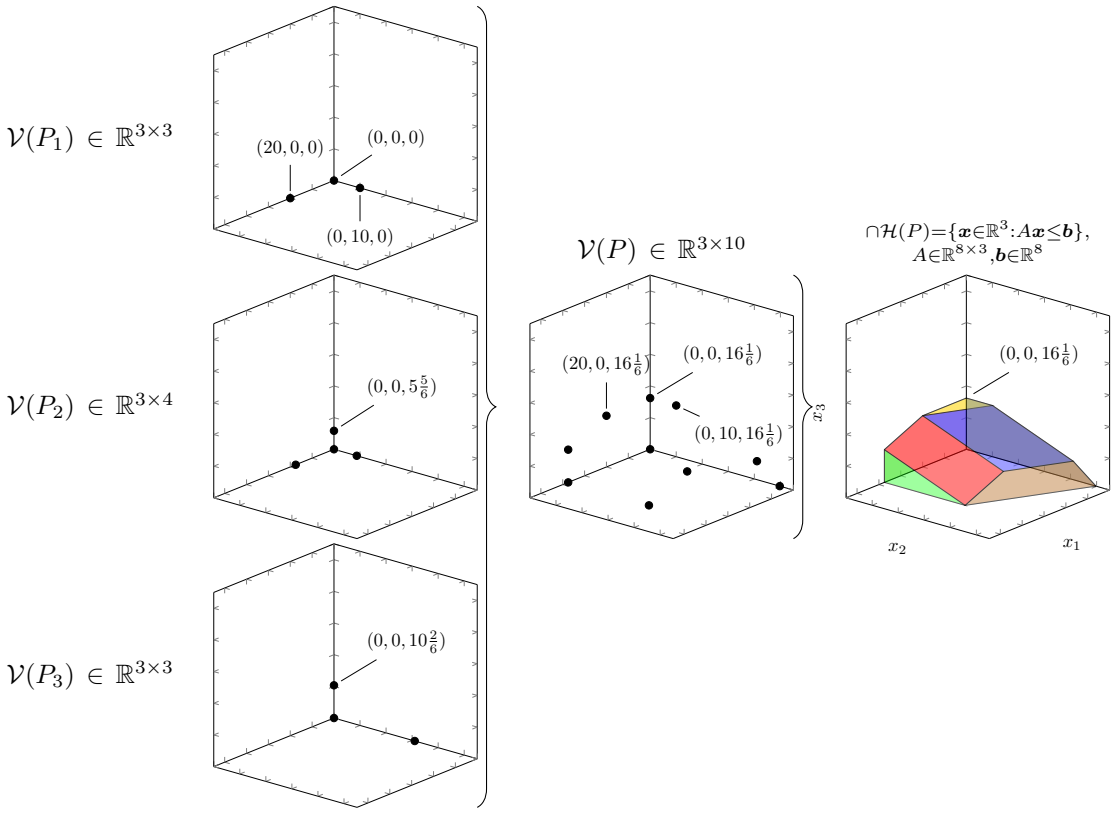
The procedure is based on the observation that the total production quantity of a SUPM(I, J) is identical to the sum of the production quantities of the m machines in I. The production quantity of a machine can be modeled as a vector in the n-dimensional space. The total production quantity of the SUPM(I, J) is therefore the sum of these vectors. We are looking for inequalities of the form (4.1) that define the set of vectors that are feasible subject to the capacity of the SUPM(I, J). The computation of these inequalities is divided into three steps:

- In Section 4.4.3, we show that the capacity of any machine $i \in I$ defines a polytope. We calculate this polytope P_i in vertex representation for every machine i.
- In Section 4.4.3, we show that the capacity of all machines in a SUPM(I, J) combined is given by the vector sum, also called Minkowski sum, of these polytopes. We calculate the Minkowski sum $P = \sum_{i \in I} P_i$ in vertex representation.
- In Section 4.4.3, we enumerate the facet-defining halfspaces of the Minkowski sum P, which provides the sought-after low-dimensional capacity constraints of the SUPM(I, J).

For the SUPM(I, J) that is sketched in Figure 4.1 and specified on the right-hand side of (4.15), these three steps are illustrated in Figure 4.2. The output of the procedure is a set of inequalities, which define the facet-defining halfspaces of the polytope illustrated in step 3. The facets are visualized in Figure 4.2 as colored surfaces and the corresponding inequalities are presented on the left-hand side of (4.25).

Calculation of the polytope P_i for each $i \in I$

The machine set I of a SUPM(I, J) can process n different products. The production quantity of any machine as well as the total production quantity of all machines combined can be modeled as a point in the n-dimensional Euclidean space $\mathbb{R}^n$. The value of the first, second, ..., nth coordinate is the production quantity of the first, second, ..., nth product. According



Step 1: Polytopes P_i in vertex representation for all machines $i \in I$

Step 2: Polytope $P = \sum_{i\in I} P_i$ in vertex representation

Step 3: Polytope $P = \sum_{i\in I} P_i$ in half-space representation

Figure 4.2: Three-step capacity constraint generation procedure.

to the main theorem of polytopes (see, e.g., Ziegler, 2007), convex polytopes can be specified in two ways: in vertex representation and in halfspace representation. In this section, we show that the set of feasible production quantities per time unit of any machine $i \in I$ defines a convex polytope, or just polytope, in $\mathbb{R}^n$. For this polytope, we first provide the well-known halfspace representation. Next, we derive the vertex representation and prove the equivalence. The vertex representation serves as the input to the calculation of the Minkowski sum in the following Section 4.4.3.

A polytope is a point set $P \subseteq \mathbb{R}^n$. A polytope in vertex representation is also called $\mathcal{V}$-polytope. A $\mathcal{V}$-polytope is defined as the convex hull of a finite set of points. If the point set $\mathcal{V} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k\} \subseteq \mathbb{R}^n$ is finite, then its convex hull $conv(\mathcal{V})$ is the set of all convex combinations of the points in $\mathcal{V}$:

$$conv(\mathcal{V}) = \left\{ \boldsymbol{x} : \boldsymbol{x} = \sum_{q=1}^{k} \lambda_q \boldsymbol{v}_q, \sum_{q=1}^{k} \lambda_q = 1, \lambda_q \in \mathbb{R}_{\geq 0} \right\}. \tag{4.16}$$

A polytope in halfspace representation is called $\mathcal{H}$-polytope. A $\mathcal{H}$-polytope is defined as the bounded intersection of finitely many closed halfspaces. A halfspace in $\mathbb{R}^n$ is defined by a hyperplane $\{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^\intercal \boldsymbol{x} = b\}$ where $\boldsymbol{a} \in \mathbb{R}^n$ is a non-zero vector, $\boldsymbol{a}^\intercal \boldsymbol{x}$ stands for the inner product of two vectors, and $b \in \mathbb{R}$. A hyperplane divides $\mathbb{R}^n$ into two halfspaces. A closed halfspace, denoted as H, is a halfspace of $\mathbb{R}^n$ unionized with its defining hyperplane, i.e., $H = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^\intercal \boldsymbol{x} \leq b\}$. If the set of closed halfspaces $\mathcal{H} = \{H_1, H_2, \ldots, H_l\}$ is finite, then its intersection is defined as

$$\cap\mathcal{H} = \{\boldsymbol{x} : \boldsymbol{x} \in H, \forall H \in \mathcal{H}\}. \tag{4.17}$$

"Bounded" means in this context that the intersection does not contain a ray $\{\boldsymbol{p} + \lambda(\boldsymbol{d} - \boldsymbol{p}) : \lambda \geq 0\}$ with any points $\boldsymbol{p}, \boldsymbol{d} \in \cap\mathcal{H}$ and an arbitrary large $\lambda \in \mathbb{R}$.

The feasible production quantities per time unit of a machine $i \in I$ are commonly modeled with the capacity constraints (4.18), (4.19) and (4.20) where x_{ij} represents the production quantity of product j on machine i. Let $\boldsymbol{x}_i = \begin{pmatrix} x_{i1} & x_{i2} & \ldots & x_{in} \end{pmatrix}^\intercal$ be the n-dimensional vector of the production quantities of machine i. The constraints (4.18) to (4.20) clearly put both an upper and lower bound on every coordinate of $\boldsymbol{x}_i$. Further, the constraints describe an intersection of finitely many closed halfspaces. As a result, the set of $\boldsymbol{x}_i$ that is feasible subject to the constraints (4.18) to (4.20) defines a $\mathcal{H}$-polytope in $\mathbb{R}^n$, which we call P_i.

The set of closed halfspaces that is defined by the inequalities (4.18) to (4.20) for machine $i \in I$ is irredundant. A set of closed halfspaces $\mathcal{H}$ is irredundant if there does not exist any $H \in \mathcal{H}$ such that $\cap\mathcal{H} = \cap(\mathcal{H} \backslash \{H\})$. In other words, it is impossible to remove any inequality from (4.18) to (4.20) so that the set of feasible $\boldsymbol{x}_i$ remains unchanged. We denote the set of irredundant closed halfspaces (4.18) to (4.20) that define P_i as $\mathcal{H}(P_i)$.

$$\sum_{j \in J(i)} p_{ij} x_{ij} \leq c_i \tag{4.18}$$

$$x_{ij} \leq 0 \qquad \forall j \in J \backslash J(i) \tag{4.19}$$

$$x_{ij} \geq 0 \qquad \forall j \in J \tag{4.20}$$

Now we define the vertex representation of P_i. Let $r_{ij} = \frac{c_i}{p_{ij}}$ be the maximum quantity of product $j \in J(i)$ that can be processed per time unit on machine i. Let $\boldsymbol{e}_j$ and $\mathbf{0}$ be the n-dimensional unit vector and the n-dimensional null vector, respectively. We denote $\boldsymbol{r}_{ij} = r_{ij}\boldsymbol{e}_j$ as the capacity vector of machine i given in units of product j. For machine i, let $R_i = \{\boldsymbol{r}_{ij} : \boldsymbol{r}_{ij} \neq \mathbf{0}, j \in J(i)\}$ be the set of capacity vectors different from the null vector. Since zero production is always feasible, $\boldsymbol{x}_i \in R_i \cup \{\mathbf{0}\}$ is feasible for all $i \in I$. Note that a vector $\boldsymbol{x}_i \in R_i \cup \{\mathbf{0}\}$ either models that machine i commits all its capacity to the processing of $\frac{c_i}{p_{ij}}$ units of product j or that the machine runs idle. Obviously the capacity vectors in R_i are orthogonal to each other and hence linearly independent. As a result, $R_i \cup \{\mathbf{0}\}$ describes the $k_i + 1$ extreme points (or vertices) of a k_i-dimensional polytope, i.e., a k_i-simplex, with $k_i = |R_i| = |J(i)|$.

Proposition 1. *For any machine $i \in I$, the set of $\boldsymbol{x}_i$ that are feasible subject to the constraints* (4.18) *to* (4.20) *is equal to the set $conv(R_i \cup \{\mathbf{0}\})$.*

Proof. For the proof, we refer to Appendix C.1. □

We denote the set of vertices of polytope P_i as $\mathcal{V}(P_i)$. A point $\boldsymbol{v}$ is a vertex of P_i if it is irredundant, i.e., if $conv(\mathcal{V}(P_i)) \neq conv(\mathcal{V}(P_i) \backslash \{\boldsymbol{v}\})$. Since the set $R_i \cup \{\mathbf{0}\}$ is the set of vertices of a k_i-simplex, $R_i \cup \{\mathbf{0}\}$ defines $\mathcal{V}(P_i)$ for all machines $i \in I$. $\mathcal{V}(P_i)$ is needed for the calculation of the Minkowski sum over all P_i's, which follows in the next section.

The $\mathcal{V}$-polytopes $conv(R_i \cup \{\mathbf{0}\})$ of the machines $i = 1$, 2, and 3 in the SUPM(I, J) that is specified on the right-hand side of (4.15) are plotted on the left-hand side of Figure 4.2. For example, the polytope of machine 1 with the vertices $(0, 0, 0)$, $(20, 0, 0)$, and $(0, 10, 0)$ in the upper-left corner of Figure 4.2 is two-dimensional because the right-hand side of (4.15) specifies that machine 1 can only process the products 1 and 2. If exclusively used for product 1, the capacity is $\frac{c_1}{p_{11}} = \frac{20}{1} = 20$ units per time period and if exclusively used for product 2, it is $\frac{c_1}{p_{12}} = \frac{20}{2} = 10$ units per time period.

Calculation of the Minkowski sum $P = \sum_{i \in I} P_i$

We have shown that for any machine $i \in I$, the production quantities per time unit can be modeled as a vector $\boldsymbol{x}_i \in \mathbb{R}^n$. Let the total production quantities per time unit of a SUPM(I, J) be modeled as the total production vector $\boldsymbol{x} \in \mathbb{R}^n$. Since the machines in I are assumed to be independent, $\boldsymbol{x}$ is the sum of vectors $\boldsymbol{x}_i$ of the machines that constitute the SUPM(I, J):

$$\boldsymbol{x} = \boldsymbol{x}_1 + \boldsymbol{x}_2 + \ldots + \boldsymbol{x}_m : \boldsymbol{x}_i \in P_i \qquad \forall i \in I. \tag{4.21}$$

We are interested in the capacity of the SUPM(I, J), i.e., the definition of the set of all feasible total production vectors $\boldsymbol{x}$. The set of all feasible total production vectors $\{\boldsymbol{x}_1 + \boldsymbol{x}_2 + \ldots + \boldsymbol{x}_m : \boldsymbol{x}_i \in P_i, \forall i \in I\}$ is indeed the Minkowski sum of the polytopes P_1, P_2, ..., P_m. Let P be the Minkowski sum of P_1, P_2, ..., P_m, then it holds

$$P = P_1 + P_2 + \ldots + P_m = \{\boldsymbol{x}_1 + \boldsymbol{x}_2 + \ldots + \boldsymbol{x}_m : \boldsymbol{x}_i \in P_i, \forall i \in I\}. \tag{4.22}$$

According to the main theorem of polytopes, the Minkowski sum of polytopes is again a polytope. That guarantees the existence of a halfspace representation of P, which is the very set of inequalities of the form (4.1) that we are looking for. Before we can enumerate these inequalities, we have to compute the Minkowski sum. In terms of computational complexity, the easiest way to calculate the Minkowski sum P is to compute the vertex representation of P based on the vertex representations of P_1, P_2, ..., P_m. The inputs to this calculation are the vertex sets $\mathcal{V}(P_1)$, $\mathcal{V}(P_2)$, ..., $\mathcal{V}(P_m)$, which we have defined in the previous section. The output is the set of vertices $\mathcal{V}(P)$ such that

$$P = conv(\mathcal{V}(P)) \qquad \mathcal{V}(P) \in \mathbb{R}^{n \times k}, \tag{4.23}$$

where $\mathcal{V}(P)$ is a set of some k vertices in the n-dimensional space.

In Figure 4.2, the set of vertices of the Minkowski sum $\mathcal{V}(P) = \{(0,0,0),\ (0,0,16\frac{1}{6}),\ (0,10,16\frac{1}{6}),\ (0,41,5\frac{5}{6}),\ (0,49\frac{3}{4},0),\ (20,0,16\frac{1}{6}),\ (20,31,5\frac{5}{6}),\ (37.5,0,0),\ (37.5,0,10\frac{1}{3}),\ (37.5,31,0)\}$ is plotted in Step 2.

Enumeration of the facets of the Minkowski sum P

Once $\mathcal{V}(P)$ is given, it remains to solve the convex hull problem, i.e., to enumerate the set of some l halfspace-defining inequalities $A\boldsymbol{x} \leq \boldsymbol{b}$ such that

$$conv(\mathcal{V}(P)) = \cap\mathcal{H}(P) = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \leq \boldsymbol{b}\} \qquad A \in \mathbb{R}^{l \times n}, \boldsymbol{b} \in \mathbb{R}^l. \tag{4.24}$$

The inequalities $A\boldsymbol{x} \leq \boldsymbol{b}$ model the feasible production quantities per time unit of the underlying SUPM(I, J) accurately. Since every product defines one dimension of P, P must be full-dimensional, i.e., P is an n-dimensional polytope in $\mathbb{R}^n$. As P is full-dimensional, every irredundant inequality in $\mathcal{H}(P)$ corresponds to a facet of P; that is, an $(n-1)$-dimensional face of P. Given that the set of facet-defining inequalities $A\boldsymbol{x} \leq \boldsymbol{b}$ does not include any multiples, it is irredundant (for more information on this topic see, e.g., Ziegler, 2007).

For the example that is defined on the right-hand side of (4.15), the facets are visualized by colored surfaces in Step 3 in Figure 4.2. The corresponding set of irredundant linear inequalities $A\boldsymbol{x} \leq \boldsymbol{b}$ is presented on the left-hand side of (4.25). This is the result of the capacity constraint generation procedure. Each inequality corresponds to one surface in Figure 4.2. Each point in $\mathcal{V}(P)$ satisfies at least three of these inequalities by equality. For example, the point $(0, 0, 16\frac{1}{6})$, which lies on the x_3-axis above the origin in Step 3, satisfies by equality the three inequalities that are marked by right-arrows on the left-hand side of (4.25).

Recall that the input to the constraint generation procedure in this example includes one representative product from the aggregation of uniform products. We disaggregate this representative product. Otherwise, there will be some product j, which is part of the original SUPM, i.e., before aggregation, but not included in the capacity constraints. The disaggregation is shown in (4.25). A decision variable (see the down-arrow) and its non-negativity constraint is added to the set of inequalities.

The set of inequalities $A\boldsymbol{x} \leq \boldsymbol{b}$ on the right-hand side of (4.25) is an accurate description of the feasible total production quantities per time unit of the SUPM that has been defined on the left-hand side of Equation (4.14). The inequalities match the form requirement (4.1). Further, the set of inequalities is irredundant. There does not exist any accurate, low-dimensional capacity model with a lower number of constraints.

$$\begin{matrix} \\ \\ \\ \\ \rightarrow \\ \rightarrow \\ \rightarrow \\ \\ \end{matrix}\begin{pmatrix} 1 & 2 & 6 \\ 1 & 2 & 3 \\ 1 & 1 & 3 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \leq \begin{pmatrix} 117 \\ 99.5 \\ 68.5 \\ 37.5 \\ 16\frac{1}{6} \\ 0 \\ 0 \\ 0 \end{pmatrix} \xrightarrow[\substack{p_{1^*}=1 \\ p_2=3}]{\substack{\text{Product} \\ \text{disaggregation}}} \begin{pmatrix} 1 & \overset{\downarrow}{3} & 2 & 6 \\ 1 & 3 & 2 & 3 \\ 1 & 3 & 1 & 3 \\ 1 & 3 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \leq \begin{pmatrix} 117 \\ 99.5 \\ 68.5 \\ 37.5 \\ 16\frac{1}{6} \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{4.25}$$

4.4.4 Computational complexity and algorithms

The presented capacity constraint generation procedure includes two problems on convex polytopes that are well-known in computational geometry. The first problem is the Minkowski addition of $\mathcal{V}$-polytopes. The second is the facet enumeration problem, also called convex hull problem. The procedure can be summarized as

$$\mathcal{V}(P_1), \mathcal{V}(P_2), \ldots, \mathcal{V}(P_m) \xrightarrow{\substack{\text{Minkowski} \\ \text{addition problem}}} \mathcal{V}(P) \xrightarrow{\substack{\text{Facet enumeration} \\ \text{problem}}} \mathcal{H}(P). \tag{4.26}$$

The size of a polytope P in halfspace representation and in vertex representation is given by $n|\mathcal{H}(P)|$ and $n|\mathcal{V}(P)|$, respectively, with n being the dimension of P. Here, the length of the encoded numbers appearing in the description of the polytopes is ignored for simplicity. For both problems in (4.26) do exist examples showing that the output size can grow exponentially in the input size (Fukuda and Weibel, 2007). However, there are also examples in which the output size is bounded by the input size (Fukuda, 2004). The Minkowski addition presented in Figure 4.2 is such an example. This diversity in output size justifies that the computational complexity, i.e., the run time, of algorithms for polytopes is commonly stated as a function of both input size and output size. An algorithm is denoted as polynomial output-sensitive if its run time is bounded by a polynomial in the size of both input and output (Fukuda, 2004).

For the Minkowski addition problem, Fukuda (2004) proposes a polynomial output-sensitive algorithm that computes the Minkowski sum in any dimension n and for any number of summands m that are given as $\mathcal{V}$-polytopes. It is a parallelizable algorithm based on reverse search and linear programming. The algorithm runs in time $O(mn\mathrm{LP}(n, mn)|\mathcal{V}(P)|)$ and space linear in the input size, where $\mathrm{LP}(\alpha, \beta)$ is the time necessary to solve a linear program with α rows and β columns.

Algorithms that solve the convex hull problem in general dimensions can be broadly divided into incremental or insertion algorithms and pivoting or graph traversal algorithms. Note that the enumeration of $\mathcal{H}(P)$ from $\mathcal{V}(P)$ is equivalent to the enumeration of $\mathcal{V}(P^*)$ from $\mathcal{H}(P^*)$ with P^* being the dual of P. Avis et al. (1997) shows that both types of algorithms, incremental and pivoting, have superpolynomial worst case run times in n, $|\mathcal{H}(P)|$, and $|\mathcal{V}(P)|$ and cannot be considered as polynomial output-sensitive.

Pivoting algorithms, such as the reverse search method from Avis and Fukuda (1992), are parallelizable. In case each facet contains exactly n vertices, reverse search is polynomial output-sensitive and solves the facet enumeration problem in $O(n|\mathcal{H}(P)||\mathcal{V}(P)|)$ time and $O(n|\mathcal{V}(P)|)$ space. A disadvantage is that the run time performance deteriorates in the degenerate case, i.e., for n-dimensional polytopes with facets that contain more than n vertices.

Incremental vertex enumeration algorithms, such as the double description method from Motzkin et al. (1953), compute the intersection of a set of halfspaces $\mathcal{H}(P) = \{H_1, H_2, \ldots, H_l\}$ by inductively computing $P_{l-1} = \cap_{i=1}^{l-1} H_i$ to eventually compute $P_{l-1} \cap H_l$. The algorithm is reported to perform surprisingly well for highly degenerate cases (Fukuda and Prodon, 1996). A disadvantage is that the intermediate polytopes can count many more vertices than the final polytope, which can make the problem intractable (Avis et al., 1997).

$$\mathcal{H}(P_1), \mathcal{H}(P_2), \ldots, \mathcal{H}(P_m) \xrightarrow{\substack{\text{Minkowski} \\ \text{addition problem}}} \mathcal{H}(P) \tag{4.27}$$

The constraint generation procedure proposed in Section 4.4.3 raises the natural question if the facets of the Minkowski sum can be enumerated directly from summands in halfspace representation, such as indicated in (4.27). The input $\mathcal{H}(P_1)$, $\mathcal{H}(P_2)$, ..., $\mathcal{H}(P_m)$ is readily available (see Section 4.4.3). The advantage would be to avoid both the enumeration of the potentially large vertex representation $\mathcal{V}(P)$ and the solution of the convex hull problem. At the time of writing, however, no algorithm exists that generally solves the problem in (4.27) faster than the procedure proposed in this paper. Fukuda and Weibel (2009) propose a polynomial output-sensitive algorithm but only for the special case when the faces of the summand polytopes are oriented in generic directions. Moreover, Tiwary (2008) proves that there cannot exist a polynomial output-sensitive algorithm that solves the problem in (4.27) in general unless P = NP.

4.5 Partition-based constraint search heuristic

In order to be useful for practical applications, the run time of the capacity constraint generation procedure has to be limited by an upper bound T. The actual run time t can grow very quickly because of the computational complexity and output size. We propose a heuristic that generates the most accurate capacity constraints possible in $t < \gamma T$ where γ is the number of iterations that the heuristic needs to finish.

The idea of the heuristic is to divide a SUPM(I, J), which requires run time $t > T$, into a set of smaller SUPMs that require in total $t < T$ for capacity constraint generation. These smaller SUPMs are defined by partitioning the graph $G = (V, E)$. G is partitioned by dividing the node set $V = I \cup J$ into some $\kappa \geq 2$ disjoint blocks of roughly equal size such that the objective function is minimized. The objective function is the sum of edge weights of edges that have to be cut in order to make the blocks disjoint. An imbalance constraint, which has to be satisfied by all blocks, ensures that the number of nodes per block does not exceed $(1 + \upsilon) \left\lceil \frac{|V|}{\kappa} \right\rceil$ with some $\upsilon \geq 0$. This problem is called the $(\kappa, 1 + \upsilon)$-balanced graph partitioning problem in graph G (Sanders and Schulz, 2013).

We define the edge weight of any edge $ij \in E$ as the product of the number of machines represented by node i and the number of products represented by node j. That gives every edge of the graph (before uniform elements have been aggregated) the weight one. In case G is partitioned after uniform machines and uniform products have been aggregated, a node can represent a set of aggregated machines or a set of aggregated products. The edge weight thus equals the number of edges that would appear if all nodes were disaggregated.

Edges that run between blocks are cut to define κ separate blocks, i.e., at least κ independent SUPMs. A smaller κ and a greater υ tend to lead to fewer edge cuts. We denote this relation as the monotonicity of the sum of weighted edge cuts in κ and υ. Let $cuts(\kappa, \upsilon, G)$ be the objective function value of the optimal solution of the $(\kappa, 1 + \upsilon)$-balanced graph partition

problem in G, i.e., the smallest sum of weighted edge cuts that can be achieved by a $(\kappa, 1+\upsilon)$-balanced partition of G. For a given imbalance υ, $cuts(\kappa, \upsilon, G)$ is monotonically increasing in the number of blocks κ (see (4.28)). For a given number of blocks κ, $cuts(\kappa, \upsilon, G)$ is monotonically decreasing in the imbalance υ (see (4.29)).

$$\kappa' > \kappa \Rightarrow cuts(\kappa', \upsilon, G) \geq cuts(\kappa, \upsilon, G) \qquad \kappa', \kappa \in K\ \upsilon \in Y \tag{4.28}$$

$$\upsilon' > \upsilon \Rightarrow cuts(\kappa, \upsilon', G) \leq cuts(\kappa, \upsilon, G) \qquad \kappa \in K\ \upsilon', \upsilon \in Y \tag{4.29}$$

There is a trade-off between constraint accuracy and block size. Cutting the edge ij means that the capability of machine i to process product j will not be captured by the capacity constraints. This makes the constraints inaccurate. While a smaller κ and a greater υ increase the accuracy of the generated capacity constraints as fewer edges tend to be cut, the resulting constraint generation problems become harder to solve and run time increases. Our objective is to find the $(\kappa, 1+\upsilon)$-balanced partition that minimizes $cuts(\kappa, \upsilon, G)$ subject to that the resulting constraint generation problem can be solved in $t < T$. Let K and Y be the domains of κ and υ, respectively. A naive approach is complete enumeration, i.e., attempting to solve the capacity constraint generation problems that are defined by the $(\kappa, 1+\upsilon)$-balanced partitions for all $(\kappa, \upsilon) \in K \times Y$. In order to avoid that, we propose the partition-based constraint search (PCS) heuristic.

In preparation of the PCS, we reduce the size of the search space $K \times Y$ by discretizing the range of υ and by defining upper bounds for both κ and υ. A natural upper bound of υ is 1 because for any $\kappa \geq 2$, $(1+1)\left\lceil \frac{|V|}{\kappa} \right\rceil \geq (1+0)\left\lceil \frac{|V|}{\kappa-1} \right\rceil$, i.e., for any κ an $\upsilon \in [0,1]$ exists such that the $(\kappa, 1+\upsilon)$-balanced partition equals the $(\kappa-1, 1+0)$-balanced partition. The upper bound of κ is defined such that the $(\kappa, 1+0)$-balanced partition provides a constraint generation problem that is solvable in $t < T$.

Let $\mathbf{K}$ and $\mathbf{Y}$ be sorted sets with zero-based indexing that have the same elements as K and Y. $\mathbf{K}[k]$ and $\mathbf{Y}[y]$ represent the kth and yth elements of $\mathbf{K}$ and $\mathbf{Y}$. The PCS heuristic is divided into two phases. Both phases call the function $cons(\kappa, \upsilon, G, T)$, which returns true if and only if the capacity constraint generation procedure successfully computes the low-dimensional capacity constraints of all SUPMs that are defined by the $(\kappa, 1+\upsilon)$-balanced partition in $t < T$.

Phase 1 starts with the smallest $\upsilon \in Y$ and largest $\kappa \in K$. Then, κ is iteratively decremented. The purpose of phase 1 is to find the $(\kappa, 1+\upsilon)$-balanced partition with the smallest $\upsilon \in Y$ and the largest $\kappa \in K$ that defines a constraint generation problem unsolvable in $t < T$. This partition defines the starting point of phase 2. The PCS heuristic is illustrated in Figure 4.3 with $\mathbf{K} = \{3,4,5,6\}$ and $\mathbf{Y} = \{0, 0.1, 0.2, 0.3\}$. Phase 1 is indicated by dashed arrows and rectangle marks, which are white in case the corresponding capacity

PCS heuristic Partition-based constraint search heuristic.

```
1: procedure FindCapacityConstraints(K, Y, G, T)
2:     k ← K.length − 1                                          ▷ Phase 1
3:     y ← 0
4:     while k ≥ 0 & cons(K[k], Y[y], G, T) do
5:         k ← k − 1
6:         cuts* ← cuts(K[k], Y[y], G)
7:     end while
8:     k ← k + 1                                                 ▷ Phase 2
9:     y ← y + 1
10:    while k < K.length & y < Y.length do
11:        if cons(K[k], Y[y], G, T) then
12:            if cuts(K[k], Y[y], G) < cuts* then
13:                cuts* ← cuts(K[k], Y[y], G)
14:                cons* ← cons(K[k], Y[y], G, T).getConstraints()
15:            end if
16:            y ← y + 1                                         ▷ Increase block size
17:        else
18:            k ← k + 1                                         ▷ Decrease block size
19:        end if
20:    end while
21:    return cons*
22: end procedure
```

constraint generation problem is solved in $t < T$ and black otherwise. For $\upsilon = \mathbf{Y}[0] = 0$, first $\kappa = \mathbf{K}[3] = 6$, then $\kappa = \mathbf{K}[2] = 5$, and finally $\kappa = \mathbf{K}[1] = 4$ are evaluated. The end of phase 1 is reached when the capacity constraints of the SUPMs defined by the $(4, 1 + 0)$-balanced partition cannot be generated in $t < T$.

Phase 2 explores the frontier of capacity constraint generation problems that are solvable in $t < T$ by iteratively increasing υ or κ. If $cons(\kappa, \upsilon, G, T)$ returns true, υ is increased, i.e., we are looking for a partition with fewer weighted edge cuts and the constraint generation problem becomes harder. If $cons(\kappa, \upsilon, G, T)$ returns false, κ is increased, i.e., we are looking for an easier constraint generation problem and the number of weighted edge cuts increases. Given that the monotonicity assumption is correct, these steps efficiently enumerate the frontier because for every $\upsilon \in Y$, the smallest $\kappa \in K$ is found that results in a capacity constraint generation problem that is solvable in $t < T$. In Figure 4.3, phase 2 is indicated by solid arrows and circular marks. As long as the capacity constraint generation problem is solved in $t < T$ (white marks), y is incremented. Once the problem is not solvable in $t < T$, k is incremented until the resulting constraint generation problem is solvable again in $t < T$.

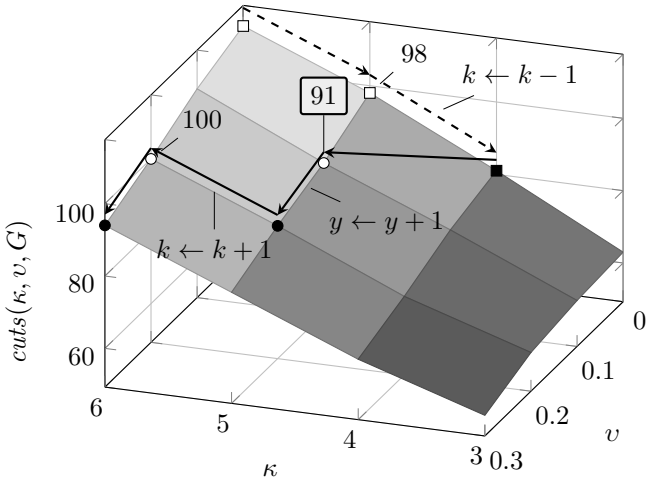


Figure 4.3: Phase 1 and phase 2 of the PCS heuristic.

The PCS heuristic returns a set of capacity constraints $cons^*$ based on the $(\kappa, 1+\upsilon)$-balanced partition that minimizes $cuts(\kappa, \upsilon, G)$ over all $(\kappa, \upsilon) \in K \times Y$ subject to $t < T$. In Figure 4.3, candidate partitions have white marks and the minimum is reached by $cuts(5, 0.1, G) = 91$. The worst case total run time of the PCS heuristic is the number of calls of the function $cons(\kappa, \upsilon, G, T)$ times T.

4.6 Experimental results

In the following Section 4.6.1, we demonstrate the aggregation of uniform elements, the capacity constraint generation procedure, and the PCS heuristic on seven SUPMs found in semiconductor device manufacturing. First, we apply the constraint generation procedure directly, which provides the accurate, low-dimensional capacity constraints of two SUPMs (see Section 4.6.1). The aggregation of uniform elements in Section 4.6.1 then increases this number to three. Since four out of seven SUPMs remain unsolved, we eventually use the PCS heuristic in Section 4.6.1, which generates sets of low-dimensional capacity constraints for the remaining four SUPMs. As the PCS heuristic trades constraint accuracy off against computation time, we evaluate constraint accuracy in comparison with the DPF proposed by Leachman and Carmon (1992). The sample of seven SUPMs is too small to allow firm conclusions about the performance of the PCS heuristic. In Section 4.6.2, we therefore conduct a full factorial experiment with randomly generated problem instances that resemble the SUPMs found in the semiconductor industry to determine the performance of the PCS heuristic.

Table 4.1: Problem instances and results of the capacity constraint generation procedure.

SUPM	$\|J\|$	$\|I\|$	ρ [%]	$\frac{\sum_{i\in I}\|\mathcal{V}(P_i)\|}{\sum_{i\in I}\|\mathcal{H}(P_i)\|}$	$\|\mathcal{V}(P)\|$	$\|\mathcal{H}(P)\|$	$\overline{icd}$	`minksum`	`cddlib`	`plrs`
1	58	3	51.1	92	15,778	104	2,369.5	6,432s	50,701s	>48h
2	491	4	35.7	706	n/a	n/a	n/a	>48h	n/a	n/a
3	474	5	30.7	732	n/a	n/a	n/a	>48h	n/a	n/a
4	484	6	23.0	675	n/a	n/a	n/a	>48h	n/a	n/a
5	24	7	22.0	44	34,428	51	10,344.3	986s	134,950s	>48h
6	490	7	41.0	1,413	n/a	n/a	n/a	>48h	n/a	n/a
7	325	30	8.9	898	n/a	n/a	n/a	>48h	n/a	n/a

4.6.1 Experiments with field data

The bottleneck machines of the testing operations at a German semiconductor manufacturer form independent SUPMs. Seven of these SUPMs are described in Table 4.1. Table 4.1 presents the number of products $|J|$, the number of machines $|I|$, and the density ρ. ρ is the fraction of product-machine tuples $(i, j) \in I \times J$ for which a processing time $p_{ij} \in \mathbb{R}_{>0}$ exists. The number of products, which defines dimension n, ranges between 24 and 491. The number of machines, which defines the number of summand polytopes, ranges between 3 and 30. The density ρ is between 8.9 % and 51.1 %. Since each summand polytope P_i of a SUPM(I, J) is a k_i-simplex, it has both $k_i + 1$ vertices and $k_i + 1$ facets. The sum of vertices over all summand polytopes $\sum_{i\in I}|\mathcal{V}(P_i)|$ equals the sum of facets over all summand polytopes $\sum_{i\in I}|\mathcal{H}(P_i)|$. The seven SUPMs describe summand polytopes with a total of 44 to 1,413 vertices and facets.

Direct application of the capacity constraint generation procedure

We solve the Minkowski addition problem in this and the following experiments with the `minksum` program (release 1.7) from Weibel (2010), which is a parallel implementation of the reverse search-based algorithm from Fukuda (2004). The convex hull problem is solved with Fukuda's implementation of the double description method `cddlib` (release 0.94g) (Fukuda, 2008) as well as the parallel implementation of the reverse search method `plrs` (release 5.0) from Avis and Roumanis (2013). All applications are compiled with the GNU Multiple Precision Arithmetic Library. Computations are performed by an Ubuntu 12.04 workstation with 32 GB of memory and an Intel Xeon CPU E3-1220 v2 with 3.1 GHz clock speed on 4 cores while no other application is running. The parallel implementations `minksum` and `plrs` run four threads in parallel (one thread per core), whereas `cddlib` runs a single thread on a single core.

For two out of seven SUPMs, the constraint generation procedure completes all computations in less than 48 hours, which was set as a time limit. Table 4.1 presents the results. $|\mathcal{V}(P)|$ and $|\mathcal{H}(P)|$ are the number of vertices and the number of facets of the Minkowski sum. $|\mathcal{H}(P)|$ is in fact the number of capacity constraints including $|J|$ non-negativity constraints. Ignoring the non-negativity constraints, the procedure generates 46 and 27 irredundant ca-

Table 4.2: The effect of aggregating uniform elements.

SUPM	$\|J\|$	$\|I\|$	ρ [%]	$\frac{\sum_{i\in I}\|\mathcal{V}(P_i)\|}{\sum_{i\in I}\|\mathcal{H}(P_i)\|}$	$\|\mathcal{V}(P)\|$	$\|\mathcal{H}(P)\|$	$\overline{icd}$	`minksum`	`cddlib`	`plrs`
1'	14	3	66.7	31	416	60	98.1	5s	10s	3,918s
2'	15	3	60.0	30	428	57	108.2	5s	10s	25,354s
3'	42	5	41.9	93	n/a	n/a	n/a	>48h	n/a	n/a
4'	20	6	29.2	41	28,410	n/a	n/a	801s	>48h	>48h
5'	10	7	25.7	25	1,033	37	329.6	4s	33s	15,287s
6'	49	5	51.4	131	n/a	n/a	n/a	>48h	n/a	n/a
7'	116	30	10.8	405	n/a	n/a	n/a	>48h	n/a	n/a

pacity constraints for the SUPMs 1 and 5. The columns `minksum`, `cddlib`, and `plrs` provide computation times. The run times of `cddlib` can be considered as too long for practical applications. The facet enumeration of SUPM 1 requires about 14 hours and of SUPM 5 about 37.5 hours, respectively. For five out of seven SUPMs, we stop the Minkowski addition after 48 hours of computation with no result.

The column $\overline{icd}$ provides the average number of incident vertices per facet. A vertex is said to be incident to a facet if it satisfies the facet-defining inequality by equality. A facet enumeration problem is called degenerate if there are more than n vertices incident to a facet of a n-dimensional polytope. The facet enumeration problems of the SUPMs 1 and 5 are highly degenerate. This explains why the run time performance of `plrs` is worse than of `cddlib`, which is less sensitive to degeneracy. We stop `plrs` after 48 hours without results.

The effect of aggregating uniform machines and products

Using the same problem instances as in the previous section, we now reduce problem size by aggregating uniform machines and uniform products before we apply the capacity constraint generation procedure. The SUPMs whose uniform elements have been aggregated are marked with a single prime. Table 4.2 presents the results.

The aggregation of uniform elements reduces problem size. Compared to the original values in Table 4.1, the number of machines $|I|$ decreases from 4 to 3 in SUPM 2' and from 7 to 5 in SUPM 6'. The number of products $|J|$ decreases between 58.3 % (SUPM 5') and 96.9 % (SUPM 2'). Both the sum of vertices $\sum_{i\in I}|\mathcal{V}(P_i)|$ and the sum of facets $\sum_{i\in I}|\mathcal{H}(P_i)|$ over all summand polytopes decrease between 43.2 % (SUPM 5') and 95.8 % (SUPM 2').

The output size of the constraint generation procedure decreases as well. The number of vertices of the Minkowski sum $|\mathcal{V}(P)|$ as well as the average size of incidence sets $\overline{icd}$ decrease in SUPMs 1' and 5' by around 96.3 %. The computation times of `minksum` and `cddlib` decrease by over 99.6 %. The number of capacity constraints $|\mathcal{H}(P)|$ decrease by 42.3 % in SUPM 1' and 27.5 % in SUPM 5'. The difference in number of capacity constraints between

the SUPMs 1 and 5 in Table 4.1 and the aggregated SUPMs 1' and 5' in Table 4.2 is caused by the difference in the number of products $|J|$. Non-negativity constraints are eliminated from $\mathcal{H}(P)$ because of the projection into a lower dimension. Note that disaggregation will reverse this effect by adding dimensions and non-negativity constraints.

The computational burden caused by degeneracy can now be quantified. For the convex hull problems of SUPMs 1', 2', and 5', the reverse search implementation `plrs` requires 392, 2,535, and 463 times as long as the double description method implemented in `cddlib`. The facet enumeration problems are still highly degenerate after the aggregation of uniform elements, which makes `plrs` less suitable. In the remaining experiments, we therefore solve facet enumeration problems solely with `cddlib`.

Application of the PCS heuristic

The capacity constraint generation procedure does not finish computing the capacity constraints of the SUPMs 3', 4', 6', and 7' within 48 hours even though uniform elements have been aggregated. In this section, we describe the implementation of the PCS heuristic and illustrate its logic at the example of SUPM 7'. We define two measures of constraint accuracy. We apply the PCS heuristic to the SUPMs 3', 4', 6', and 7', and then determine the accuracy of the resulting capacity constraints in comparison with the DPF.

We implement the PCS heuristic using the distributed evolutionary algorithm `KaFFPaE`, which is part of the partitioning framework KaHIP (release 0.62) from Sanders and Schulz (2013). `KaFFPaE` solves the $(\kappa, 1+\upsilon)$-balanced partition problem heuristically as it is NP-hard (see, e.g., Garey and Johnson, 1979). We set the timeout of `KaFFPaE` to 180 seconds per partition problem. Initial runs have shown that 180 seconds suffice for the sum of weighted edge cuts $cuts(\kappa, \upsilon, G)$ in the graphs of the SUPMs 3', 4', 6', and 7' to converge. Initial runs have also shown that the constraint generation procedure computes the capacity constraints within a few seconds if the graph includes at most 15 nodes. SUPM 7' defines the largest graph in Table 4.2 and has a total of 146 nodes. We use therefore $\left\lceil \frac{|V|}{15} \right\rceil = \lceil \frac{146}{15} \rceil$ as an upper bound for κ so that the domain of κ is defined as $\mathbf{K} = \{1, 2, \ldots, 10\}$. The domain of υ is defined as $\mathbf{Y} = \{0.0, 0.2, \ldots, 1.0\}$. We run the PCS heuristic with a time limit of $T = 3{,}600$ seconds. Phase 1 starts with $\kappa = 10$ and $\upsilon = 0$.

For the purpose of demonstration, we apply the PCS heuristic to SUPM 7' and present the results in a surface plot in Figure 4.4. The surface illustrates the relationship between κ, υ, and the sum of weighted edge cuts $cuts(\kappa, \upsilon, G)$ of the underlying $(\kappa, 1+\upsilon)$-balanced partitions. Phase 1 and phase 2 of the PCS heuristic are indicated by rectangle marks and circle marks, respectively. These marks are white in case the underlying capacity constraint generation problem is solved in $t < T$ and black otherwise. The frontier of constraint generation problems that are solvable in $t < T$ forms roughly a diagonal of white circles across the surface. Pins are labeled with the value of $cuts(\kappa, \upsilon, G)$. The lowest objective

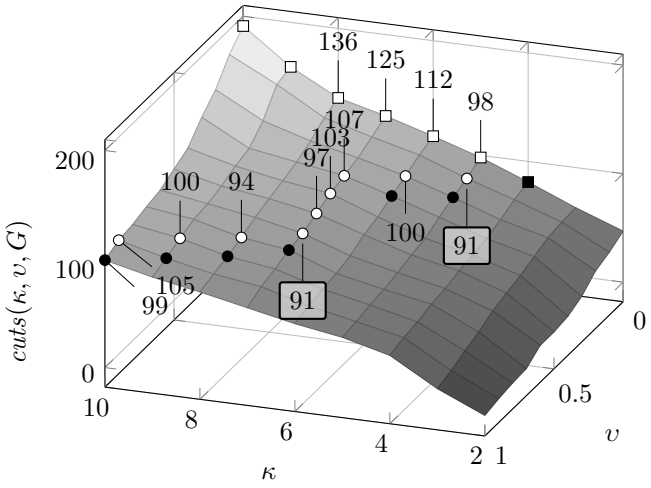


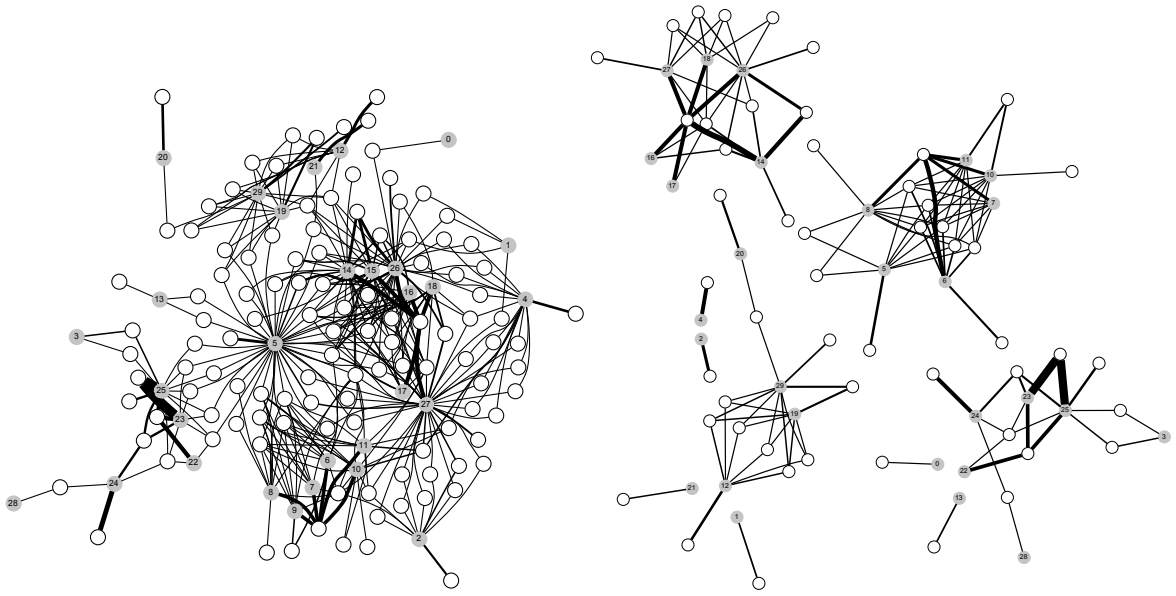
Figure 4.4: The PCS heuristic applied to SUPM 7'.

Table 4.3: Result of the PCS heuristic applied to SUPM 7'.

SUPM	$\|J\|$	$\|I\|$	ρ [%]	$\frac{\sum_{i\in I}\|\mathcal{V}(P_i)\|}{\sum_{i\in I}\|\mathcal{H}(P_i)\|}$	$\|\mathcal{V}(P)\|$	$\|\mathcal{H}(P)\|$	$\overline{icd}$	`minksum`	`cddlib`
7'.1'	12	6	47.2	40	1,670	55	386.7	14s	693s
7'.2'	14	6	60.7	57	1,152	30	374.9	19s	113s
7'.3'.a	1	1	100.0	2	2	2	1.0	0s	0s
7'.3'.b	1	1	100.0	2	2	2	1.0	0s	0s
7'.4'.a	11	4	56.8	29	294	29	91.5	2s	2s
7'.4'.b	1	1	100.0	2	2	2	1.0	0s	0s
7'.4'.c	1	1	100.0	2	2	2	1.0	0s	0s
7'.5'.a	9	6	37.0	26	469	32	145.2	2s	5s
7'.5'.b	1	1	100.0	2	2	2	1.0	0s	0s
7'.5'.c	1	1	100.0	2	2	2	1.0	0s	0s

function value of partitions that define solvable constraint generation problems appears twice: $cuts(5, 0.1, G) = cuts(7, 0.6, G) = 91$. The capacity constraints returned by the PCS heuristic are based on the $(5, 1.1)$-balanced partition. Table 4.3 describes these constraints and the underlying SUPMs, which are the result of partitioning the graph of SUPM 7' into five blocks.

Two blocks define a single SUPM each, i.e., 7'.1' and 7'.2'. One block defines two SUPMs, i.e., 7'.3'.a and 7'.3'.b. And, two blocks define three SUPMs each. The columns $|J|$ and $|I|$ in Table 4.3 do not sum up to $|J|= 116$ and $|I|= 30$ of SUPM 7' in Table 4.2 because uniform machines and uniform products have been aggregated once again in each of the SUPMs in Table 4.3. The SUPMs in Table 4.3 define in total 158 capacity constraints including 52 non-negativity constraints. The capacity constraints are not accurate. 91 out of 868 product-machine assignments that are feasible according to the original SUPM 7, which is around 10.5 %, are not captured by the capacity constraints because of edge cuts. Figure 4.5 illustrates this effect. Figure 4.5a shows the graph G of SUPM 7' and Figure 4.5b shows the



(a) SUPM 7' (b) $(5, 1.1)$-balanced partition of SUPM 7'

Figure 4.5: The effect of graph partitioning.

$(5, 1.1)$-balanced partition of G after cutting the edges between blocks and aggregating the uniform elements within each block. Every connected component in Figure 4.5b corresponds to one of the SUPMs in Table 4.3. Gray and white nodes represent machines and products, respectively. Edge thickness is proportional to the square root of edge weight.

We are interested in the accuracy of the capacity constraints that are provided by the PCS heuristic and we want to compare it with the accuracy of the DPF proposed by Leachman and Carmon (1992). Therefore, we define measures for objective function value accuracy and production plan feasibility. For each type of capacity constraints, we compute these measures based on the production plans that maximize 1,000 randomly generated objective functions.

For a SUPM that has n products, a set of $q = 1, 2, \ldots, 1{,}000$ n-dimensional vectors $\boldsymbol{u}_q$ is generated with coordinates that are uniformly distributed between 1 and 999. The vector $\boldsymbol{u}_q$ defines the coefficients of the qth objective function. $\boldsymbol{x}_q$ represents the qth n-dimensional vector of decision variables, i.e., of production quantities. The following formulation represents a single-period production planning problem. We solve it for each q subject to different types of capacity constraints that are generated for the same SUPM.

$$z_q = \max \boldsymbol{u}_q \boldsymbol{x}_q \tag{4.30}$$

s.t.

$$\text{capacity constraints} \tag{4.31}$$

First, (4.30) is maximized subject to the set of high-dimensional, accurate capacity constraints according to the SSF (see equations (4.2) and (4.3)). The optimal objective function value and the optimal production plan of the qth problem are denoted as z_q^{SSF} and $\boldsymbol{x}_q^{\mathrm{SSF}}$. Second, we use the low-dimensional capacity constraints provided by the PCS heuristic. The optimal objective function value and the optimal production plan are denoted as z_q^{PCS} and $\boldsymbol{x}_q^{\mathrm{PCS}}$. Third, (4.31) is replaced with the low-dimensional constraints provided by the DPF (see inequality (4.6)). We denote the optimal objective function value and the optimal production plan as z_q^{DPF} and $\boldsymbol{x}_q^{\mathrm{DPF}}$.

The DPF requires uniformity, i.e., the equation $p_{ij} = \frac{p_j}{\mu_i}$ must hold for every machine $i \in I$ and every product $j \in J(i)$. Following the suggestion of Leachman and Carmon (1992), we create uniformity by averaging. That means every processing time $p_{ij} \in \mathbb{R}_{>0}$ is replaced with $\frac{\overline{p}_{i\cdot}\overline{p}_{\cdot j}}{\overline{p}_{\cdot\cdot}}$, where $\overline{p}_{i\cdot}$, $\overline{p}_{\cdot j}$, and $\overline{p}_{\cdot\cdot}$ represent the column average, the row average, and the matrix average of the processing times. If averaging is applied to the non-uniform processing times in (4.9), for example, the result is:

$$\begin{array}{cccc|cccc} p_{11} & p_{12} & p_{13} & c_1 & 1 & 2 & \infty & 20 \\ p_{21} & p_{22} & p_{23} & c_2 = & 2 & 4 & 6 & 35 \\ p_{31} & p_{32} & p_{33} & c_3 & \infty & 4 & 12 & 124 \end{array} \xrightarrow{\text{Averaging}} \begin{array}{cccc} 0.51 & 1.13 & \infty & 20 \\ 1.35 & 3.01 & 8.13 & 35 \\ \infty & 6.02 & 16.26 & 124 \end{array} \tag{4.32}$$

Using the $q = 1, 2, \ldots, 1{,}000$ solutions of the production planning problem in (4.30) to (4.31), we define the objective function value inaccuracy (*OFI*) as the mean absolute percentage deviation from the optimal objective function value subject to the accurate SSF:

$$OFI^{\mathrm{PCS}} = \frac{\sum_{q=1}^{1000} \left| \frac{z_q^{\mathrm{PCS}} - z_q^{\mathrm{SSF}}}{z_q^{\mathrm{SSF}}} \right|}{1000} \tag{4.33}$$

$$OFI^{\mathrm{DPF}} = \frac{\sum_{q=1}^{1000} \left| \frac{z_q^{\mathrm{DPF}} - z_q^{\mathrm{SSF}}}{z_q^{\mathrm{SSF}}} \right|}{1000} \tag{4.34}$$

The production plan feasibility (*Feas*) is defined as the percentage of production plans that are feasible subject to the SSF capacity constraints. In order to determine if a production plan, e.g., $\boldsymbol{x}_q^{\mathrm{PCS}}$, is feasible subject to the SSF constraints, a ray shooting problem is initialized. The following maximization problem shoots a ray starting from the origin in the direction of $\boldsymbol{x}_q^{\mathrm{PCS}}$ until it hits the boundary of the polytope that is defined by the SSF constraints. Note that in this formulation, $\boldsymbol{x}_q^{\mathrm{PCS}}$ is fixed as a vector of parameters and $\boldsymbol{x}$ is a vector of decision variables.

$$\begin{aligned} &\max \lambda_q^{\mathrm{PCS}} \\ \text{s.t.} \\ &\boldsymbol{x} = \lambda_q^{\mathrm{PCS}} \boldsymbol{x}_q^{\mathrm{PCS}} \end{aligned}$$

Table 4.4: Results of the PCS heuristic applied to the SUPMs 3', 4', 6', and 7'.

SUPM	κ	υ	$cuts(\kappa,\upsilon,G)^{\text{rel}}$ [%]	OFI^{PCS} [%]	OFI^{DPF} [%]	$Feas^{\text{PCS}}$ [%]	$Feas^{\text{DPF}}$ [%]	`minksum`	`cddlib`
3'	3	0.4	13.3	0.5	32.5	100.0	0.0	46s	1,827s
4'	2	0.5	4.0	2.1	140.0	100.0	0.0	12s	324s
6'	4	0.8	17.0	10.1	6.0	100.0	0.0	93s	3,145s
7'	5	0.1	10.5	8.6	56.6	100.0	0.0	37s	813s

step-separated formulation of capacities

The production plan $\boldsymbol{x}_q^{\text{PCS}}$ is feasible subject to the SSF if and only if the optimal value of the scalar λ_q^{PCS} is greater or equal to one. The ratio of feasible production plans $Feas^{\text{PCS}}$ is defined as

$$Feas^{\text{PCS}} = \frac{\sum_{q=1}^{1000} 1_{Feas}(\lambda_q^{\text{PCS}})}{1000}, \tag{4.35}$$

with indicator function

$$1_{Feas}(\lambda_q^{\text{PCS}}) = \begin{cases} 1 & \text{if } \lambda_q^{\text{PCS}} \geq 1 \\ 0 & \text{otherwise.} \end{cases} \tag{4.36}$$

The ratio of feasible production plans provided by the DPF, $Feas^{\text{DPF}}$, is calculated analogously but based on the maximizer $\boldsymbol{x}_q^{\text{DPF}}$.

We apply the PCS heuristic to the SUPMs 3', 4', 6', and 7' and present the results in Table 4.4. The columns κ, υ, and $cuts(\kappa,\upsilon,G)^{\text{rel}}$ describe the $(\kappa, 1+\upsilon)$-balanced partitions that are returned by the PCS heuristic. $cuts(\kappa,\upsilon,G)^{\text{rel}}$ stands for the relative number of cut edges, i.e., $cuts(\kappa,\upsilon,G)$ divided by the total number of edges in graph G of the non-aggregated SUPM. Between 4.0 % and 17.0 % of the feasible product-machine assignments are cut and are therefore not captured by the capacity constraints. The OFI^{PCS} ranges between 0.5 % and 10.1 %. The capacity constraints that are generated by the PCS heuristic tend to be more accurate than the DPF. The average gap between OFI^{PCS} and OFI^{DPF} is 53.4 percentage points. The relative number of edge cuts $cuts(\kappa,\upsilon,G)^{\text{rel}}$ and OFI^{PCS} are not perfectly rank correlated. Note that the effect of cutting an edge can vary depending on the production capacity that is cut. As expected, all of the 1,000 production plans that are optimized subject to the capacity constraints of the PCS heuristic are feasible. It is remarkable that none of the

production plans that are generated subject to the DPF of capacities are feasible. The two rightmost columns of Table 4.4 provide the total computation times of the PCS heuristic on the partition that is specified by κ and υ. The sum of `minksum` and `cddlib` is below the time limit of $T = 3{,}600$ seconds.

4.6.2 Experiment with randomly generated data

The results of the experiment in the previous section indicate that the PCS heuristic provides more accurate capacity constraints than Leachman and Carmon's DPF. As the sample size is too small to draw firm conclusions, we conduct a comprehensive full factorial experiment, in which problem instances, i.e., SUPMs, are randomly generated such that the diversity of constraint generation problems observed in the field is largely captured.

A base problem instance counts 500 products and 12 parallel machines and satisfies the uniformity condition (see Table C.1 in Appendix C.2). All machines are assumed to have a capacity of 10,000 machine hours. In addition to the constraint generation method, three factors are examined: the density of the processing time matrix ρ (25 % and 50 %), the range of processing times ($\{1, 5, 10, \infty\}$ and $\{1, 100, 200, \infty\}$), and the non-uniformity of the processing time matrix (four levels). The levels of ρ and the ranges of processing times are motivated by the SUPMs introduced in Table 4.1. Three levels non-uniformity are created incrementally by swapping random pairs of 25×3-submatrices of the 500×12-processing time matrix (Table C.2 in Appendix C.2 shows an example). The intention is to create the structure of block-wise uniformity, which we have found in the processing time matrices of the SUPMs from semiconductor manufacturing.

The 2^2-×-4 full factorial experiment is replicated 30 times, resulting in 480 separate constraint generation problems. Proper variation and coordination of random number streams are implemented. The timeout of `KaFFPaE` and T of the PCS heuristic are set to 60 seconds. Where possible, uniform elements are aggregated and independent SUPMs are treated separately. The results are presented in Table 4.5. We have verified that the assumptions of error term normality and equality of variance are not violated and conducted an analysis of variance (ANOVA) for OFI^{PCS} (see Table C.3 in Appendix C.2). All main effects and all two-way interaction effects are significant at the 95 % confidence level. A paired t-test with two-sided alternative hypothesis comparing OFI^{PCS} with OFI^{DPF} rejects the null hypothesis of equality for every treatment combination at a significance level of 5 % or less (see p-values in Table 4.5). Here, a normal probability plot of the residuals of OFI^{PCS} has revealed seven outliers that are much greater than the replication means. Since these outliers do not change the direction of the results but affect the normality assumption of the t-test, we have replaced them with the replication means.

Table 4.5: Results of the factorial experiment (replication means).

Treatment combination	ρ [%]	Range of p_{ij}	Number of swaps	$\lvert J\rvert$	$\lvert I\rvert$	$f(cons(1,v,G,T))$ [%]	κ	v	$cuts(\kappa,v,G)^{\text{rel}}$ [%]	OFI^{PCS} [%]	OFI^{DPF} [%]	$Feas^{\text{PCS}}$ [%]	$Feas^{\text{DPF}}$ [%]
1	25	$\{1,5,10,\infty\}$	0	4.0	4.0	100.0	1.0	0.00	0.0	0.0**	0.0	100.0	100.0
2	25	$\{1,5,10,\infty\}$	3	11.0	9.7	36.7	1.7	0.24	1.2	2.4**	11.6	100.0	15.8
3	25	$\{1,5,10,\infty\}$	6	18.7	11.3	0.0	2.4	0.33	4.9	4.3**	19.3	100.0	9.5
4	25	$\{1,5,10,\infty\}$	9	27.6	11.7	0.0	3.5	0.48	10.0	8.2*	17.9	100.0	2.1
5	25	$\{1,100,200,\infty\}$	0	4.0	4.0	100.0	1.0	0.00	0.0	0.0**	0.0	100.0	100.0
6	25	$\{1,100,200,\infty\}$	3	11.0	9.7	36.7	1.7	0.24	1.2	2.8**	23.9	100.0	22.3
7	25	$\{1,100,200,\infty\}$	6	18.7	11.3	0.0	2.5	0.34	4.9	5.4**	39.0[1]	100.0	22.2
8	25	$\{1,100,200,\infty\}$	9	27.6	11.7	0.0	3.4	0.44	10.1	10.3**	37.3[2]	100.0	9.9
9	50	$\{1,5,10,\infty\}$	0	2.0	2.0	100.0	1.0	0.00	0.0	0.0**	0.0	100.0	100.0
10	50	$\{1,5,10,\infty\}$	3	7.8	9.1	46.7	1.5	0.23	1.8	3.5**	14.0	100.0	52.6
11	50	$\{1,5,10,\infty\}$	6	15.6	11.5	3.3	2.1	0.14	7.6	11.5**	26.2	100.0	70.5
12	50	$\{1,5,10,\infty\}$	9	25.9	11.9	0.0	3.3	0.47	12.3	15.0**	32.9	100.0	70.3
13	50	$\{1,100,200,\infty\}$	0	2.0	2.0	100.0	1.0	0.00	0.0	0.0**	0.0	100.0	100.0
14	50	$\{1,100,200,\infty\}$	3	7.8	9.1	46.7	1.5	0.23	1.8	4.6**	34.7	100.0	86.2
15	50	$\{1,100,200,\infty\}$	6	15.6	11.5	3.3	2.1	0.13	7.6	15.0**	53.7[1]	100.0	86.9
16	50	$\{1,100,200,\infty\}$	9	25.9	11.9	0.0	3.3	0.47	12.3	19.2**	69.8	100.0	85.6

* significant at $p < 0.05$; ** significant at $p < 0.0001$

[1] one outlier replaced with replication mean; [2] four outliers replaced with replication mean

The results in Table 4.5 show that the PCS heuristic outperforms Leachman and Carmon's DPF in both objective function value accuracy and production plan feasibility. OFI^{DPF} is between 2.2 and 8.6 times greater than OFI^{PCS} in treatment combinations with non-uniform problem instances. The fraction of feasible production plans of the DPF, $Feas^{\mathrm{DPF}}$, can deteriorate to 2.1 %, whereas the capacity constraints of the PCS heuristic define feasible production plans. The column $f(cons(1, \upsilon, G, T))$ presents the ratio of constraint generation problems that are solved by the PCS heuristic within the time limit and without partitioning, i.e., with $\kappa = 1$. The PCS heuristic is in particular effective with thin processing time matrices ($\rho = 25\,\%$) and non-uniformity caused by 3 to 6 submatrix swaps. In the treatment combinations 3 and 7, none of the problem instances can be solved without cutting edges ($f(cons(1, \upsilon, G, T)) = 0\,\%$) and 4.9 % of the edges are cut. The objective function value inaccuracy of the PCS heuristic, OFI^{PCS}, does not exceed 5.4 % in these treatment combinations. The objective function value inaccuracy of the DPF, OFI^{DPF}, is in contrast at least 19.3 % and only up to 22.2 % of the generated production plans are feasible. Note the risk that the accuracy of the DPF can deteriorate even further as the averaging of processing times can produce outliers.

4.7 Conclusion

We propose a capacity constraint generation procedure and a partition-based constraint search heuristic for the generation of low-dimensional capacity constraints of unrelated parallel machines. Because of its computational complexity, the capacity constraint generation procedure provides accurate capacity constraints up to a certain problem size. We propose the aggregation of uniform elements to reduce problem size, which expands the range of application of the procedure. The PCS heuristic trades off constraint accuracy against computation time and can offer a solution for otherwise intractable constraint generation problems.

Our numerical results show that the number of generated capacity constraints is acceptable for practical applications and computation times can be in the range of hours. As capacity constraints are recalculated only when changes to the production system or the product range occur, there is however sufficient time for using the proposed methods in practice. The accuracy of the capacity constraints generated by the PCS heuristic is superior to the direct product mix formulation of capacities. The objective function value deviates less from the true optimum and the resulting production plans are feasible. The PCS heuristic performs in particular well with thin processing time matrices, e.g., when the feasible product sets are on average half the size or smaller than the complete set of products that can be processed by the parallel machines. The experiments with both field data and randomly generated data suggest that the objective function value deviates here from the true optimum by around 5 %.

An opportunity for future research is to enhance the graph partitioning, which is an input to the PCS heuristic. The number of weighted edge cuts used by us only approximates the operational relevance of product-machine assignments. Defining edge weights as a function of the size and the need for the machine capacity or defining certain edges as unbreakable could improve the quality of the capacity constraints. Another future development could be to combine the PCS heuristic with averaging to increase uniformity punctually and thus avoid partitioning. This would make the aggregation of uniform elements more effective at the cost of constraint accuracy. Clearly, also future research about algorithms that solve the underlying geometric problems in shorter time will be of great interest.

Chapter 5

Summary

5.1 Summary of findings

In this section, we summarize the conclusions drawn in the previous chapters and relate our findings to the research questions formulated in Section 1.3.

Research question 1. *In an engineer-to-order manufacturing environment with a rapidly evolving product portfolio and uncertain future demand, what is the optimal product platform design?*

In Chapter 2, we explore the opportunity of designing intermediate products as product platforms for both present and expected future customer orders. Over-designing a product platform in the present can prevent future customer orders from triggering a full product design cycle, which is usually the process in engineer-to-order companies. While a product platform can reduce product design costs, it can also increase manufacturing costs since it has to meet the feature value requirements of the most demanding of the served product variants in every feature.

We propose a two-stage stochastic programming formulation with recourse of the tactical product platform design problem. The first stage models product platform design decisions in the present. The design of additional product platforms in the future, which become necessary if first-stage designs do not cover every possible future customer order, is modeled as a second-stage recourse action. The formulation balances platform design costs with manufacturing costs and takes both customer orders on hand and expectations about future customer orders explicitly into account.

We apply the model to a case from the silicon wafer manufacturing industry. Single-crystal rods of pure silicon (ingots), which are an intermediate product in silicon wafer manufacturing, are considered as potential product platforms. Final product variants, i.e., silicon wafers, are manufactured based on an ingot in downstream processing steps, which include ingot slicing and various surface treatments. We compare the performance of the proposed model with the performance of its deterministic counterpart in a designed experiment that is inspired by problem instances found in silicon wafer manufacturing.

Our analyses show that the tactical over-design of product platforms can generate a significant saving compared to single-stage deterministic platform design. By taking the stochasticity of future customer orders into account, product platform design can anticipate future requirements accordingly, and thus avoid the costly design of additional platforms in the future. Experiments show that this is in particular effective in situation with high design-to-manufacturing cost ratios and with small innovation steps. In addition to the saving due to a reduced design workload, the introduction of product platforms also enables a reduction of variability in the manufacturing system as well as a postponement of the customer order decoupling point and hence a reduction of safety stock levels due to risk-pooling.

Research question 2. *Which mid-term production planning method meets best the conflicting objectives in wafer fabrication and how does it interact with production control?*

In Chapter 3, we study the mid-term production planning problem in wafer fabrication. Despite the uncertainty of output targets and machine availability, mid-term production planning guides the flow of material through the complex production system of a wafer fab and ensures high performance in service level, throughput, and cycle time. It interacts with production control by influencing scheduling decisions at bottleneck work centers, and thus ensures that the assignment of jobs to machines is aligned with the overall fab objectives.

We propose a novel CT-oriented mid-term production planning method as an alternative to popular WIP-oriented mid-term planning. While WIP-oriented planning defines throughput and WIP level targets for bottleneck work centers, CT-oriented planning defines release quantities and cycle time targets. On the shop floor, these targets are consumed by scheduling methods that prioritize the production lots that are most behind their target for the respective work center. Due to this integration with scheduling, CT-oriented planning manages the cycle time and thus the completion date of lots directly, whereas in WIP-oriented planning, cycle time is an indirect planning result.

We apply both planning methods to a reference model of a wafer fab. Comprehensive analyses are conducted using a rolling horizon framework that simulates the iterative process of production planning and production control over several weeks. Both planning methods are tested under different levels of uncertainty and we investigate their ability to meet spikes in fab output targets at short notice.

The results show that CT-oriented mid-term production planning outperforms WIP-oriented planning. For a given throughput level, CT-oriented planning delivers higher service levels and shorter cycle times. CT-oriented planning effectively accelerates lots by tightening target cycle times and thus maintains a high service level even when output targets and machine capacity are uncertain. The negative effect of inappropriate priority changes is limited since opportunistic non-delay scheduling keeps throughput high. WIP-oriented planning, in contrast, suffers from targets that stringently control the throughput of every bottleneck and, which result in elevated WIP levels and a lack of responsiveness to demand changes. Another advantage of CT-oriented planning is that it involves much less communication with production control as it only shares target cycle time changes, while WIP-oriented planning has to provide a full set of targets for all bottleneck work centers every time a new plan is generated. This makes CT-oriented mid-term production plans easier to comprehend and adjust by human planners and operators.

Research question 3. *How can we accurately capture the capacity of parallel machines such that the total production rates of product types are the only variables of the model?*

Company-wide production planning generates a master production schedule, which defines production targets for all facilities of the production network. While the accurate representation of capacity in the production planning model is a prerequisite for generating feasible and efficient plans, detailed capacity allocation decisions are beyond the scope of a typical master production schedule. In Chapter 4, we therefore explore methods for the generation of low-dimensional capacity constraints, which accurately model the capacity of parallel machines and only rely on variables that model the total production rates of product types.

We propose a novel constraint generation procedure, which generates the irredundant set of exact, low-dimensional, and linear capacity constraints for unrelated parallel machines. This procedure is based on standard algorithms developed for convex polytopes. Since the computational complexity of the procedure is critical, we exploit the uniformity of both machines and products to reduce problem size. Further, we propose a heuristic constraint generation procedure based on graph partitioning, which solves even bigger problem instances by trading off constraint accuracy against computation time.

The proposed methods are applied to problem instances from semiconductor manufacturing. In addition, a comprehensive designed experiment is conducted with randomly generated problem instances that mimic real-world settings. Both the exact and the heuristic constraint generation method are evaluated based on the accuracy and the feasibility of the resulting capacity constraints. The results show that exact low-dimensional capacity constraints can be calculated for realistic problem instances up to a certain size. For larger problems, our heuristic constraint generation procedure outperforms an existing heuristic in both accuracy and feasibility. This holds especially for problem instances that are typical for the semiconductor industry.

To conclude, this dissertation provides quantitative approaches to decision making problems that are prevalent in the planning and control of semiconductor supply chains, i.e., supply chains that are strongly affected by a fierce international competition, technological complexity, and rapid innovation. The intense competition in the semiconductor industry incentivizes manufacturers to improve operational efficiency, which is also the objective of tactical product platform design, mid-term production planning, and accurate capacity modeling. The high level of technological complexity justifies the relevance of the studied decision problems since technological sophistication translates into expensive production systems, whose utilization can be improved through better planning and control. At last, the high innovation rate of the semiconductor industry adds an extra level of difficulty as it requires planning and control to manage a source of variability that is characteristic for semiconductors. Although the presented approaches are tailored to industry-specific cases, the proposed optimization and simulation models are formulated generically enough to be applied to any other setting with similar requirements. It is demonstrated how mathematical modeling and optimization can be used by planning and control to improve key metrics, such as cost per unit, customer lead time, and service level, through better decision making. Numerical experiments and simulations are proven to be powerful tools for validation and providing confidence in the benefit of the proposed changes in planning and control. Eventually, quantitative approaches offer a path to the complete automation of decision making processes, which can lead to further enhancements of decision quality and cost.

5.2 Future research opportunities

Future research opportunities have been outlined in the concluding sections of the previous chapters. In this section, we expand on this more generally.

This dissertation focuses on the modeling, optimization, and simulation of planning and control problems in semiconductor supply chains. The values of model parameters are either derived as statistics from historical observations or based on results published in related studies. In reality, the proper parametrization of a model is difficult because conditions are usually less stable than in controlled experiments. In addition, it is desirable that model parametrization is an automated process, requiring little human interaction. The ever increasing volume of data that is collected by production monitoring and the rise of large-scale data analysis enabled by an abundance of computing power suggest future research opportunities in the accurate estimation of important production planning parameters. For example, CT-oriented mid-term production relies on accurate distributional assumptions of segment cycle times, which are affected by several factors, such as product mix, priority mix, and fab utilization. While these distributional assumptions can be specified based on simulation results, actual production data could help to predict cycle time distributions more accurately and thus improve the quality of plans.

Another research opportunity lies in the application of pricing and revenue management techniques in the semiconductor industry. Speed matters and despite many types of semiconductors being considered as commodities, there might exist niche markets, such as prototype development, in which customers are willing to pay a premium for reduced lead times. This will not only affect investment decisions but can also offer opportunities for revenue maximization. For example, in production networks with an option to assign workload to contractors, capacity allocation decisions have to be made such that revenue is maximized. Further, production planning can provide valuable information about the marginal cost of prioritizing production lots, which can be used in pricing decisions.

The semiconductor industry is predicted to grow as both a supplier and a customer of IoT and data analytics technology. The collection of big data through embedded sensors, its transmission to the cloud, and data analysis for corrective actions in real-time enable new business models that are based on servitization. For example, semiconductor equipment manufacturers could servitize expensive process steps in wafer fabrication to increase equipment uptime, improve fab efficiency, and reduce the overall cost of ownership. As a supplier of IoT components, the semiconductor industry could servitize the provision of compute and storage hardware in data centers. This could reduce demand uncertainty and improve fab utilization and it also offers new research opportunities in, e.g., predictive and preventive maintenance.

Appendix A

A.1 Multifactor ANOVA

The ANOVA Tables A.1, A.2, and A.3 decompose the variability of the response variables η, $relVTSM$, and $relVSS$, respectively, into contributions due to various factors. The contribution of every factor is measured having removed the effects of all other factors. The P-values test the statistical significance of each of the factors. All factors, except for demand uncertainty σ^{D}, have a statistically significant main effect on the three response variables at the 99.0 % confidence level as the corresponding P-values are less than 0.01.

Table A.1: ANOVA for the number of second-stage orders that are served with the first-stage platform design (η).

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A: Fix costs FC	630.71	1	630.71	1,318.07	0.0000
B: Demand growth μ^{D}	962.35	1	962.35	2,011.12	0.0000
C: Demand uncertainty σ^{D}	0.00	1	0.00	0.01	0.9193
D: Feature value growth μ^{FM}	312.05	1	312.05	652.12	0.0000
E: Feature value variability σ^{FM}	25.84	1	25.84	54.00	0.0000
F: Feature value uncertainty σ^{FV}	768.30	1	768.30	1,605.61	0.0000
INTERACTIONS					
AB	14.12	1	14.12	29.51	0.0000
AC	0.07	1	0.07	0.14	0.7109
AD	40.01	1	40.01	83.61	0.0000
AE	3.21	1	3.21	6.70	0.0096
AF	4.13	1	4.13	8.63	0.0033
BC	0.05	1	0.05	0.09	0.7580
BD	20.19	1	20.19	42.19	0.0000
BE	1.16	1	1.16	2.43	0.1187
BF	3.30	1	3.30	6.89	0.0087
CD	0.01	1	0.01	0.01	0.9056
CE	0.02	1	0.02	0.05	0.8243
CF	0.05	1	0.05	0.09	0.7580
DE	3.58	1	3.58	7.49	0.0062
DF	9.74	1	9.74	20.36	0.0000
EF	2.53	1	2.53	5.30	0.0214
RESIDUAL	1,367.59	2,858	0.48		
TOTAL (CORRECTED)	4,169.01	2,879			

Table A.2: ANOVA for the relative value of two-stage modelling ($relVTSM$).

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A: Fix costs FC	1.1971	1	1.1971	3,866.86	0.0000
B: Demand growth μ^{D}	1.5707	1	1.5707	5,073.83	0.0000
C: Demand uncertainty σ^{D}	0.0000	1	0.0000	0.02	0.8853
D: Feature value growth μ^{FM}	0.4327	1	0.4327	1,397.69	0.0000
E: Feature value variability σ^{FM}	0.0854	1	0.0854	275.93	0.0000
F: Feature value uncertainty σ^{FV}	0.8792	1	0.8792	2,839.94	0.0000
INTERACTIONS					
AB	0.5028	1	0.5028	1,624.26	0.0000
AC	0.0000	1	0.0000	0.00	0.9633
AD	0.0581	1	0.0581	187.66	0.0000
AE	0.0182	1	0.0182	58.75	0.0000
AF	0.1449	1	0.1449	468.10	0.0000
BC	0.0000	1	0.0000	0.02	0.9022
BD	0.1166	1	0.1166	376.68	0.0000
BE	0.0305	1	0.0305	98.66	0.0000
BF	0.2367	1	0.2367	764.70	0.0000
CD	0.0000	1	0.0000	0.00	0.9692
CE	0.0000	1	0.0000	0.00	0.9721
CF	0.0000	1	0.0000	0.00	0.9933
DE	0.0183	1	0.0183	59.27	0.0000
DF	0.0810	1	0.0810	261.74	0.0000
EF	0.0279	1	0.0279	90.05	0.0000
RESIDUAL	0.8848	2,858	0.0003		
TOTAL (CORRECTED)	6.2851	2,879			

Table A.3: ANOVA for the the relative value of stochastic solution (*relVSS*).

Source	Sum of Squares	Df	Mean Square	*F*-Ratio	*P*-Value
MAIN EFFECTS					
A: Fix costs FC	1.1312	1	1.1312	3,752.75	0.0000
B: Demand growth μ^{D}	2.0710	1	2.0710	6,870.77	0.0000
C: Demand uncertainty σ^{D}	0.0000	1	0.0000	0.02	0.8834
D: Feature value growth μ^{FM}	0.0415	1	0.0415	137.72	0.0000
E: Feature value variability σ^{FM}	0.0224	1	0.0224	74.29	0.0000
F: Feature value uncertainty σ^{FV}	0.8580	1	0.8580	2,846.53	0.0000
INTERACTIONS					
AB	0.3590	1	0.3590	1,191.10	0.0000
AC	0.0000	1	0.0000	0.00	0.9613
AD	0.0573	1	0.0573	190.09	0.0000
AE	0.0018	1	0.0018	5.89	0.0153
AF	0.1367	1	0.1367	453.61	0.0000
BC	0.0000	1	0.0000	0.02	0.9020
BD	0.0510	1	0.0510	169.26	0.0000
BE	0.0019	1	0.0019	6.14	0.0132
BF	0.2249	1	0.2249	746.27	0.0000
CD	0.0000	1	0.0000	0.00	0.9819
CE	0.0000	1	0.0000	0.00	0.9972
CF	0.0000	1	0.0000	0.00	0.9954
DE	0.0074	1	0.0074	24.63	0.0000
DF	0.0959	1	0.0959	318.21	0.0000
EF	0.0280	1	0.0280	92.84	0.0000
RESIDUAL	0.8615	2,858	0.0003		
TOTAL (CORRECTED)	5.9496	2,879			

A.2 Mean and confidence interval of total costs by treatment combination

The sample mean $\overline{\zeta}$ of optimal total costs and the relative half-width of the approximate 95.0 % confidence interval for the true optimal total costs ζ^* are provided for every treatment combination in Table A.4. According to the central limit theorem, the approximate half-width is the 0.975 critical value of the standard normal distribution $z_{0.975}$ multiplied with the unbiased estimator of the standard deviation of the sample mean $\sqrt{S^2/45}$. S^2 represents the sample variance of ζ and 45 is the number of replications per treatment combination. The central limit theorem can be applied because ζ is i.i.d. for every treatment combination. The relative half-width is defined as the half-width divided by $\overline{\zeta}$. The grand mean of the relative half-width is 1.20 %, which means that the interval $\overline{\zeta} \pm 0.012\overline{\zeta}$ covers the true optimal total costs with 95.0 % probability.

Table A.4: Mean total costs and the relative half-width of the 95.0 % confidence interval for the true optimal total costs ζ^* by treatment combination.

Treatment	FC	μ^{D}	σ^{D}	μ^{FM}	σ^{FM}	σ^{FV}	$\overline{\zeta}$ [\$]	$\frac{z_{0.975}\sqrt{\frac{S^2}{45}}}{\overline{\zeta}}$ [%]
1	70,000	1,000	100	16	1.6	1.6	329,274.3	1.11
2	70,000	1,000	100	16	1.6	3.2	360,713.9	0.83
3	70,000	1,000	100	16	3.2	1.6	342,543.7	1.93
4	70,000	1,000	100	16	3.2	3.2	366,147.0	1.36
5	70,000	1,000	100	20	1.6	1.6	380,111.2	0.89
6	70,000	1,000	100	20	1.6	3.2	398,008.8	0.54
7	70,000	1,000	100	20	3.2	1.6	389,793.3	1.41
8	70,000	1,000	100	20	3.2	3.2	402,038.6	1.07
9	70,000	1,000	250	16	1.6	1.6	329,062.5	1.15
10	70,000	1,000	250	16	1.6	3.2	360,461.2	0.90
11	70,000	1,000	250	16	3.2	1.6	342,328.1	1.95
12	70,000	1,000	250	16	3.2	3.2	365,864.7	1.38
13	70,000	1,000	250	20	1.6	1.6	379,849.2	0.96
14	70,000	1,000	250	20	1.6	3.2	397,845.6	0.66
15	70,000	1,000	250	20	3.2	1.6	389,464.8	1.44
16	70,000	1,000	250	20	3.2	3.2	401,780.7	1.11
17	70,000	2,000	100	16	1.6	1.6	523,826.4	1.03
18	70,000	2,000	100	16	1.6	3.2	549,387.0	0.81
19	70,000	2,000	100	16	3.2	1.6	535,739.3	1.70
20	70,000	2,000	100	16	3.2	3.2	555,628.3	1.50
21	70,000	2,000	100	20	1.6	1.6	597,878.6	0.72
22	70,000	2,000	100	20	1.6	3.2	615,271.0	0.68
23	70,000	2,000	100	20	3.2	1.6	606,691.5	1.46
24	70,000	2,000	100	20	3.2	3.2	622,722.0	1.37
25	70,000	2,000	250	16	1.6	1.6	523,611.3	1.05
26	70,000	2,000	250	16	1.6	3.2	549,192.7	0.84
27	70,000	2,000	250	16	3.2	1.6	535,542.8	1.70
28	70,000	2,000	250	16	3.2	3.2	555,380.9	1.50
29	70,000	2,000	250	20	1.6	1.6	597,710.9	0.76
30	70,000	2,000	250	20	1.6	3.2	615,092.0	0.71
31	70,000	2,000	250	20	3.2	1.6	606,441.2	1.47
32	70,000	2,000	250	20	3.2	3.2	622,457.8	1.38
33	110,000	1,000	100	16	1.6	1.6	372,033.5	1.01
34	110,000	1,000	100	16	1.6	3.2	409,791.3	0.87
35	110,000	1,000	100	16	3.2	1.6	387,118.0	1.90
36	110,000	1,000	100	16	3.2	3.2	421,019.5	1.67
37	110,000	1,000	100	20	1.6	1.6	422,869.6	0.83
38	110,000	1,000	100	20	1.6	3.2	458,887.2	0.75
39	110,000	1,000	100	20	3.2	1.6	437,540.4	1.65
40	110,000	1,000	100	20	3.2	3.2	469,279.0	1.35
41	110,000	1,000	250	16	1.6	1.6	371,809.7	1.05
42	110,000	1,000	250	16	1.6	3.2	409,524.2	0.94
43	110,000	1,000	250	16	3.2	1.6	386,873.1	1.91
44	110,000	1,000	250	16	3.2	3.2	420,750.8	1.68
45	110,000	1,000	250	20	1.6	1.6	422,591.9	0.89
46	110,000	1,000	250	20	1.6	3.2	458,597.8	0.84
47	110,000	1,000	250	20	3.2	1.6	437,255.8	1.67
48	110,000	1,000	250	20	3.2	3.2	468,938.4	1.37
49	110,000	2,000	100	16	1.6	1.6	574,449.9	1.13
50	110,000	2,000	100	16	1.6	3.2	625,719.6	0.82
51	110,000	2,000	100	16	3.2	1.6	596,761.5	1.93
52	110,000	2,000	100	16	3.2	3.2	633,797.0	1.44
53	110,000	2,000	100	20	1.6	1.6	666,004.6	0.90
54	110,000	2,000	100	20	1.6	3.2	695,312.4	0.60
55	110,000	2,000	100	20	3.2	1.6	682,167.5	1.48
56	110,000	2,000	100	20	3.2	3.2	703,042.9	1.22
57	110,000	2,000	250	16	1.6	1.6	574,263.3	1.14
58	110,000	2,000	250	16	1.6	3.2	625,527.5	0.85
59	110,000	2,000	250	16	3.2	1.6	596,580.9	1.93
60	110,000	2,000	250	16	3.2	3.2	633,558.9	1.44
61	110,000	2,000	250	20	1.6	1.6	665,765.3	0.91
62	110,000	2,000	250	20	1.6	3.2	695,144.9	0.63
63	110,000	2,000	250	20	3.2	1.6	681,865.8	1.48
64	110,000	2,000	250	20	3.2	3.2	702,770.8	1.23
Grand mean							503,991.8	1.20

A.3 Factor means plot

The plots on the main diagonal of the factor means plot show the mean relative value of two-stage modelling *relVTSM* at every level of the factors, while the off-diagonal plots display the mean response at every combination of two factors. For example, the plot in the upper left corner shows the mean *relVTSM* for both levels of fixed costs *FC*. The plot below shows the same means where the observations have been subdivided by the mean second-stage demand μ^{D}. Level 1 and Level 2 refer to the factor levels in Table 2.1.

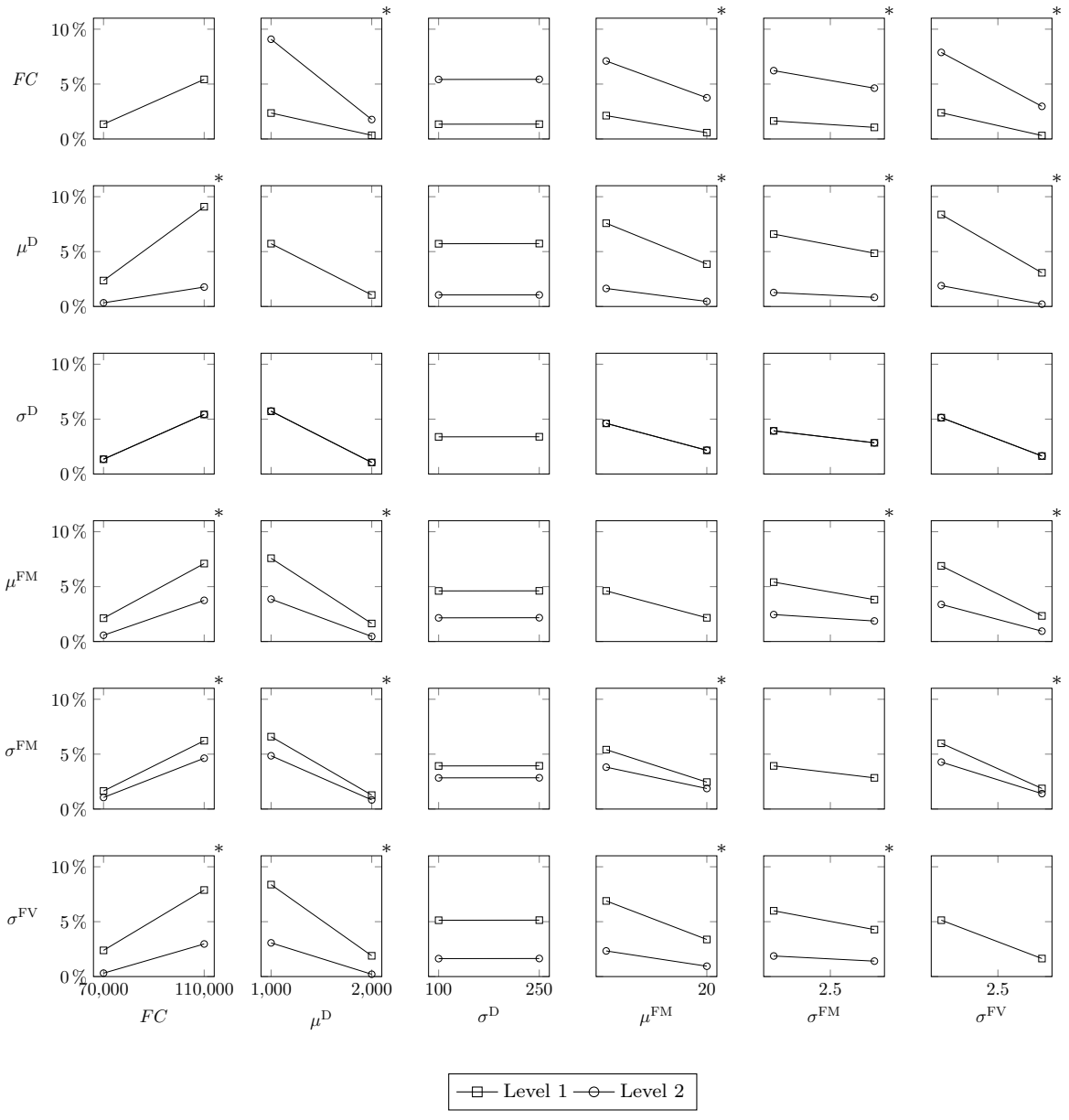


Figure A.1: Factor means plot for *relVTSM* (* significant interaction effect at the 99.0 % confidence level).

A.4 Reference model

The following two-stage stochastic programming formulation is equivalent to the formulation presented in Section 2.3.2. Given that both formulations solve the same TPPDP and the problem is not degenerated, both formulation will provide the same optimal solution. The difference is that this formulation relies on the modelling approach proposed by Boysen and Scholl (2009). The assumptions and the notation introduced in Section 2.3.2 remain valid. In addition, we define the following parameters:

T^1_{rfv} 1 if value $v \in V_f$ is in the tolerance interval defined for feature $f \in F$ of variant $r \in R^1$ and 0 otherwise.

T^2_{rfvs} 1 if value $v \in V_f$ is in the tolerance interval defined for feature $f \in F$ of variant $r \in R^2$ in scenario $s \in S$ and 0 otherwise.

We also define additional variables following the notation of Boysen and Scholl:

z^1_{pfv} 1 if value $v \in V_f$ of feature $f \in F$ is realized in platform $p \in P^1$ and 0 otherwise.

z^2_{pfvs} 1 if value $v \in V_f$ of feature $f \in F$ is realized in platform $p \in P^2$ in scenario $s \in S$ and 0 otherwise.

q^1_{rpfv} 1 if variant $r \in R^1$ is assigned to platform $p \in P^1$ that has value $v \in V_f$ in feature $f \in F$ and 0 otherwise.

q^2_{rpfvs} 1 if variant $r \in R^2$ is assigned in scenario $s \in S$ to platform $p \in P^1 \cup P^2$ that has value $v \in V_f$ in feature $f \in F$ and 0 otherwise.

$$\min \sum_{r\in R^1}\sum_{p\in P^1}\sum_{f\in F}\sum_{v\in V_f} D^1_r\, VC_{fv} q^1_{rpfv} + \sum_{p\in P^1} FC y^1_p + \sum_{s\in S} Pr_s Q(x^1_{rp}, z^1_{pfv}, q^1_{rpfv}, y^1_p, s) \quad \text{(A.1)}$$

$$\sum_{p\in P^1} x^1_{rp} \geq 1 \qquad \forall r \in R^1 \quad \text{(A.2)}$$

$$x^1_{rp} \leq y^1_p \qquad \forall r \in R^1, p \in P^1 \quad \text{(A.3)}$$

$$\sum_{v\in V_f} z^1_{pfv} = 1 \qquad \forall p \in P^1, f \in F \quad \text{(A.4)}$$

$$q^1_{rpfv} \geq x^1_{rp} + z^1_{pfv} - 1 \qquad \forall r \in R^1, p \in P^1, f \in F, v \in V_f \quad \text{(A.5)}$$

$$T^1_{rfv} \geq q^1_{rpfv} \qquad \forall r \in R^1, p \in P^1, f \in F, v \in V_f \quad \text{(A.6)}$$

$$q^1_{rpfv}, y^1_p \in \mathbb{R}_{\geq 0} \quad \text{(A.7)}$$

$$x^1_{rp}, z^1_{pfv} \in \{0, 1\} \quad \text{(A.8)}$$

The recourse function is:

$$Q(x^1_{rp}, z^1_{pfv}, q^1_{rpfv}, y^1_p, s) = \sum_{f\in F}\sum_{v\in V_f} C_{fv}\left(\sum_{r\in R^1}\sum_{p\in P^1} D^2_{rs} q^1_{rpfv} + \sum_{r\in R^2}\sum_{p\in P^1\cup P^2} D^2_{rs} q^2_{rpfvs}\right) + \sum_{p\in P^2} FC y^2_{ps} \quad \text{(A.9)}$$

$$\sum_{p\in P^1\cup P^2} x^2_{rps} \geq 1 \qquad \forall r \in \{r : r \in R^2 \wedge D^2_{rs} > 0\} \quad \text{(A.10)}$$

$$x^2_{rps} \leq \begin{cases} y^1_p & \forall r \in R^2, p \in P^1 \\ y^2_{ps} & \forall r \in R^2, p \in P^2 \end{cases} \quad \text{(A.11)}$$

$$\sum_{v\in V_f} z^2_{pfvs} = 1 \qquad \forall p \in P^2, f \in F \quad \text{(A.12)}$$

$$q^2_{rpfvs} \geq \begin{cases} x^2_{rps} + z^1_{pfv} - 1 & \forall r \in R^2, p \in P^1, f \in F, v \in V_f \\ x^2_{rps} + z^2_{pfvs} - 1 & \forall r \in R^2, p \in P^2, f \in F, v \in V_f \end{cases} \quad \text{(A.13)}$$

$$T^2_{rfvs} \geq q^2_{rpfvs} \qquad \forall r \in R^2, p \in P^1 \cup P^2, f \in F, v \in V_f \quad \text{(A.14)}$$

$$q^2_{rpfvs}, y^2_{ps} \in \mathbb{R}_{\geq 0} \quad \text{(A.15)}$$

$$x^2_{rps}, z^2_{pfvs} \in \{0, 1\} \quad \text{(A.16)}$$

Appendix B

B.1 Multifactor ANOVA for γ-service level and number of lots in system

The ANOVA Tables B.1 and B.2 decompose the variability of γ-service level and the variability of number of lots in system ($WIP + FWS$) into contributions due to various factors. The contribution of each factor is measured having removed the effects of all other factors. The P-values test the statistical significance of each of the factors. Since all P-values are less than 0.05, these factors have a statistically significant effect on γ-service level and the number of lots in system at the 95.0 % confidence level.

Table B.1: ANOVA for γ-service level (all F-ratios are based on the residual mean square error).

Source	Sum of squares	Degrees of freedom	Mean square	F-ratio	P-value
MAIN EFFECTS					
A:Method	3,348.54	2	1,674.27	103.12	0.0000
B:Demand uncertainty	7,076.71	2	3,538.36	217.92	0.0000
C:Supply uncertainty	960.56	1	960.56	59.16	0.0000
INTERACTIONS					
AB	368.12	4	920.30	5.67	0.0005
AC	981.13	2	490.57	30.21	0.0000
BC	209.94	2	104.97	6.46	0.0026
RESIDUAL	1,233.98	76	162.37		
TOTAL (CORRECTED)	14,179.00	89			

Table B.2: ANOVA for number of lots in system (all F-ratios are based on the residual mean square error).

Source	Sum of squares	Degrees of freedom	Mean square	F-ratio	P-value
MAIN EFFECTS					
A:Method	9,035.52	2	4,517.76	2,253.60	0.0000
B:Demand uncertainty	13,864.90	2	6,932.44	3,458.12	0.0000
C:Supply uncertainty	26,191.50	1	26,191.50	13,065.18	0.0000
INTERACTIONS					
AB	2,799.72	4	699.93	349.15	0.0000
AC	2,455.90	2	1,227.95	612.54	0.0000
BC	102.57	2	512.85	25.58	0.0000
RESIDUAL	152.36	76	200.47		
TOTAL (CORRECTED)	54,602.50	89			

B.2 Tukey's HSD test for differences between means

The Tables B.3 to B.6 present treatment means that are plotted in Figure 3.6 and the result of Tukey's HSD test. At each level of uncertainty, we test all pairs of treatment means for equality at an overall significance level of 5 %. The result is provided by letters on the left-hand side next to each column of treatment means. If the same letter stands next to two treatment means in a column, the hypothesis of equality cannot be rejected. Configurations without machine failures and $CV_{D_{i,t}} = 0.0$ are not included in the test because the response is not a random variable. The constant release rate method is not included in the test because variance is much greater than for the other MTPPS methods, which violates the assumption of homogeneity of variances.

Table B.3: Comparing treatment means of γ-service level at each level of demand uncertainty *without* machine failures.

	$CV_{D_{i,t}}$ 0.1	0.2	0.3
CT-oriented planning	a 99.7	a 95.3	a 85.6
Release planning	b 94.5	b 83.8	b 77.0
WIP-oriented planning	c 96.4	b 86.8	b 74.0

Table B.4: Comparing treatment means of γ-service level at each level of demand uncertainty *with* machine failures.

	$CV_{D_{i,t}}$ 0.0	0.1	0.2	0.3
CT-oriented planning	a 100.0	a 99.8	a 95.3	a 80.4
Release planning	a 99.5	b 94.8	b 83.9	a 70.5
WIP-oriented planning	b 94.0	c 86.8	c 68.4	b 54.3

Table B.5: Comparing treatment means of $\overline{WIP} + \overline{FWS}$ at each level of demand uncertainty *without* machine failures.

	$CV_{D_{i,t}}$ 0.1	0.2	0.3
CT-oriented planning	b 216.8	b 231.9	b 255.3
Release planning	c 203.2	c 216.1	c 236.7
WIP-oriented planning	a 246.1	a 252.3	a 257.4

Table B.6: Comparing treatment means of $\overline{WIP} + \overline{FWS}$ at each level of demand uncertainty *with* machine failures.

	$CV_{D_{i,t}}$ 0.0	0.1	0.2	0.3
CT-oriented planning	b 255.2	b 259.3	a 272.8	a 302.4
Release planning	c 234.9	c 238.5	b 254.0	b 281.3
WIP-oriented planning	a 259.9	a 265.6	a 271.8	c 277.1

Appendix C

C.1 Proof of Proposition 1

We observe:

1. For $j \in J \backslash J(i)$, the inequalities (4.18) to (4.20) imply $x_{ij} = 0$ and there does not exist any $\boldsymbol{r}_{ij}$ in R_i.

2. For $j \in J(i)$, we have developed the set of $|J(i)|+1$ extreme points $R_i \cup \{\mathbf{0}\}$, which are affinely independent and therefore define a simplex.

3. This simplex has $|J(i)|+1$ vertices, which can also be obtained by imposing that $|J(i)|$ out of the $|J(i)|+1$ inequalities (4.18) to (4.20) are satisfied at equality. By direct calculation, this yields the points in $R_i \cup \{\mathbf{0}\}$.

This proves that $conv(R_i \cup \{\mathbf{0}\}) = \cap\mathcal{H}(P_i)$ with $\cap\mathcal{H}(P_i)$ representing the set of feasible $\boldsymbol{x}_i$ subject to (4.18) to (4.20). For more information on convex polytopes see, e.g., Grünbaum (2003) and Ziegler (2007).

C.2 Experiment with randomly generated data

Table C.1: Uniform problem instance with $\rho = 25\,\%$ and $p_{ij} \in \{1, 100, 200, \infty\}$.

		Machine i											
	p_{ij}	1	2	3	4	5	6	7	8	9	10	11	12
Product j	1	1	∞	∞	∞	100	∞	∞	∞	200	∞	∞	∞
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	125	1	∞	∞	∞	100	∞	∞	∞	200	∞	∞	∞
	126	∞	∞	∞	100	∞	∞	∞	200	∞	∞	∞	1
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	250	∞	∞	∞	100	∞	∞	∞	200	∞	∞	∞	1
	251	∞	∞	100	∞	∞	∞	200	∞	∞	∞	1	∞
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	375	∞	∞	100	∞	∞	∞	200	∞	∞	∞	1	∞
	376	∞	100	∞	∞	∞	200	∞	∞	∞	1	∞	∞
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	500	∞	100	∞	∞	∞	200	∞	∞	∞	1	∞	∞

Table C.2: Problem instance with $\rho = 50\,\%$, $p_{ij} \in \{1, 5, 10, \infty\}$, and a swapped pair of randomly selected 25×3-submatrices (marked by dashed rectangles).

		Machine i											
	p_{ij}	1	2	3	4	5	6	7	8	9	10	11	12
Product j	1	1	∞	5	∞	10	∞	1	∞	5	∞	10	∞
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	249	1	∞	5	∞	10	∞	1	∞	5	∞	10	∞
	250	1	∞	1	∞	10	∞	1	∞	5	∞	10	∞
	251	∞	∞	1	∞	∞	10	∞	1	∞	5	∞	10
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	274	∞	∞	1	∞	∞	10	∞	1	∞	5	∞	10
	275	∞	1	∞	5	∞	10	∞	1	∞	5	∞	10
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	474	∞	1	∞	5	∞	10	∞	1	∞	5	∞	10
	475	∞	1	∞	5	∞	10	∞	5	∞	5	∞	10
	476	∞	1	∞	5	∞	10	1	∞	5	5	∞	10
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	499	∞	1	∞	5	∞	10	1	∞	5	5	∞	10
	500	∞	1	∞	5	∞	10	∞	1	∞	5	∞	10

Table C.3: ANOVA for OFI^{PCS} (all F-ratios are based on the residual mean square error).

Source	Sum of squares	Df	Mean square	F-ratio	P-value
MAIN EFFECTS					
A: Constraint generation method	9.69	1	9.69	450.84	0.0000
B: ρ	1.73	1	1.73	80.69	0.0000
C: Range of p_{ij}	2.78	1	2.78	129.10	0.0000
D: Number of swaps	2.52	2	1.26	58.50	0.0000
INTERACTIONS					
AB	0.27	1	0.27	12.73	0.0004
AC	1.92	1	1.92	89.52	0.0000
AD	0.27	2	0.13	6.23	0.0021
BC	0.19	1	0.19	8.72	0.0031
BD	0.42	2	0.21	9.77	0.0001
CD	0.15	2	0.08	3.53	0.0297
RESIDUAL	15.16	705	0.02		
TOTAL (CORRECTED)	35.10	719			

Bibliography

Asmundsson, J., Rardin, R. L., and Uzsoy, R. (2006). Tractable nonlinear production planning models for semiconductor wafer fabrication facilities. *IEEE Transactions on Semiconductor Manufacturing*, 19(1):95–111.

Atherton, L. F. and Atherton, R. W. (1995). *Wafer Fabrication: Factory Performance and Analysis*, volume 339. Springer.

Avis, D., Bremner, D., and Seidel, R. (1997). How good are convex hull algorithms? *Computational Geometry*, 7(5):265–301.

Avis, D. and Fukuda, K. (1992). A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete and Computational Geometry*, 8(1):295–313.

Avis, D. and Roumanis, G. (2013). A portable parallel implementation of the lrs vertex enumeration code. In Widmayer, P., Xu, Y., and Zhu, B., editors, *Combinatorial Optimization and Applications*, volume 8287 of *Lecture Notes in Computer Science*, pages 414–429. Springer, Cham, Switzerland.

Bard, J. F., Deng, Y., Chacon, R., and Stuber, J. (2010). Midterm planning to minimize deviations from daily target outputs in semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing*, 23(3):456–467.

Barua, A., Raghavan, N., Upasani, A., and Uzsoy, R. (2005). Implementing global factory schedules in the face of stochastic disruptions. *International Journal of Production Research*, 43(4):793–818.

Ben-Arieh, D., Easton, T., and Choubey, A. M. (2009). Solving the multiple platforms configuration problem. *International Journal of Production Research*, 47(7):1969–1988.

Bermon, S. and Hood, S. J. (1999). Capacity optimization planning system. *Interfaces*, 29(5):31–50.

Billington, P. J., McClain, J. O., and Thomas, L. J. (1983). Mathematical programming approaches to capacity-constrained MRP systems: review, formulation and problem reduction. *Management Science*, 29(10):1126–1141.

Birge, J. R. and Louveaux, F. (2011). *Introduction to Stochastic Programming*. Springer Science & Business Media.

Boysen, N. and Scholl, A. (2009). A general solution framework for component-commonality problems. *BuR - Business Research*, 2(1):86–106.

Briant, O. and Naddef, D. (2004). The optimal diversity management problem. *Operations Research*, 52(4):515–526.

Cai, Y., Kutanoglu, E., and Hasenbein, J. (2011). Production planning and scheduling: Interaction and coordination. In Kempf, K. G., Keskinocak, P., and Uzsoy, R., editors, *Planning Production and Inventories in the Extended Enterprise*, volume 152 of *International Series in Operations Research & Management Science*, pages 15–42. Springer.

Centeno, G. and Armacost, R. L. (2004). Minimizing makespan on parallel machines with release time and machine eligibility restrictions. *International Journal of Production Research*, 42(6):1243–1256.

Chung, S.-H., Huang, C.-Y., and Lee, A. H. I. . (2008). Heuristic algorithms to solve the capacity allocation problem in photolithography area. *OR Spectrum*, 30(3):431–452.

Cisco (2018). Cisco Visual Networking Index: Forecast and Trends, 2017–2022. White Paper.

Ehm, H., Ponsignon, T., and Kaufmann, T. (2011). The global supply chain is our new fab: Integration and automation challenges. In *Advanced Semiconductor Manufacturing Conference (ASMC), 2011 22nd Annual IEEE/SEMI*, pages 1–6. IEEE.

Fisher, M., Ramdas, K., and Ulrich, K. (1999). Component sharing in the management of product variety: A study of automotive braking systems. *Management Science*, 45(3):297–315.

Fixson, S. K. (2007). Modularity and commonality research: Past developments and future opportunities. *Concurrent Engineering*, 15(2):85–111.

Fowler, J. W. and Robinson, J. (1995). Measurement and improvement of manufacturing capacities (MIMAC): Final report. Technical report, SEMATECH. 95062861A-TR.

Fronckowiak, D., Peikert, A., and Nishinohara, K. (1996). Using discrete event simulation to analyze the impact of job priorities on cycle time in semiconductor manufacturing. In *Advanced Semiconductor Manufacturing Conference and Workshop, 1996. ASMC 96 Proceedings. IEEE/SEMI 1996*, pages 151–155.

Fujita, K. and Yoshida, H. (2004). Product variety optimization simultaneously designing module combination and module attributes. *Concurrent Engineering*, 12(2):105–118.

Fukuda, K. (2004). From the zonotope construction to the Minkowski addition of convex polytopes. *Journal of Symbolic Computation*, 38(4):1261–1272.

Fukuda, K. (2008). *cddlib Reference Manual*. Swiss Federal Institute of Technology, Zurich, Switzerland.

Fukuda, K. and Prodon, A. (1996). Double description method revisited. In Deza, M., Euler, R., and Manoussakis, I., editors, *Combinatorics and Computer Science*, volume 1120 of *Lecture Notes in Computer Science*, pages 91–111. Springer, Berlin, Germany.

Fukuda, K. and Weibel, C. (2007). f-vectors of Minkowski additions of convex polytopes. *Discrete and Computational Geometry*, 37(4):503–516.

Fukuda, K. and Weibel, C. (2009). Facet computation for Minkowski sums of polytopes. Preprint received from Komei Fukuda.

Garey, M. R. and Johnson, D. S. (1979). *Computers and intractability: A guide to NP-completeness*. Freeman, New York, NY.

Geng, H. (2005). *Semiconductor manufacturing handbook*. McGraw-Hill, Inc.

Govind, N., Bullock, E. W., He, L., Iyer, B., Krishna, M., and Lockwood, C. S. (2008). Operations management in automated semiconductor manufacturing with integrated targeting, near real-time scheduling, and dispatching. *IEEE Transactions on Semiconductor Manufacturing*, 21(3):363–370.

Grünbaum, B. (2003). *Convex Polytopes*. Springer, New York, NY.

Hackman, S. T. and Leachman, R. C. (1989). A general framework for modeling production. *Management Science*, 35(4):478–495.

Hillier, M. S. (2000). Component commonality in multiple-period, assemble-to-order systems. *IIE Transactions*, 32(8):755–766.

Hopp, W. J. and Spearman, M. L. (2011). *Factory physics*. Waveland, Long Grove, IL.

Hung, Y.-F. and Cheng, G.-J. (2002). Hybrid capacity modeling for alternative machine types in linear programming production planning. *IIE Transactions*, 34:157–165.

Hwang, T.-K. and Chang, S.-C. (2003). Design of a Lagrangian relaxation-based hierarchical production scheduling environment for semiconductor wafer fabrication. *IEEE Transactions on Robotics and Automation*, 19(4):566–578.

IEEE-SA (2015). IEEE-SA Internet of Things (IoT) Ecosystem Study. IEEE Standards Association.

Intel (2017). Intel supports American innovation with $7 billion investment in next-generation semiconductor factory in Arizona. http://fw.to/M8rFgAM. Intel Corporation. Accessed: 2018-11-18.

Irdem, D. F., Kacar, N. B., and Uzsoy, R. (2010). An exploratory analysis of two iterative linear programming-simulation approaches for production planning. *IEEE Transactions on Semiconductor Manufacturing*, 23(3):442–455.

Jans, R., Degraeve, Z., and Schepens, L. (2008). Analysis of an industrial component commonality problem. *European Journal of Operational Research*, 186(2):801–811.

Jiao, J. R., Simpson, T. W., and Siddique, Z. (2007). Product family design and platform-based product development: A state-of-the-art review. *Journal of Intelligent Manufacturing*, 18(1):5–29.

Johnson, L. A. and Montgomery, D. C. (1974). *Operations Research in Production Planning, Scheduling, and Inventory Control*. Wiley, New York, NY.

Jula, P. and Leachman, R. C. (2008). Coordinating decentralized local schedulers in complex supply chain manufacturing. *Annals of Operations Research*, 161(1):123–147.

Kilic, O. A., Akkerman, R., van Donk, D. P., and Grunow, M. (2013). Intermediate product selection and blending in the food processing industry. *International Journal of Production Research*, 51(1):26–42.

Kim, B. and Kim, S. (2001). Extended model for a hybrid production planning approach. *International Journal of Production Economics*, 73(2):165–173.

Kim, D.-W., Kim, K.-H., Jang, W., and Frank Chen, F. (2002). Unrelated parallel machine scheduling with setup times using simulated annealing. *Robotics and Computer-Integrated Manufacturing*, 18(3):223–231.

Kim, J. S. and Leachman, R. C. (1994). Decomposition method application to a large scale linear programming WIP projection model. *European Journal of Operational Research*, 74(1):152–160.

Kriett, P. O., Eirich, S., and Grunow, M. (2017). Cycle time-oriented mid-term production planning for semiconductor wafer fabrication. *International Journal of Production Research*, 55(16):4662–4679.

Kriett, P. O. and Grunow, M. (2017). Generation of low-dimensional capacity constraints for parallel machines. *IISE Transactions*, 49(12):1189–1205.

Kumar, S. and Wellbrock, J. (2009). Improved new product development through enhanced design architecture for engineer-to-order companies. *International Journal of Production Research*, 47(15):4235–4254.

Labro, E. (2004). The cost effects of component commonality: A literature review through a management-accounting lens. *Manufacturing & Service Operations Management*, 6(4):358–367.

Leachman, R. C. (2002). Semiconductor production planning. In Pardalos, P. M. and Resende, M. G. C., editors, *Handbook of Applied Optimization*, pages 746–762. Oxford University Press.

Leachman, R. C. and Carmon, T. F. (1992). On capacity modeling for production planning with alternative machine types. *IIE Transactions*, 24(4):62–72.

Leachman, R. C., Kang, J., and Lin, V. (2002). SLIM: Short cycle time and low inventory in manufacturing at samsung electronics. *Interfaces*, 32(1):61–77.

L'Ecuyer, P., Meliani, L., and Vaucher, J. (2002). SSJ: A framework for stochastic simulation in Java. In *Proceedings of the 34th Conference on Winter Simulation: Exploring New Frontiers*, pages 234–242.

Lee, H. L. and Tang, C. S. (1997). Modelling the costs and benefits of delayed product differentiation. *Management Science*, 43(1):40–53.

Lee, Y. H., Park, J., and Kim, S. (2002). Experimental study on input and bottleneck scheduling for a semiconductor fabrication line. *IIE Transactions*, 34(2):179–190.

Liberopoulos, G. (2002). Production capacity modeling of alternative, nonidentical, flexible machines. *International Journal of Flexible Manufacturing Systems*, 14(4):345–359.

Logendran, R. and Subur, F. (2004). Unrelated parallel machine scheduling with job splitting. *IIE Transactions*, 36(4):359–372.

Lu, S. C. H., Ramaswamy, D., and Kumar, P. R. (1994). Efficient scheduling policies to reduce mean and variance of cycle-time in semiconductor manufacturing plants. *IEEE Transactions on Semiconductor Manufacturing*, 7(3):374–388.

Mack, C. A. (2011). Fifty years of moore's law. *IEEE Transactions on semiconductor manufacturing*, 24(2):202–207.

Menezes, M. B., Ruiz-Hernández, D., and Guimaraes, R. (2016). The component commonality problem in a real multidimensional space: An algorithmic approach. *European Journal of Operational Research*, 249(1):105–116.

Missbauer, H. (2002). Aggregate order release planning for time-varying demand. *International Journal of Production Research*, 40(3):699–718.

Missbauer, H. and Uzsoy, R. (2011). Optimization models of production planning problems. In Kempf, K. G., Keskinocak, P., and Uzsoy, R., editors, *Planning Production and Inventories in the Extended Enterprise*, volume 151 of *International Series in Operations Research & Management Science*, pages 437–507. Springer, New York, NY.

Mönch, L., Fowler, J. W., and Mason, S. J. (2013). *Production Planning and Control for Semiconductor Wafer Fabrication Facilities*, volume 52 of *Operations Research/Computer Science Interfaces Series*. Springer, New York, NY.

Moore, G. E. (1965). Cramming more components onto integrated circuits. *Electronics*, 38(8):114–117.

Motzkin, T. S., Raiffa, H., Thompson, G. L., and Thrall, R. M. (1953). The double description method. In Kuhn, H. W. and Tucker, A. W., editors, *Contributions to the Theory of Games*, volume 2, pages 51–73. Princeton University Press, Princeton, NJ.

Perera, H., Nagarur, N., and Tabucanon, M. T. (1999). Component part standardization: A way to reduce the life-cycle costs of products. *International Journal of Production Economics*, 60:109–116.

Pinedo, M. (2012). *Scheduling: Theory, Algorithms, and Systems*. Springer, New York, NY.

Samsung (2017). Samsung Electronics begins mass production at new semiconductor plant in Pyeongtaek, South Korea. https://www.samsung.com/semiconductor/insights/news-events/samsung-electronics-begins-mass-production-at-new-semiconductor-plant-in-pyeongtaek-south-korea/. Samsung. Accessed: 2018-11-18.

Sanders, P. and Schulz, C. (2013). Think locally, act globally: Highly balanced graph partitioning. In *Proceedings of the 12th International Symposium on Experimental Algorithms (SEA'13)*, volume 7933 of *Lecture Notes in Computer Science*, pages 164–175. Springer, Heidelberg, Germany.

Sawik, T. (2006). Hierarchical approach to production scheduling in make-to-order assembly. *International Journal of Production Research*, 44(4):801–830.

Selçuk, B., Fransoo, J. C., and De Kok, A. G. (2008). Work-in-process clearing in supply chain operations planning. *IIE Transactions*, 40(3):206–220.

Simpson, T. W. (2004). Product platform design and customization: Status and promise. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 18(01):3–20.

Song, Y., Zhang, M. T., Yi, J., Zhang, L., and Zheng, L. (2007). Bottleneck station scheduling in semiconductor assembly and test manufacturing using ant colony optimization. *IEEE Transactions on Automation Science and Engineering*, 4(4):569–578.

Sourirajan, K. and Uzsoy, R. (2007). Hybrid decomposition heuristics for solving large-scale scheduling problems in semiconductor wafer fabrication. *Journal of Scheduling*, 10(1):41–65.

Stadtler, H. (2015). Supply chain management: An overview. In Stadtler, H., Kilger, C., and Meyr, H., editors, *Supply Chain Management and Advanced Planning*, Springer Texts in Business and Economics, pages 3–28. Springer Berlin Heidelberg.

Su, J. C., Chang, Y.-L., and Ferguson, M. (2005). Evaluation of postponement structures to accommodate mass customization. *Journal of Operations Management*, 23(3):305–318.

Swaminathan, J. M. and Tayur, S. R. (1998). Managing broader product lines through delayed differentiation using vanilla boxes. *Management Science*, 44(12-part-2):161–172.

Thonemann, U. W. and Brandeau, M. L. (2000). Optimal commonality in component design. *Operations Research*, 48(1):1–19.

Tiwary, H. R. (2008). On the hardness of computing intersection, union and Minkowski sum of polytopes. *Discrete and Computational Geometry*, 40(3):469–479.

TSMC (2018). TSMC breaks ground on Fab 18 in Southern Taiwan Science Park. http://www.tsmc.com/tsmcdotcom/PRListingNewsAction.do?action=detail&language=E&newsid=THGOHITHTH. Taiwan Semiconductor Manufacturing Company Limited. Accessed: 2018-11-18.

Turley, J. L. and Turley, J. (2003). *The essential guide to semiconductors*. Prentice Hall Professional.

Uzsoy, R., Lee, C.-Y., and Martin-Vega, L. A. (1992). A review of production planning and scheduling models in the semiconductor industry part I: System characteristics, performance evaluation and production planning. *IIE Transactions*, 24(4):47–60.

Uzsoy, R., Lee, C.-Y., and Martin-Vega, L. A. (1994). A review of production planning and scheduling models in the semiconductor industry part II: Shop-floor control. *IIE Transactions*, 26(5):44–55.

Vollmann, T. E., Berry, W. L., and Whybark, D. C. (1997). *Manufacturing Planning and Control Systems*. Irwin/McGraw-Hill, New York, NY.

Vollmann, T. E., Berry, W. L., Whybark, D. C., and Jacobs, F. R. (2005). *Manufacturing planning and control for supply chain management*. McGraw-Hill/Irwin.

Wazed, M. A., Ahmed, S., and Nukman, Y. (2010). Commonality in manufacturing resources planning - issues and models: a review. *European Journal of Industrial Engineering*, 4(2):167–188.

Weibel, C. (2010). Implementation and parallelization of a reverse-search algorithm for Minkowski sums. In Halperin, D. and Blelloch, G., editors, *2010 Proceedings of the Twelfth Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 34–42, Austin, TX. SIAM.

WSTS (2018). Worldwide semiconductor market is expected to be up 15.7 percent in 2018 after 21.6 percent growth in 2017. https://www.wsts.org/esraCMS/extension/media/f/WST/3613/WSTS-nr-2018_08.pdf. World Semiconductor Trade Statistics. Accessed: 2018-11-18.

Yu, L., Shih, H. M., Pfund, M., Carlyle, W. M., and Fowler, J. W. (2002). Scheduling of unrelated parallel machines: An application to PWB manufacturing. *IIE Transactions*, 34(11):921–931.

Ziegler, G. M. (2007). In Axler, S., Gehring, F. W., and Ribet, K. A., editors, *Lectures on Polytopes*, volume 152 of *Graduate Texts in Mathematics*. Springer, New York, NY.

ZVEI (2017). Deutscher Halbleitermarkt: 2017 hohes Wachstum. https://www.zvei.org/presse-medien/pressebereich/deutscher-halbleitermarkt-2017-hohes-wachstum/. Zentralverband Elektrotechnik- und Elektronikindustrie e.V.. Accessed: 2018-11-18.

ZVEI (2018). Deutsche Elektroindustrie mit höchster Beschäftigtenzahl seit 17 Jahren. https://www.zvei.org/presse-medien/pressebereich/deutsche-elektroindustrie-mit-hoechster-beschaeftigtenzahl-seit-17-jahren/. Zentralverband Elektrotechnik- und Elektronikindustrie e.V.. Accessed: 2018-11-18.