MODELING AND SOLVING MASTER PLANNING PROBLEMS IN SEMICONDUCTOR MANUFACTURING

Dissertation
zur Erlangung des Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.)

Fakultät für Mathematik und Informatik
der FernUniversität in Hagen

vorgelegt von
Thomas Ponsignon, M.Eng.

Hagen, 2012

Foreword

This dissertation came into existence as a part of a collaboration between the Chair of Enterprise-wide Software Systems of the University of Hagen and Infineon Technologies AG. The development of this thesis took place during my employment time at Infineon in Munich, Germany and Dublin, Ireland first as an external PhD candidate, and later as a supply chain engineer.

I would like to acknowledge and extend my heartfelt thanks to the following people for making this dissertation possible. First of all, I would like to express my utmost gratitude to Professor Lars Mönch who supervised this thesis. Throughout the years, he provided me with always constructive and valuable suggestions that contributed much to the contents of this work. I could take advice from him, which will be of great value for my further professional career.

I am also very grateful to Professor Stéphane Dauzère-Pérès for accepting to act as a reviewer of this thesis.

I would like to express my deepest appreciation to Hans Ehm who made possible the collaboration between Infineon and the University of Hagen. I am also thankful to Tony Smyth and Sascha Krall for allowing me to pursue my research during my employment time at Infineon.

Finally, I owe sincere thanks to my family for their steadfast support. Last but not least, I extend my special thanks to Géraldine for her abiding patience and her unceasing confidence in my ability to succeed.

Abstract

This thesis deals with mid-term production planning problems, i.e., master planning, that arise in semiconductor manufacturing. Given the specifics of semiconductor manufacturing networks, the development of enterprise-wide planning approaches that are computationally tractable and address the uncertainties typically encountered in this industry remains particularly challenging.

The purpose of production planning is to allocate limited resources to competing demands over time with respect to often conflicting economic objectives. Depending on the nature of the considered problems, production planning models may require the usage of integer-valued variables. Such models may be difficult to solve in a reasonable amount of time for large-scale instances as usually encountered in enterprise-wide planning environments. Therefore, efficient optimization approaches have to be used to reduce the computational effort while achieving optimal, or near optimal, problem solutions.

The performance of the designed optimization algorithms is assessed, at first, using single problem instances. However, given the uncertainty that is typical for the semiconductor industry, there is a need for incorporating different sources of environment- and system-related disruptions into the evaluation of the planning approaches. For this purpose, a simulation model appears to adequately mimic the stochastic behavior of a semiconductor manufacturing network. In real-world situations, the planning activities occur on a regular basis; it allows for replanning the production plan by taking the current state of the input parameters into account. The investigation of the obtained rolling plans provides further insights into the performance of the planning approach used, which is usually not achievable when considering single problem instances.

Production planning is complicated by the interaction between lead time and resource utilization. It is known from queueing theory that the cycle time increases nonlinearly with the utilization of the resources. However, the utilization is a result of the release schedule used. This leads to circularity in production planning. On the one hand, the planning approach determines the release schedule based on a prescribed lead time. On the other hand, the cycle time depends on the release schedule. The models presented previously in this thesis assume a fixed product lead time as an exogenous parameter of the planning approach. Although this assumption makes sense for highly aggregated strategic planning problems, it is not desirable for mid-term production planning decisions. Among other approaches, the iterations between a planning approach that determines the releases of production quantities based on a prescribed lead time and a simulation model that uses these production quantities to calculate cycle time estimates seem to adequately tackle the circularity in production planning.

Contents

Index of Tables

Index of Figures

Index of Algorithms

List of Abbreviations

ANOVA	Analysis of variance
APS	Advanced planning system
BB-MPSC	Branch-and-bound algorithm for solving MPSC
CAPEX	Capital expenditure
CI	Confidence interval
CRP	Capacity requirements planning
CTM	Capable-to-match
CVD	Chemical vapor deposition
EDD	Earliest due date
ERP	Enterprise resource planning
FIFO	First-in, first-out
IC	Integrated circuit
IDM	Integrated device manufacturer
GA	Genetic algorithm
GA-MPSC	Genetic algorithm for solving MPSC
HLA	High-level architecture
KSP	Knapsack problem
MAD	Mean absolute deviation
MD	Mean deviation
MES	Manufacturing execution system
MIMAC	Measurement and improvement of manufacturing capacity
MIP	Mixed integer programming
MP	Master planning
MPC	Manufacturing planning and control
MPS	Master production scheduling
MPSC	Master planning in semiconductor manufacturing
MRP	Material requirements planning
MRP-II	Manufacturing resources planning
MTTF	Mean time to failure
MTTR	Mean time to repair
NP-hard	Non-deterministic polynomial-time hard
PD-MPSC	Product-based decomposition scheme for solving MPSC
PM	Preventive maintenance
PMT	Preventive maintenance time
PVD	Physical vapor deposition

RA-MPSC	Rule-based assignment scheme for solving MPSC
SAP® APO	SAP® Advanced planner optimizer
SCM	Supply chain management
SCOR	Supply chain operations reference (model)
SCPM	Supply chain planning matrix
TTPM	Time to preventive maintenance
UML	Unified modeling language
WIP	Work in progress

1. Introduction

1.1 Motivation

The semiconductor industry is known as one of the most dynamic businesses in today's worldwide economy. Over the last fifteen years, the semiconductor market rose at an average annual growth rate of nearly 11%, whereas the worldwide inflation-adjusted Gross Domestic Product increased by only 3% during the same period (cf. WSTS, 2012). However, this fast growth pace underwent high fluctuations. Years of booming business (1999-2000, 2003-2004, 2010) were followed by severe economic downturns (2001, 2008-2009, 2011) indicating that the semiconductor market is highly dependent on economic and geopolitical conditions. Figure 1.1 shows a chart of the annual revenue and market growth rate of the worldwide semiconductor market.

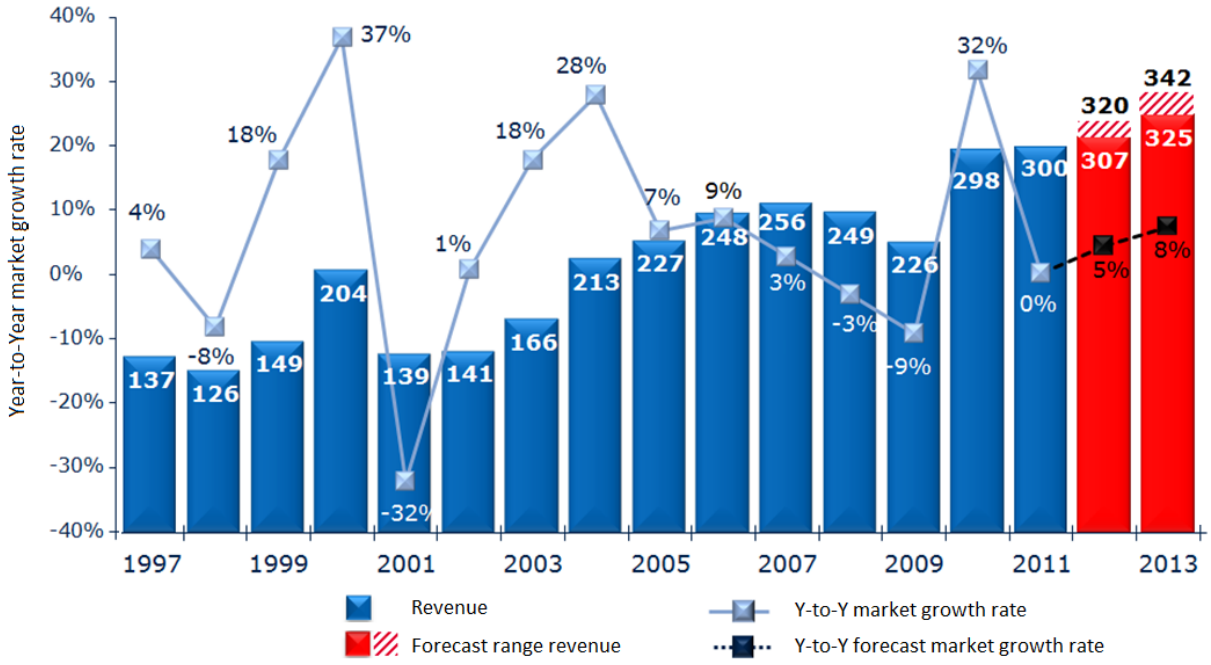

Figure 1.1: Annual revenue in billion US dollars and market growth rate of the worldwide semiconductor market (cf. WSTS, 2012).

Along with the economic climate, the evolution of the semiconductor industry is driven by Moore's Law (cf. Moore, 1965). Gordon E. Moore observed in 1965 that the number of

transistors in integrated circuits (ICs) had doubled every two years since its invention in 1958, and he foresaw that the trend would continue. His prediction has proved to be accurate until today. Using Moore's Law as a technological roadmap allowed semiconductor manufacturers to enter a virtuous circle: increased chip complexity leads to a better performance-to-cost ratio that induces a market growth; the realized profits are invested in preparation for the next technological leap, which in turn enables the development of more complex chips.

To avoid an increase of chip prices, the doubling of transistors in ICs must be accompanied by a continuous effort to reduce costs and improve productivity. The following approaches strive for enhancing the relative manufacturing cost per function (cf. ITRS, 2010):

- shrinking the feature sizes of IC's components via geometric downscaling to increase the density of chips on the silicon wafers,
- enlarging the wafer diameter to achieve a higher output of chips per wafer, and
- improving the equipment yield despite the increasing chip complexity.

These levers have successfully improved the relative manufacturing cost per function since the beginning of the semiconductor industry. However, the effort that is required to achieve the next technological stage implies always higher investments; for instance the migration from 300 to 450mm wafer diameter using 22nm technology process is expected to reduce the cost per function by 29%, but the required capital expenditure (CAPEX) to build a new generation fabrication facility is estimated at six to ten billion US dollars (cf. Chien *et al*., 2007). In periods of economic downturn such investments are only affordable for a few of the semiconductor manufacturers. In this context, another stream of research has gained increasing attention in the last decades that takes advantage of the already available equipment. Many researchers and practitioners strive for optimizing the resource utilization by planning and scheduling activities, from machine, to factory, to supply chain levels. It is referred to as operational excellence. It has been identified to offer the most potential for future gains in semiconductor manufacturing (cf. Chien *et al*., 2011).

Having the above as a background, the present thesis focuses on mid-term production planning problems, i.e., master planning (MP), that arise in semiconductor manufacturing. The set of considered problems is denoted as MPSC throughout this document. Given the specifics of semiconductor manufacturing networks, the development of enterprise-wide planning approaches that are computationally tractable and address the uncertainties typically encountered in this industry remains particularly challenging (cf. Chien *et al*., 2011).

1.2 Goals of the Thesis

The purpose of production planning is to allocate limited resources to competing demands over time with respect to often conflicting economic objectives. Depending on the nature of the considered problems, production planning models may require the usage of integer-valued variables; the obtained models are called mixed integer programming (MIP) models. Such MIP models may be difficult to solve in a reasonable amount of time for large-scale instances as usually encountered in enterprise-wide planning environments. Therefore, efficient optimization approaches have to be used to reduce the computational effort while achieving optimal, or near optimal, problem solutions.

- **The first goal of the thesis** is to propose an appropriate mathematical formulation of MPSC problems and to design efficient optimization algorithms to solve MPSC problems with respect to solution quality and computational effort.

The performance of the designed optimization algorithms is assessed, at first, using single problem instances. However, given the uncertainty that is typical for the semiconductor industry, there is a need for incorporating different sources of environment- and system-related disruptions into the evaluation of the planning approaches. For this purpose, a simulation model appears to adequately mimic the stochastic behavior of a semiconductor manufacturing network. In real-world situations, the planning activities occur on a regular basis; it allows for replanning the production plan by taking the current state of the input parameters into account. The investigation of the obtained rolling plans provides further insights into the performance of the planning approach used, which is usually not achievable when considering single problem instances.

- **The second goal of the thesis** is to propose an appropriate simulation-based framework for the performance assessment of MP approaches and to apply the framework to compare two optimization algorithms in a rolling horizon setting while considering demand and execution uncertainties.

Production planning is complicated by the interaction between lead time (for a definition, cf. Subsection 3.1.1) and resource utilization. It is known from queueing theory that the cycle time (for a definition, cf. Subsection 3.1.1) increases nonlinearly with the utilization of the resources. However, the utilization is a result of the release schedule used. This leads to circularity in production planning. On the one hand, the planning approach determines the release schedule based on a prescribed lead time. On the other hand, the cycle time depends on the release schedule. The models presented previously in this thesis assume a fixed product lead time as an exogenous parameter of the planning approach. Although this assumption makes sense for highly aggregated strategic planning problems, it is not desirable for mid-term production planning decisions such as MP. Among other approaches, the iterations between a planning approach that determines the releases of production quantities based on a prescribed lead time and a simulation model that uses these production quantities to calculate cycle time estimates seem to adequately tackle the circularity in production planning.

- **The third goal of the thesis** is to propose an appropriate iterative simulation approach to incorporate load-dependent lead times in MP approaches and to computationally investigate the convergence of the iterative scheme.

1.3 Outline of the Thesis

The present thesis is organized as follows. Chapter 2 presents the specifics of semiconductor industry and introduces the challenges found in fabrication facilities and in manufacturing networks. Chapter 3 provides a description of the setting and analysis of the researched problem on MP approaches. A MIP formulation of MPSC is proposed, and its computational complexity is investigated. Heuristic approaches for solving MPSC are proposed in Chapter 4, and the performance of those heuristics is assessed using single problem instances. In Chapter 5, a simulation-based framework is presented to evaluate solution approaches of MPSC in a rolling horizon setting while considering uncertainty. Two of the proposed heuristics are investigated by using the framework. Chapter 6 tackles the circularity in production planning. An iterative simulation scheme is suggested to incorporate load-dependent lead times in MPSC approaches. The convergence of the scheme is computationally analyzed. Chapter 7 summarizes the results presented in this thesis and gives directions for future research.

2. Challenges in Semiconductor Manufacturing

Semiconductor manufacturing, especially wafer fabrication, is widely considered to be among the most complex of all manufacturing environments (cf., among others, Atherton and Atherton, 1995). This chapter describes the challenges encountered in semiconductor industry.

The present chapter is organized as follows. First, the focus is put on the difficulties found in fabrication facilities. After introducing the main stages of semiconductor manufacturing, the wafer fabrication process is presented in more detail, and the specifics that complicate the control of operations within wafer fabrication facilities (short: wafer fabs) are discussed. Then, the scope is enlarged to challenges found in manufacturing networks. The paradigm of global semiconductor supply chain is introduced, which comes along with the need for operational excellence in order to stay competitive on the market. This chapter concludes with the description of approaches to manage semiconductor manufacturing networks, i.e., supply chain management and advanced planning systems.

2.1 Challenges in Semiconductor Manufacturing Facilities

2.1.1 Overview of Semiconductor Manufacturing

A semiconductor device is a highly miniaturized, integrated electronic circuit that consists of millions of transistors on only a few square centimeters. The sequence of operations that is required to produce the electronic chips can be divided in five major steps: wafer preparation, wafer fabrication, wafer probe, chip assembly, and final chip test. Since this thesis focuses on the application of planning approaches in the second step, i.e., wafer fabrication, the other steps will only be briefly described. Figure 2.1 shows the main semiconductor manufacturing steps starting with raw wafers and ending with final chips.

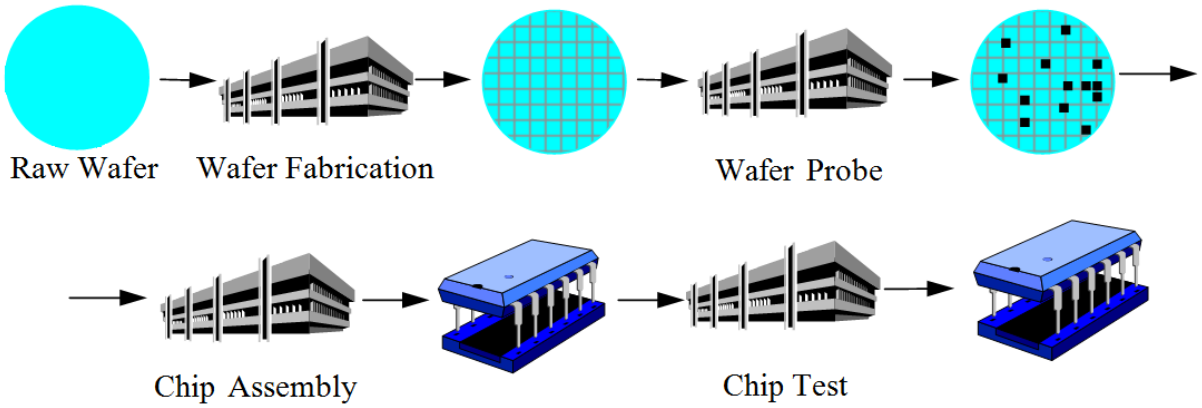

Figure 2.1: Overview of semiconductor manufacturing (cf. Schömig and Fowler, 2000).

Wafer preparation consists of the processes required to obtain raw wafers out of semiconductor materials. First, a cylindrical ingot of highly pure silicon crystal is created after melting and pulling operations. Then, the ingot is sawed in round, thin slices to form the wafers. In mass production, the wafer thickness typically ranges from 200 to 300µm, and the wafer diameter from 150 to 300mm. Some semiconductor manufacturers plan to ramp up the processing of 450mm large wafers from 2013 onwards (cf. ITRS, 2011). In order to prepare the raw wafers for the next manufacturing step, additional surface finishing operations are needed such as polishing and edge grinding. Wafer preparation operations are usually performed by raw material suppliers.

During wafer fabrication, the surface of the wafers is modified to create patterns of integrated electronic circuits. A single wafer contains a large number of (usually) identical chips. More details on wafer fabrication will be presented in the next subsection.

After the fabrication step, the individual circuits or dies on the wafer pass a series of electrical test patterns to identify functional defects. This process is referred to as wafer probe (also: wafer sort). An electronic map of the wafer is made on the condition of each die. With respect to the chip specifications, the non-passing dies are marked with a small dot of ink in order to discard them at a later stage. Frontend operations are completed after wafer probe. Tested wafers are now stored in a die bank that serves as an intermediate buffer stock.

During chip assembly, the individual dies on the wafer are separated by means of mechanical sawing or laser cutting. The non-marked dies are encapsulated into a plastic or ceramic package that prevents physical damage and corrosion. The die pads are connected to the pins of a so-called lead frame with tiny gold wires. Finally, the package is sealed, and a laser etches the name of the device on the package.

A final chip test is conducted to check the performance of the chip features. The extent of the test program may vary depending on the targeted application of the chip. Faulty chips are discarded, while goods chips are sent to a distribution center before shipment to the customer. For specific market segments such as microcontrollers and -processors, good chips are sorted (or binned) after final test into speed performance categories, e.g., fast, medium, and slow (cf. Denton *et al.*, 2006). The distribution of attributes results from random variables in the manufacturing process. The devices sold under a certain designation must meet the requirements at a minimum, that means, a high-grade chip can be substituted for a chip with a lower grade, for instance if it allows avoiding chip shortage. Backend operations are concluded after final chip test

Depending on the device complexity, the processing of today's ICs may involve up to several hundred single process steps along the entire manufacturing process, and requires up to three months to go through.

2.1.2 Wafer Fabrication Process

Wafer fabrication involves a product-specific sequence of chemical and physical process steps that is repeatedly applied to each layer of circuitry on the wafer. A common set of operations consists of layering, patterning, doping, and heat treatments (cf. Zant, 1996). As an illustration, Figure 2.2 depicts the wafer fabrication steps required to build a metal gate Metal Oxide Semiconductor transistor.

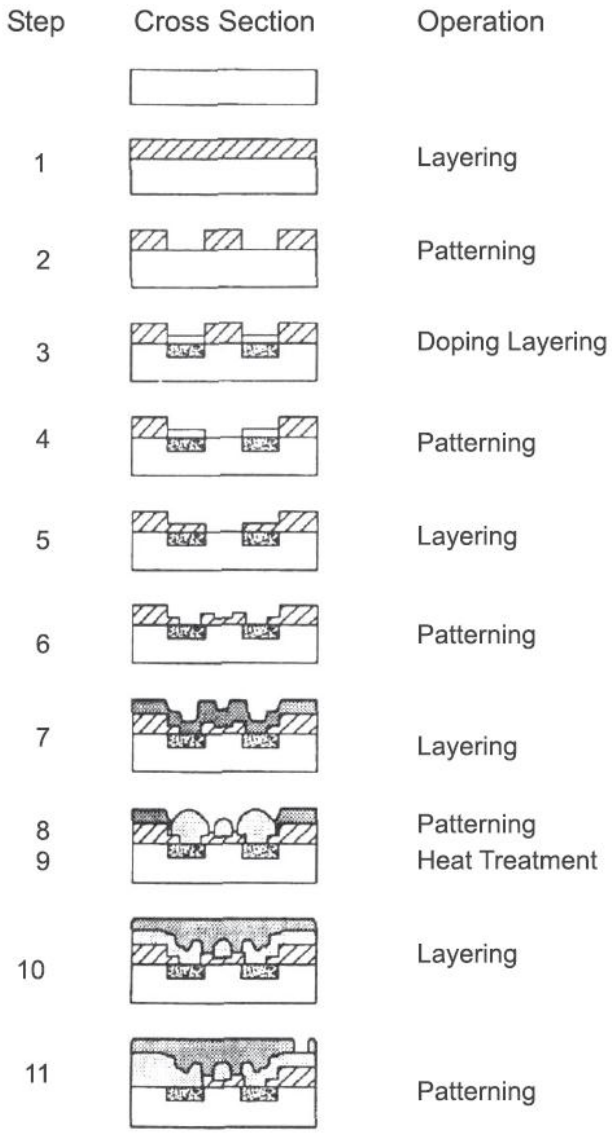

Figure 2.2: Fabrication of a metal gate Metal Oxide Semiconductor transistor (cf. Zant, 1996).

During a layering step, thin layers of an insulating, semiconducting or conducting material are added to the wafer surface (cf. steps 1, 5, 7, and 10 in Figure 2.2). The layers are added using oxidation or deposition. Deposition can be performed in the form of evaporation, chemical vapor deposition (CVD), or physical vapor deposition (PVD) that is also called sputtering. CVD is the most common deposition technique.

In a patterning step, parts of the layer that has been created during the preceding layering step are removed to form the required geometrical structures on the wafer surface (cf. steps 2, 4, 6, 8, and 11 in Figure 2.2). The patterning process is usually referred to as photolithography or masking process. Figure 2.3 provides a simplified illustration of the steps during the photolithography process.

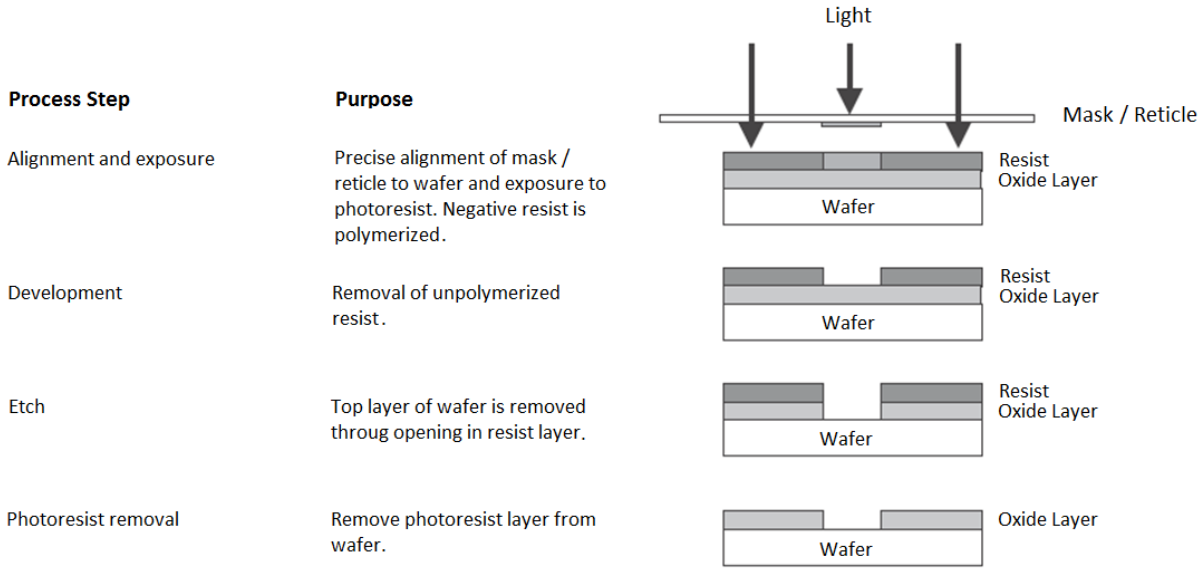

Figure 2.3: Photolithography process steps using negative photoresist (cf. Zant, 1996).

During the photolithography process, the desired horizontal structures are transferred from a pattern plate, i.e., photomask or reticle, to the wafer surface. First, a uniform layer of light-sensitive chemical, i.e., photoresist, covers the wafer surface by spin coating. Then, the surface is exposed to intense light through the photomask (cf. step "Alignment and exposure" in Figure 2.3). The photoresist areas that were not exposed to light can be removed with chemical solvents, i.e., developer, whereas the portions of the photoresist that were exposed to light change their chemical conditions and become resistant to these chemicals (cf. step "Development" in Figure 2.3). Photolithography may be applied using either positive or negative photoresist. Figure 2.3 assumes the usage of negative photoresist. Etchants now remove the parts of the uppermost wafer layer that are not protected by the photoresist (cf. step "Etch" in Figure 2.3). Two techniques are possible: dry etching or wet etching. The final step is to remove the photoresist from the wafer surface (cf. step "Photoresist removal" in Figure 2.3).

To obtain the desired electronic properties on the wafer surface, either thermal diffusion or ion implantation is performed during the doping step. The goal is to create conductive regions and so-called N-P junctions, i.e., separations between regions with a surplus of electrons (N-type) and regions with a surplus of holes (P-type).

The heat treatment step consists of heating the wafer to temperatures of about 500 to 1000 degrees Celsius. This treatment is necessary to repair disruptions of the crystal structure caused for example by ion implantation and is performed either by thermal techniques or using infrared radiation.

2.1.3 Wafer Fabrication Operations

This subsection highlights some of the main factors that make wafer fabrication operations particularly challenging to manage (cf. Uzsoy *et al.*, 1992, 1994; Gupta *et al.*, 2006; Chien *et al.*, 2011; Mönch *et al.*, 2011). A wafer fab is typically organized in work areas. Each area contains several work centers, i.e., set of parallel machines that are specialized in a particular function. Figure 2.4 shows the most common work areas in a wafer fab. With regard to their fabrication process, the wafers travel between the work centers in the form of lots, i.e., groups of wafers contained in carriers or cassettes. The maximum lot size is standardized across the shop-floor, e.g., 25 or 50 wafers per lot. The arrows in Figure 2.4 represent exemplary moves of lots through a wafer fab. At first sight, this factory layout may be assimilated to job-shop or process-oriented production in opposition to flow-shop or product-oriented production. However, it will be demonstrated in the remainder of this subsection that job-shop design does not entirely fit with the specifics of wafer fabrication operations.

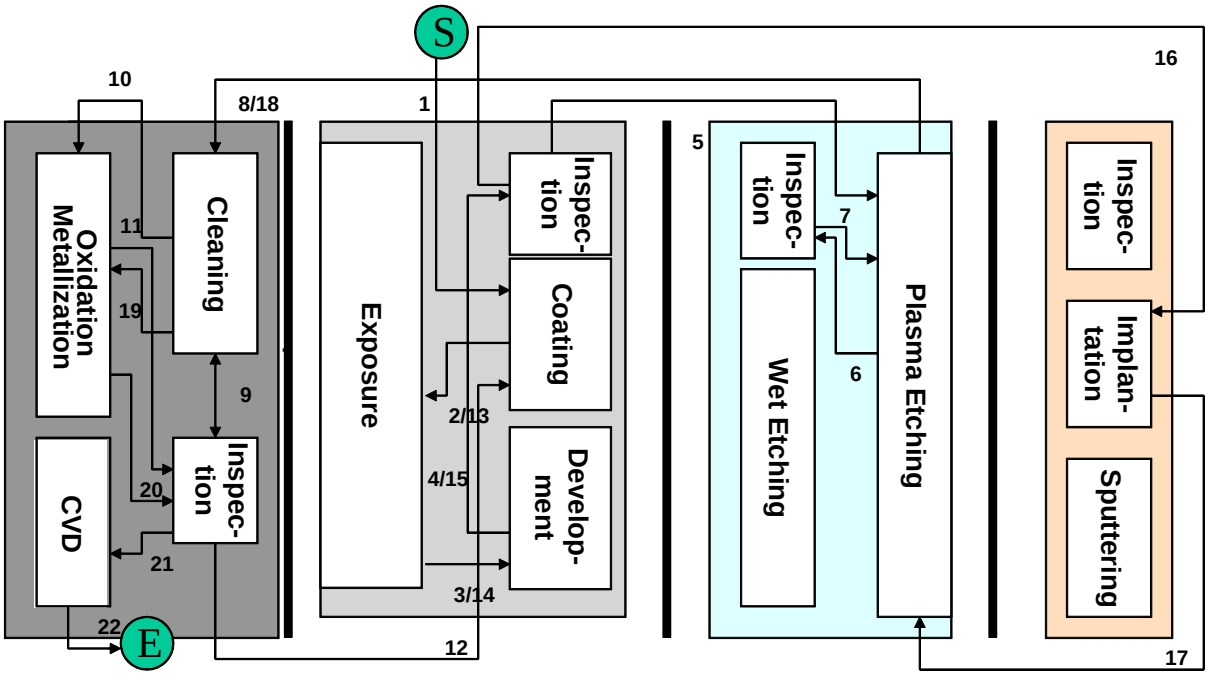

Figure 2.4: Work areas in a wafer fab (cf. Mönch and Gmilkowsky, 2001).

The required CAPEX for a typical wafer fab that is, for instance, designed for 300mm wafer diameter and 65nm technology process is approaching four billion US dollars. In particular, the cost of equipment reaches 75% of the total factory capital cost. It is mainly due to high-precision tools such as steppers for the photolithography that are worth up to eighteen million US dollar per unit (cf. Gupta *et al.*, 2006). The economic necessity to reduce capital spending dictates that an expensive machine is shared by all jobs that require the processing operation provided by the machine. Given the recirculating nature of wafer fabrication as described in Subsection 2.1.2, a wafer must visit the same work centers many times during its manufacturing process. Consequently, these work centers have to deal with competing re-entrant flows of material that may be at different stages of their processing. This characteristic differentiates wafer fabrication operations from the classical job-shop production design. Thus, wafer fabrication requires a dedicated class of manufacturing

system that is called re-entrant lines (cf. Kumar, 1993). Figure 2.5 shows exemplary re-entrant lines in wafer fabrication operations.

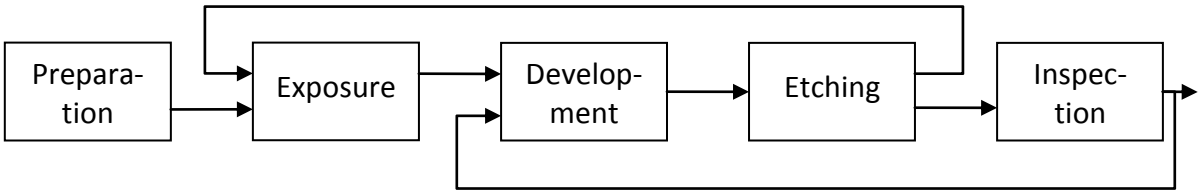

Figure 2.5: Re-entrant lines in wafer fabrication operations (cf. Mönch *et al*., 2011).

Having in mind that the wafer fabrication requires several hundred single process steps, the moves of a lot between the work centers follow very complex flows as schematically depicted in Figure 2.6.

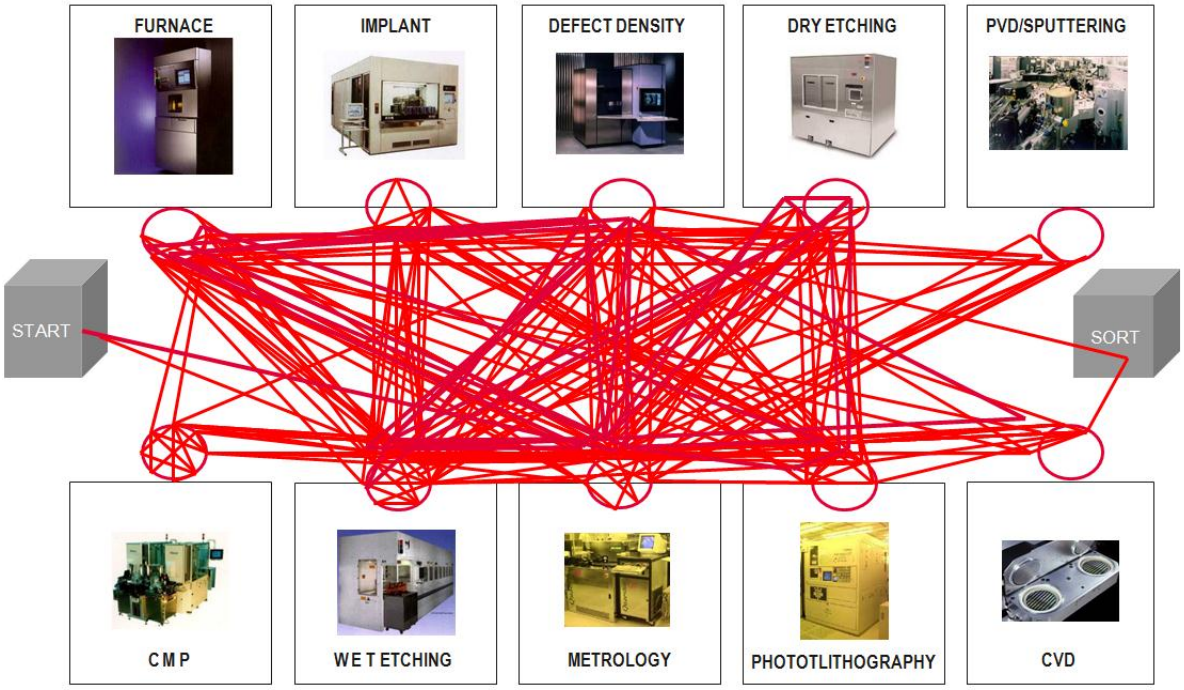

Figure 2.6: Moves of a wafer lot through a wafer fab (cf. Ehm and Ponsignon, 2010).

As a result of the numerous process steps, it takes between eight and twelve weeks, depending on the device complexity, to achieve the processing of a wafer. Obviously the way the competition between the re-entrant lines at so-called bottleneck resources is resolved has a clear impact on the performance measures of the wafer fab.

Further features complicate the control of wafer fabrication operations (cf. Mönch *et al*., 2011). Two of these factors are now briefly described.

- **Heterogeneous operations:** A wafer fab often contains a dozen of different process flows for which the product mix is changing over time. Furthermore, depending on the nature of the operations, the duration of process steps significantly varies between a couple of minutes and several hours. Long operations, which may represent up to one-third of the fab operations, typically involve batch processes wherein multiple lots are processed simultaneously. As lots move through the fab, they are constantly being collected into batches and then dispersed into lots. Hence, batch machines tend to off-

load multiple lots onto machines that are capable of processing only one lot or one wafer at a time. It results in congestion, and thus in the formation of long queues in front of these serial machines. The increased flow variability brings along challenges for the coordination of operations of different natures.

- **Unreliable equipment:** The wafer fabrication process involves sophisticated tools that are subject to unpredictable breakdowns despite all efforts of preventive maintenance (PM). Equipment downtime is believed to be the main source of uncertainty in semiconductor manufacturing operations. It increases the variability of production times and prevents the accurate prediction of completion dates.

2.2 Challenges in Semiconductor Manufacturing Networks

2.2.1 The Global Semiconductor Supply Chain Paradigm

Over the last two decades, the development of information technology as well as the globalization of the world economy had profound effects on how semiconductor devices are produced. One of the most significant paradigm shifts is that semiconductor manufacturers no longer compete as individual companies but rather as supply chains (cf. Chien, 2007). A supply chain is defined as a network of organizations that are involved, through upstream and downstream material, information, and financial flows to produce and deliver goods and services to customers (cf. Stadtler, 2007). Nowadays, semiconductor supply chains are vertically integrated with multiple parallel facilities at each manufacturing stage and several distribution centers. Since the sites are usually dispersed all over the globe, it is referred to global supply chains. Moreover, driven by the increasing product complexity and the pressure on profitability, semiconductor manufacturers tend to focus on their core competencies by following the make-or-buy strategy. Two types of company emerge from this specialization, also called horizontal integration: integrated device manufacturers (IDMs) that design, produce and sell chips, and production partners that perform outsourced production from IDMs or fabless companies. Production partners are named silicon foundries and subcontractors when performing frontend and backend operations, respectively. The resulting modularity of the manufacturing process increases the need for collaboration between the partners of the supply chain (cf. Wu *et al.*, 2012). Figure 2.7 depicts a global semiconductor supply chain from the point of view of an IDM.

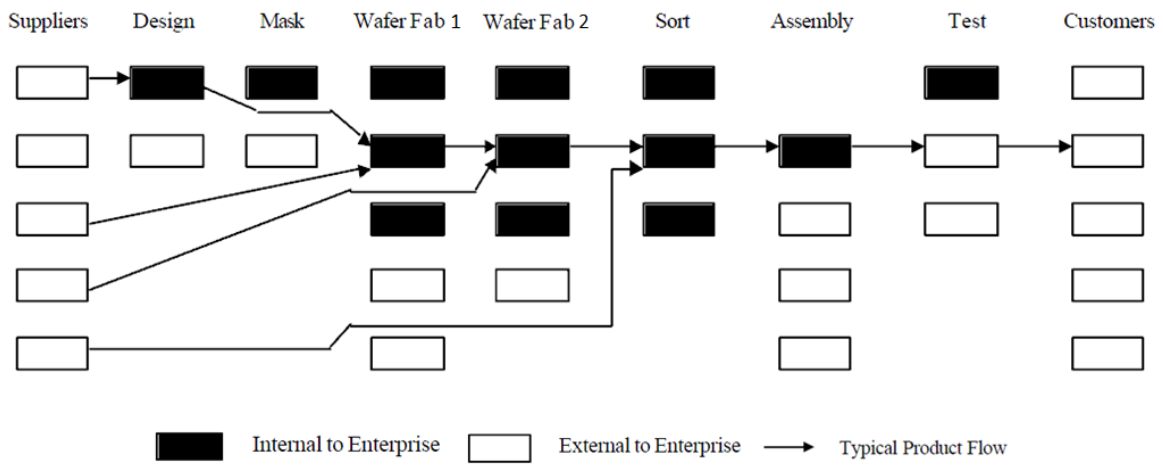

Figure 2.7: Global semiconductor supply chain (cf. Schömig and Fowler, 2000).

Depending on the supply chain configuration, the flows of material between the nodes of a semiconductor manufacturing network may show complex interactions. The sequence of facilities used to process a highly integrated chip for a platform chipset that was commercialized by Infineon Technologies between 2008 and 2010 is taken for illustrative purpose (cf. Ehm *et al*., 2011a). The device was initially manufactured at the only technically qualified facilities: wafer fabrication in Germany, bumping in Taiwan (i.e., a packaging technique also called flip chip), testing back in Germany, assembly in Korea, and final test back in Germany. Whilst the product matured and penetrated the market, it required more capacity and new routing opportunities. Since the manufacturing cost begun to dominate further growth, some parts of the fabrication process were outsourced to production partners. Thus, after a year of booming demand the chip had successively used more than fifteen different supply chains. Each new route allowed either increasing throughput or decreasing cost. It is notable that the cost would have been even lower using one single low cost production site for the entire value chain, but it was technically hardly achievable within the short product life cycle, also the risk would have been too high due to the loss of manufacturing flexibility, and the learning from best practices of neighbor routes would have been lost too. Hence, by harvesting the opportunities of a global manufacturing network, cost reduction of mid-double digit percent ranges has been reached within one year. Figure 2.8 shows the successive supply chains used to produce the chip.

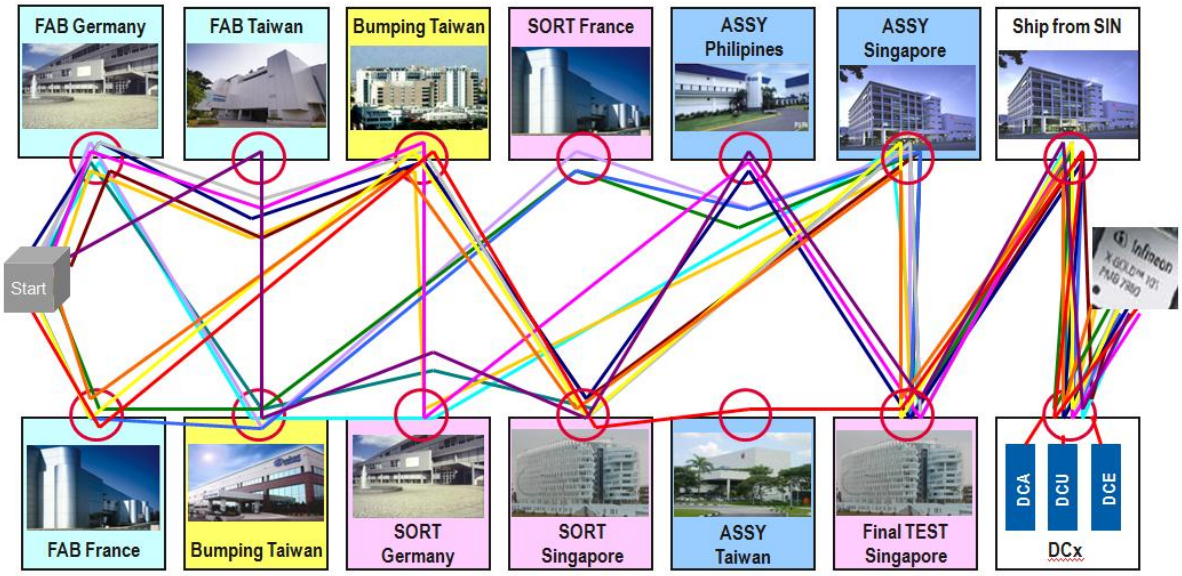

Figure 2.8: Successive supply chains used to produce a highly integrated chip for a platform chipset from Infineon Technologies (cf. Ehm *et al*., 2011a).

2.2.2 Operational Excellence as a Key Competitive Advantage

In the past, the main factor that contributed to the success of a semiconductor company was the design of its semiconductor devices. Nowadays, the fierce competition that takes place between the members of the semiconductor market requires new abilities. Today's major key competitive advantages are:

- quality and reliability of the products,
- short development cycle of new products,
- cost-effective production, and
- ability to meet customers' delivery requirements.

The last two factors are referred to as operational excellence. It will be demonstrated that these two factors come into conflict. On the one hand, semiconductor manufacturers strive for a high utilization rate of available resources because of the costly equipment. On the other hand, the ordering behavior of the customers is uncertain and short-term oriented. Due to high holding cost and risk of obsolescence, keeping extra inventory to prevent shortage is economically not viable. Thus, the decisive factor to cope with challenging delivery due dates is the reduction of production times, i.e., cycle times, that allows for enhanced responsiveness. However, it is known from queuing theory that cycle time increases nonlinearly with capacity utilization. The latter statement can be proved by considering a queuing system that consists of an arrival process, a service, i.e., production process, and a queue (cf. Hopp and Spearman, 1996). The expected cycle time CT that a job spends in such a system is defined as the sum of the mean effective process time t_e and the expected waiting time t_q. Hence, $CT := t_q + t_e$ holds. The rate (or capacity) of the production process is given by $r_e := 1/t_e$. The quantity t_a describes the average time between job arrivals, and the reciprocal value $r_a := 1/t_a$ is the rate of job arrivals. The utilization u of the production process results from the following ratio: $u := r_a / r_e$. The expected waiting time t_q depends on t_e, u, and the variability of process and interarrival time. Therefore, the coefficients of variation of process and interarrival time are used that are defined as $c_e := \sigma_e / t_e$ and $c_a := \sigma_a / t_a$, respectively. The parameters σ_e and σ_a are the standard deviations of the process and interarrival times, respectively. Thus, the approximation of t_q, which was first introduced by Kingman (1961), is given by:

$$t_q := t_e \left(\frac{u}{1-u} \right) \left(\frac{c_e^2 + c_a^2}{2} \right). \tag{2.1}$$

The equation (2.1) can be separated into the capacity term t_e, the utilization term $u/(1-u)$, and the variability term $(c_e^2 + c_a^2)/2$. It is reasonably accurate in the case of a G/G/1 queuing system, i.e., interarrival times and process times are generally distributed, and the production system includes a single machine (cf. Hopp and Spearman, 1996). It turns out that the waiting time, and thus the cycle time, increases with the process time, the utilization, and the variability. Figure 2.9 shows the relationship between utilization and cycle time for two variability levels with $c_{e,1} > c_{e,2}$ and $c_{a,1} > c_{a,2}$. It is often referred to as operating curve. For a given cycle time, a higher utilization $u_2 > u_1$ is reached with a lower variability term. As a result, one of the main drivers to achieve operational excellence consists in mitigating variability in the manufacturing process.

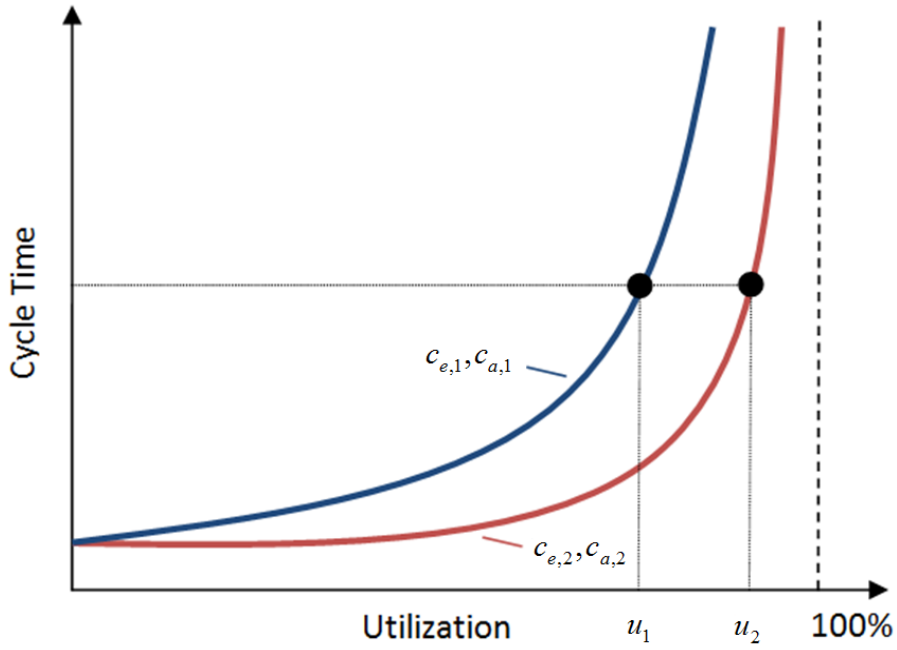

Figure 2.9: Relationship between utilization and cycle time for two variability levels.

The scope of operational excellence goes beyond the frontiers of single fabrication sites. The pressure of effectiveness and speed of execution emphasizes the need for agile, adaptable, and aligned supply chains that is named the Triple-A Challenge (cf. Lee, 2004).

- **Agility** is the ability to quickly respond to uncertainties in the supply chain such as erratic changes in demand and sudden disruptions in supply.
- **Adaptability** is the ability to adjust the supply chain design to accommodate market changes; in semiconductor business it implies trading off conflicting objectives such as short product life cycles versus long production times, long capacity expansion and qualification times versus delivery responsiveness.
- **Alignment** is the ability to balance the interests of multiple members in the supply chain.

2.2.3 Levers for Operational Excellence: Supply Chain Management and Advanced Planning

This subsection presents two complementary levers that have been adopted by many semiconductor companies in quest of operational excellence and Triple-A supply chains, namely supply chain management and advanced planning.

2.2.3.1 Supply Chain Management

Companies have recourse to Supply Chain Management (SCM) to govern material, information, and financial flows within and across the organizations of the supply chain in order to fulfill customer demands with the ultimate aim of improving the competitiveness of the entire supply chain (cf. Stadtler, 2007). The two pillars of SCM consist in the integration of the network and the coordination of the flows. SCM techniques focus on:

- setting, aligning, and propagating supply chain strategies,
- improving supply chain network design,
- ensuring efficient communication,
- fostering inter-organizational collaboration,
- documenting and enhancing supply chain processes, and
- planning supply chain activities.

The Supply Chain Operations Reference (SCOR) model developed by the Supply Chain Council provides a standardized terminology for representing, configuring, and benchmarking supply chains (cf. Supply Chain Council, 2012). The SCOR notation has been adopted by many semiconductor companies to describe their business activities from suppliers' suppliers to customers' customers. It is notable that the reference model addresses neither sales nor marketing nor product development activities. The SCOR model covers three hierarchical levels of processes: process types (Level 1), process categories (Level 2), and process elements (Level 3). In the following paragraphs, selected components from each level will be briefly described. Level 1 includes five elementary process types: Plan, Source, Make, Deliver, and Return.

- **Plan** covers processes related to resources and requirements; it establishes and communicates plans across the supply chain; it measures supply chain performance; and it manages assets.
- **Source** manages supplier network; it receives materials; and it manages inventories of materials.
- **Make** transforms materials into finished products; and it manages work in progress (WIP), equipment, and facilities.
- **Deliver** manages distribution network; it issues finished products; and it manages inventories of finished products.
- **Return** takes care of defective and excess products; and it manages post-delivery customer services.

Figure 2.10 shows the five elementary process types in Level 1 of the SCOR model on the example of a five echelon supply chain.

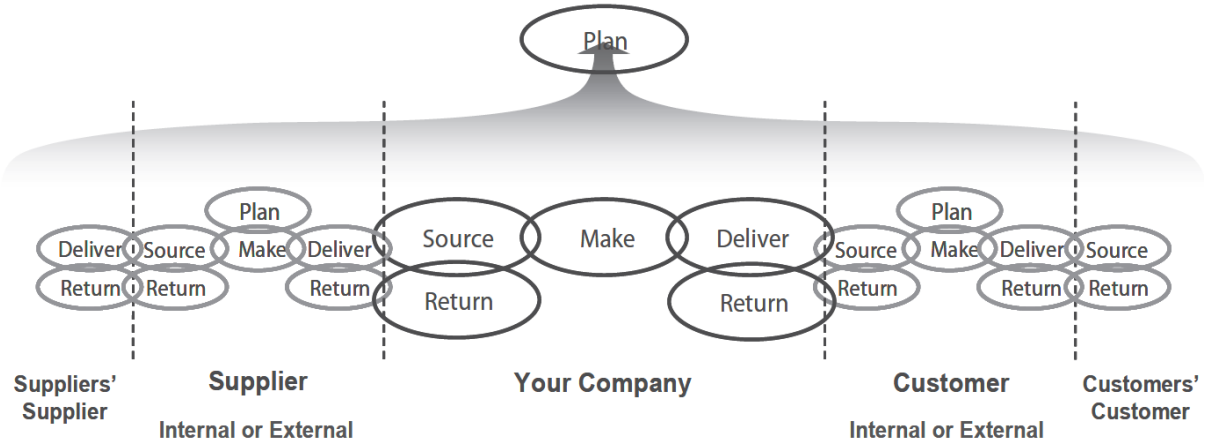

Figure 2.10: Level 1 of the SCOR model (cf. Supply Chain Council, 2012).

Since the present thesis concentrates in planning approaches, Plan is depicted in more detail. Level 2 of Plan encompasses, among others, the process category Plan Supply Chain that supports the allocation of resources to requirements in adequate planning horizons. In Level 3 of the SCOR model, Plan Supply Chain is composed of four process elements that are defined as follows:

- identify, prioritize, and aggregate supply chain requirements (P1.1),
- identify, prioritize, and aggregate supply chain resources (P1.2),
- balance supply chain requirements with supply chain resources (P1.3), and
- establish and communicate supply chain plans (P1.4).

Figure 2.11 shows the process elements of Plan Supply Chain as well as input and output streams as stated in Level 3 of the SCOR model. The process elements are decomposed into company-specific practices on the fourth level that is not in the scope of the SCOR model.

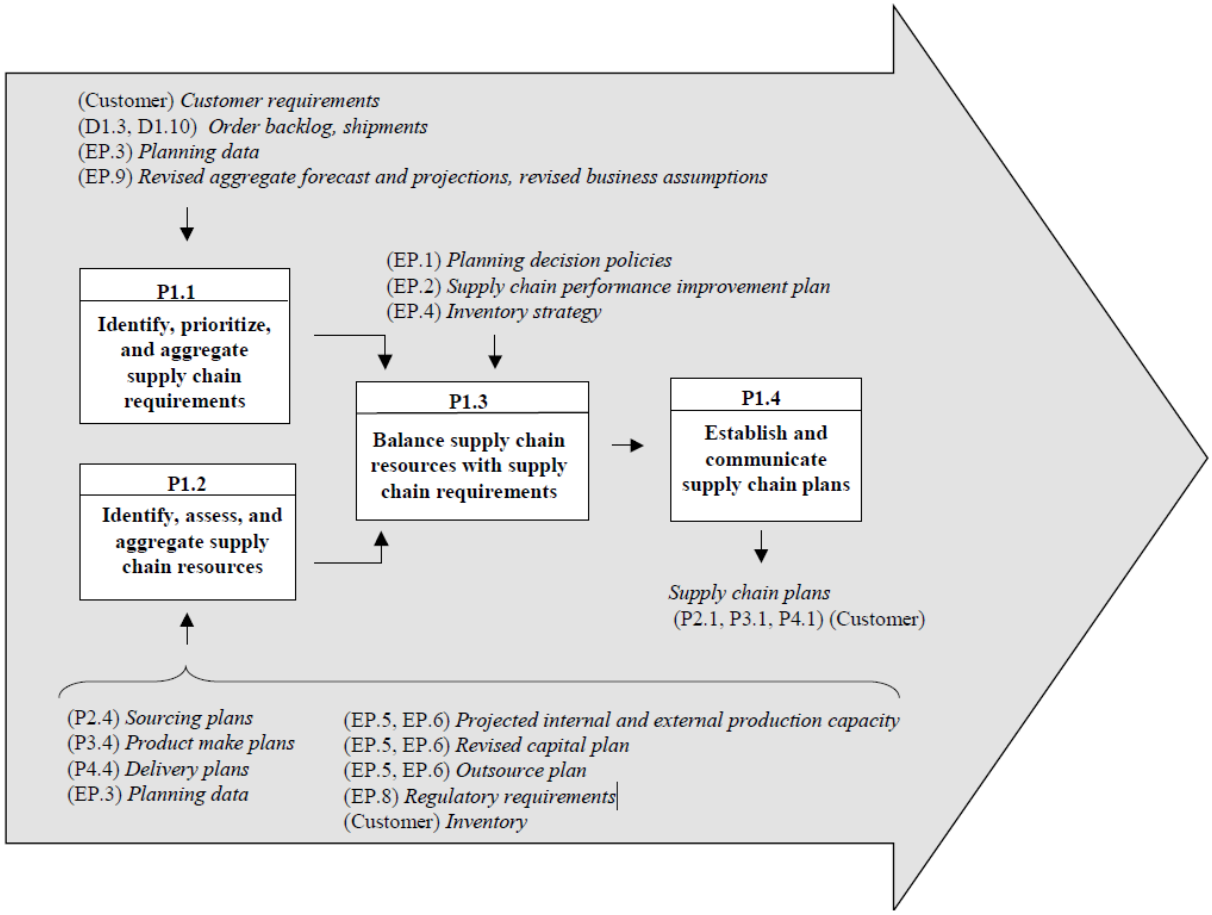

Figure 2.11: Process elements of Plan Supply Chain in Level 3 of the SCOR model (cf. Supply Chain Council, 2012).

2.2.3.2 Advanced Planning

The SCOR model proposes a framework of supply chain activities, but it does not state how they have to be accomplished. The advanced planning approach fills in this gap. An Advanced Planning System (APS) is an application system used for supply chain planning. APS have been introduced to cope with the deficits of the planning approaches that are implemented in most of the Enterprise Resource Planning (ERP) systems (cf. Tempelmeier, 2001). The main features of APS will be presented in the remainder of this subsection. Figure 2.12 describes the structure of planning tasks of APS. It is referred to as Supply Chain Planning Matrix (SCPM) (cf. Fleischmann *et al*., 2007).

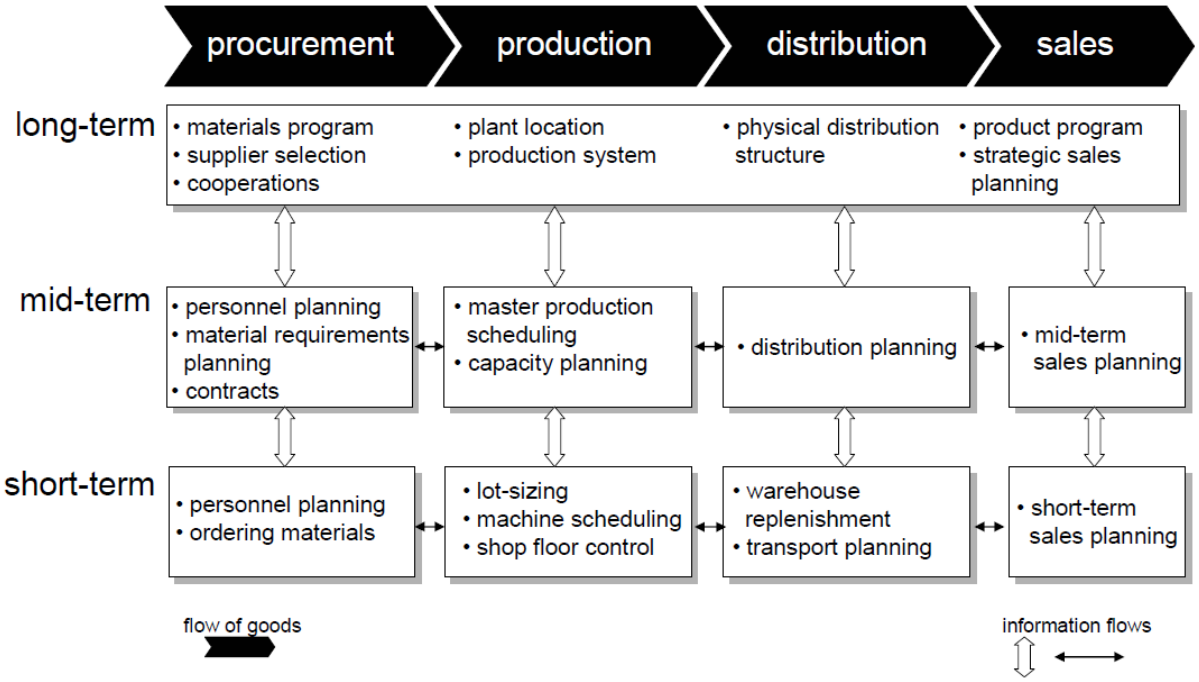

Figure 2.12: The supply chain planning matrix (cf. Fleischmann *et al.*, 2007).

The major characteristics of an APS are the following:

- integral or global planning,
- optimization focus, and
- hierarchical approach.

The integral planning implies spatial integration of the tasks as well as functional integration of the primary activities, i.e., procurement, production, distribution (in the SCOR model: Source, Make, Deliver), and sales. The inter-temporal integration (also: hierarchical planning) leads to decomposed planning activities. APS uses transactional data that are gathered, for instance, in an ERP system. A crucial factor for the success of APS is the usage of appropriate aggregation and disaggregation procedures for horizontal and vertical information flows (e.g., product, resource, and time information). The different levels of SCPM distinguish between strategic, tactical, and operational planning activities. The characteristics of each level are now described.

- **Strategic planning** deals with the management of change in the production process and the acquisition of resources over long-term horizons that range from two to five years using highly aggregated data. The frequency of strategic planning is quarterly or yearly.
- **Tactical planning** focuses on resource allocation and utilization problems over mid-term horizons that range from six months to two years using aggregated data. The frequency of tactical planning is weekly or monthly.
- **Operational planning** aims at planning and controlling the execution of production tasks over short-term horizons that range from a couple of production shifts to three months using detailed data. The frequency of operational planning is daily or weekly.

APS are decision support systems that help to visualize information, reduce planning effort, and enable application of optimization techniques. However, modeling is always a relaxation of reality. Hence, human knowledge, experience, and skill are required to fine-tune plans that result from APS.

Selected planning tasks of SCPM are now presented in more detail.

- **Capacity planning** is the process of determining the amount of resources that is required to meet firm customer orders and forecast demand over a mid-term horizon of usually one and a half year to two years expressed in months by considering aggregated data. Capacity planning can be divided into a strategic decision problem that leads to one or more tactical decision problems. The strategic problem decides which type, how much, and when should capacity be used, added, and removed. The resulting strategic plan refers to allocated capacities and investment decisions. The aim of the tactical problem is then to maximize the capacity utilization subject to the constraints imposed by the strategic plan. The resulting tactical plan refers to production starts and product mix decisions (cf., among others, Bermon and Hood, 1999).
- **Master planning (MP) or master production scheduling (MPS)** consists in planning the production of end-products in a manufacturing network to meet firm customer orders and forecast demand over a mid-term horizon of usually six months expressed in weeks, taking into account capacity utilization and aggregated inventory levels. The outcomes of MP are capacitated production requests. It corresponds to the process element P1.3 in Level 3 of the SCOR model. The balancing of requirements with resources is sometimes referred to as demand-supply match process (cf. Kallrath and Maindl, 2006). Some researchers distinguish between MPS and MP by arguing that MP focuses on rather tactical decisions such as capacity utilization, while MPS takes care of rather operational decisions based on MP decisions such as production planning (cf. Pochet and Wolsey, 2006). However, this distinction does not seem to be reflected in all APS. Hence, MPS is assimilated to MP throughout this thesis.
- **Material requirements planning (MRP)** establishes mid-term procurement and production plans based on MP decisions for all components, i.e., from raw materials to purchase, to intermediate products to process, to final products to sell, according to product structure information, i.e., bill of materials, in order to satisfy external customer demand. MRP uses a decomposition approach into uncapacitated single-item subproblems that are solved independently and sequentially. Several major drawbacks were identified in the MRP concept that lead to productivity and flexibility losses, among others, capacity infeasibility, long planned lead times, and system nervousness (cf. Hopp and Spearman, 1996). Eventually, MRP evolved to a larger construct known as MRP-II, i.e., manufacturing resources planning. A main feature of MRP-II is the consideration of other aspects than production and procurement such as demand management, forecasting, capacity planning, MP, and production control. However, the capacity check that is implemented in MRP-II, known as capacity requirements planning (CRP), does not perform a finite capacity analysis. Instead, CRP predicts the lot completion times for each process center using given fixed lead times. A predicted loading is then computed over time, but it is not adjusted in situations of overloading (cf. Hopp and Spearman, 1996). As a first step towards integral planning, MRP-II systems are designed around a centrally held database that is the cornerstone for the implementation of so-called Manufacturing Planning and Control (MPC) systems. Despite the lack of optimization taking finite capacity into account in MPC, planning approaches in most of today's ERP systems use algorithms inherited from the MRP-II approach. It is one of the reasons for the development of optimization-focused planning approaches as found in APS (cf. Tempelmeier, 2001).

- **Lot-sizing and machine scheduling** comprise the determination of lot sizes and the sequencing of the lots on the machines of single production sites. Lot-sizing has to balance costs of setups and stock holding with respect to dependencies between the products. Lot-sizing is a part of the MRP approach. As a result of a lot-machine assignment, a production schedule is obtained. The common objectives for scheduling decisions are to maintain due date integrity, keep machine utilization high, and achieve low manufacturing times. The considered horizon is typically short. Manufacturing environments with prevalent machine breakdowns require frequent rescheduling. Machine scheduling is a part of the MRP-II approach.

Capacity planning and MP are challenging planning activities in global semiconductor supply chains, and lot-sizing and machine scheduling are particularly difficult in wafer fabs due to the specifics presented in Subsection 2.1.3. The planning of material procurement is of minor importance for frontend operations, since the procurement of raw materials (e.g., raw wafers, gases…) is rarely critical.

2.3 Conclusion

This chapter introduced the features that make semiconductor fabrication one of today's most complex manufacturing environments. The operational excellence challenges that complicate the control of wafer fabs and the planning of global semiconductor supply chains have been presented. MP decisions are the cornerstone for balancing market requirements with supply chain resources. The next chapter addresses MP problems that arise in networks of wafer fabs.

3. Problem Setting and Analysis

In the context of enterprise-wide planning, decisions have to be made that involve conflicting economic goals, e.g., the customer service level ought to be as high as possible while the inventory level has to be as small as possible. A common way to handle multiple objectives consists in pricing the goals monetarily by revenues and costs and maximizing the resulting marginal profit (cf. Pochet and Wolsey, 2006). This approach will be applied to MP decisions for networks of wafer fabs in this chapter.

The present chapter is organized as follows. First, the problem of interest is described, and the necessary notation is introduced. Then, a mathematical formulation of the researched problem is presented. The computational complexity of the problem is investigated. Finally, the related literature is discussed.

3.1 Master Planning in Semiconductor Manufacturing (MPSC)

3.1.1 Problem Description

The problem of interest consists in determining appropriate wafer production quantities for several products in several parallel wafer fabs over a mid-term planning horizon with regard to competing demands and limited capacity. As showed in Figure 3.1, the fabrication sites constitute a one-layer manufacturing network with no flow of material across the wafer fabs. The facilities may be either in-house locations or silicon foundries. The processed wafers are kept in a centralized inventory prior to the next production step.

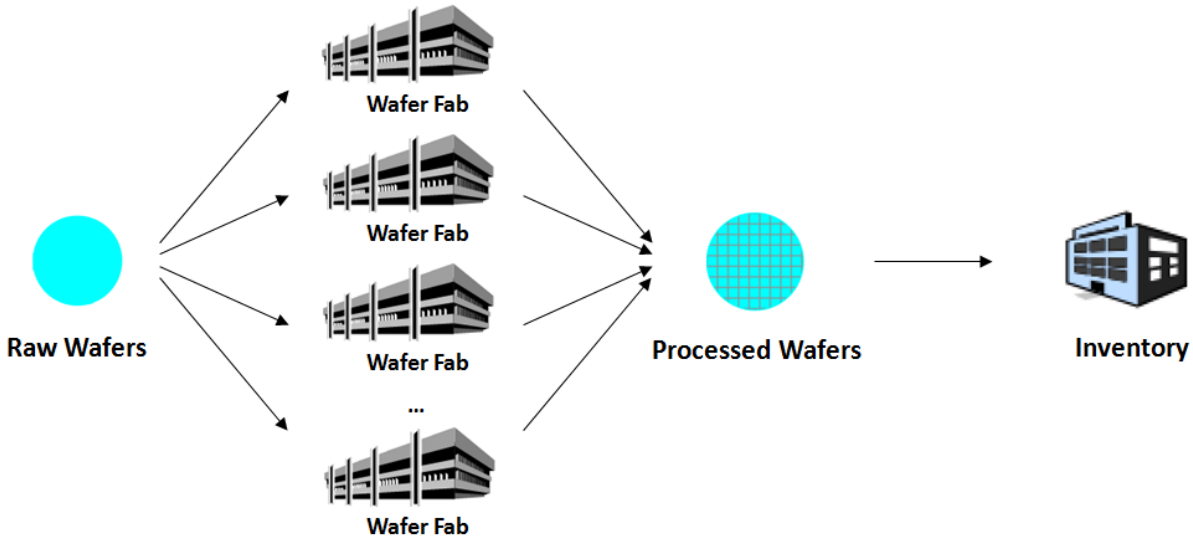

Figure 3.1: One-layer manufacturing network of wafer fabs.

The master plan typically has a horizon of six months divided into weekly time buckets. Since market demand is not entirely known when planning a couple of weeks or months ahead, it is distinguished between firm customer orders and supply reservations. A firm customer order is a customer demand that has already been confirmed by the order management system of the company based on the available supply, i.e., process P1.4 in Figure 2.11. It is a binding request that can be modified or cancelled by the customer under certain conditions only. A supply reservation is an additional forecast demand that comes either from the demand planning process of the company, i.e., process P1.1 in Figure 2.11, or directly from the customer. A customer forecast is a non-binding demand that can usually be modified or cancelled by the customer without any restriction. Hence, a supply reservation is used as a placeholder for orders that may arrive at a later stage (cf. Vieira, 2006). Given the difference in the certainty, the demand elements are prioritized as follows: orders are fulfilled at first, and supply reservations are satisfied when capacity is available. It is assumed that unmet firm customer orders are carried over as backlog to the next planning period. On the contrary, unfulfilled supply reservations are ignored for the rest of the planning horizon.

The cycle time of a product refers to the time that is required on average to complete orders of this product in the production system. It is also called flow time. The lead time of a product is an estimate of the cycle time in the planning system. The problem setting assumes that all products have the same fixed lead time.

The capacity modeling approach is a crucial point for MP decisions. In the present problem setting, the capacity limits are related to bottleneck work centers. For each of them, minimum and maximum loading bounds are considered. The minimum threshold ensures that facilities are used even if demand is low to avoid ramp up effects (cf. Gupta *et al.*, 2006). Given the completion period of a wafer and its fabrication process, it is possible to compute when it arrives at a certain bottleneck work center. Next, the time that the product spends being processed on the machines of the bottleneck resource within a specific period is accumulated. Finally, the sum of machine-hours considering all wafers has to be contained within the capacity limits of the bottleneck work center. This method allows taking re-entrant flows of material into account (cf. Subsection 2.1.3). The proposed capacity model is similar to other approaches used for capacity planning in semiconductor manufacturing (cf. Bermon and Hood, 1999; Barahona *et al.*, 2005). However, this representation is not

appropriate for silicon foundries because information on bottleneck resources is not always available outside the company. In this case, the capacity limits are converted from hours to pieces, and the number of outsourced wafers is considered. In the problem setting, the capacity limits are time-dependent as the capacity allocations that result from the capacity planning process (cf. Subsection 2.2.3.2) may not be constant over time.

It is also assumed that all products can be processed in every wafer fab of the network. However, the model includes decisions related to the number of facilities used to fulfill the demand for one product in one period. A master plan that spreads production requests for one single product all over the manufacturing network is not economically viable due to transport cost and logistic issues. Therefore, fixed production costs, so-called location costs, are incorporated in the model to minimize production partitioning.

Finally, the decision problem contains an objective function related to the difference between revenues and total costs, i.e., marginal profit, and it is subject to a set of constraints. The objective function strives to keep the number of unmet firm customer orders low and to satisfy supply reservations if capacity is sufficient, whereas a low inventory level is of interest. The production can be outsourced to production partners, but an inexpensive assignment of the products to in-house locations and silicon foundries is privileged. The production partitioning over different facilities is limited with respect to fixed production costs.

3.1.2 Notation

A set of products $P := \{p_1,..., p_{\max}\}$ is considered that can be processed in $m_{\max}$ wafer fabs. The manufacturing network consists of $ih_{\max}$ in-house locations and $sf_{\max}$ silicon foundries, i.e., $m_{\max} = ih_{\max} + sf_{\max}$. The total number of bottleneck work centers associated with the facilities is represented by $b_{\max}$. Each bottleneck is assigned to exactly one facility, and each facility has at least one bottleneck. This assumption is reasonable because planned bottleneck resources exist in all wafer fabs due to the re-entrant flows and the expensive machines. The number of bottlenecks in facility m is denoted by $b_{m,\max}$. Clearly, $\sum_{m=1}^{m_{\max}} b_{m,\max} = b_{\max}$ holds. In the case of silicon foundries, one single bottleneck is modeled, i.e., $b_{m,\max} = 1$. The quantity $t_{\max}$ stands for the planning horizon measured in periods. The length of a single time bucket is one week. The index $k = 0,...,k_{\max}$ is used to calculate the capacity consumption of the products. The products have the same lead time of $q = k_{\max} + 1$ periods. A capacity consumption matrix $\widetilde{C}_{pm} \in \mathbb{R}^{b_{m,\max} \times q}$ is used for each combination of product $p \in P$ and facility m, where the elements $cc_{bk}^{pm} \geq 0$ of $\widetilde{C}_{pm}$ model the capacity consumption of one wafer of product p when it is processed on the machines of the bottleneck work center $1 \leq b \leq b_{m,\max}$ in facility m and its completion period is $0 \leq k \leq k_{\max}$ periods ahead. In the following, the decision variables and parameters of the model are introduced.

The decision variables of the model are:

B_{pt} : backlog of firm customer orders of product *p* at the end of period *t*,

I_{pt} : inventory level of product *p* at the end of period *t*,

s_{pt}^{fo} : sales quantity of firm customer orders of product *p* in period *t*,

s_{pt}^{sr} : sales quantity of supply reservations of product *p* in period *t*,

u_{pmt} : indicator variable for the occurrence of fixed production cost of product *p* in facility *m* in period *t*,

$$u_{pmt} := \begin{cases} 1 & \text{if product } p \text{ is processed in facility } m \text{ in period } t, \\ 0 & \text{otherwise,} \end{cases}$$

x_{pmt} : number of wafers of product *p* to be completed at the end of period *t* in facility *m*.

It is notable that $B_{pt}, I_{pt}, s_{pt}^{sr}, s_{pt}^{fo}, x_{pmt} \in \mathbb{R}_+$, whereas $u_{pmt} \in \{0,1\}$.

The parameters used in the model are:

B_{p0} : initial backlog of product *p* at the beginning of the planning horizon,

$C_{mbt}^{\max}$: maximum capacity of bottleneck *b* in facility *m* in period *t* (in hours or pieces),

$C_{mbt}^{\min}$: minimum utilization of bottleneck *b* in facility *m* in period *t* (in hours or pieces),

cc_{bk}^{pm} : capacity consumption of one wafer of product *p* when it is processed in facility *m* at bottleneck *b* and its completion period is *k* periods ahead,

d_{pt}^{fo} : firm customer orders for product *p* at the end of period *t*,

d_{pt}^{sr} : supply reservations for product *p* at the end of period *t*,

hc_{pt} : inventory cost for holding one wafer of product *p* within period *t*,

I_{p0} : initial inventory level of product *p* at the beginning of the planning horizon,

lc_{pmt} : location cost when product *p* is processed in facility *m* in period *t*, i.e., fixed cost,

mc_{pmt} : cost to produce one wafer of product *p* in facility *m* in period *t,* i.e., variable cost,

rev_{pt} : expected revenue per wafer for satisfying supply reservations of product *p* in period *t*,

udc_{pt} : cost due to unmet firm customer orders, i.e., backlog cost, for one wafer of product *p* postponed from period *t* to period *t+1*,

x_{pmt}^{i} : initial number of wafers of product *p* to be completed at the end of period *t* in facility *m*, i.e., WIP started before the beginning of the planning horizon, and

δ : large number, i.e., $\delta \geq \max_{m,b,t} \left\{ C_{mbt}^{\max} \Big/ \min_{p} \left\{ \sum_{k=0}^{\min(k_{\max}, t_{\max}-t)} cc_{bk}^{pm} \right\} \right\}$.

3.2 Mixed Integer Programming (MIP) Formulation of MPSC

This section proposes a MIP formulation for solving MPSC problems. The MIP model is given by the objective function (3.1) and the constraints (3.2)-(3.8).

Maximize the objective function f :

$$f\left(x,u,I,B,s^{sr}\right) := \sum_{p=1}^{p_{\max}}\sum_{t=1}^{t_{\max}}\left(rev_{pt}s_{pt}^{sr} - hc_{pt}I_{pt} - udc_{pt}B_{pt} - \sum_{m=1}^{m_{\max}} mc_{pmt}x_{pmt} - \sum_{m=1}^{m_{\max}} lc_{pmt}u_{pmt}\right), \quad (3.1)$$

subject to the following constraints:

$$I_{pt} = I_{pt-1} - s_{pt}^{fo} - s_{pt}^{sr} + \sum_{m=1}^{m_{\max}} x_{pmt} + \sum_{m=1}^{m_{\max}} x_{pmt}^{i}, \qquad \forall p = 1,..,p_{\max}, \forall t = 1,..,t_{\max}, \quad (3.2)$$

$$s_{pt}^{fo} + B_{pt} = d_{pt}^{fo} + B_{pt-1}, \qquad \forall p = 1,..,p_{\max}, \forall t = 1,..,t_{\max}, \quad (3.3)$$

$$s_{pt}^{sr} \le d_{pt}^{sr}, \qquad \forall p = 1,..,p_{\max}, \forall t = 1,..,t_{\max}, \quad (3.4)$$

$$C_{mbt}^{\min} \le \sum_{p=1}^{p_{\max}} \sum_{k=0}^{\min\left(k_{\max},t_{\max}-t\right)} cc_{bk}^{pm}\left(x_{p,m,t+k} + x_{p,m,t+k}^{i}\right) \le C_{mbt}^{\max}, \qquad \forall m = 1,..,m_{\max}, \forall b = 1,..,b_{m,\max}, \forall t = 1,..,t_{\max}, \quad (3.5)$$

$$x_{pmt} \le \delta \cdot u_{pmt}, \qquad \forall p = 1,..,p_{\max}, \forall m = 1,..,m_{\max}, \forall t = 1,..,t_{\max}, \quad (3.6)$$

$$x_{pmt} \ge 0, s_{pt}^{fo} \ge 0, s_{pt}^{sr} \ge 0, I_{pt} \ge 0, B_{pt} \ge 0, \qquad \forall p = 1,..,p_{\max}, \forall m = 1,..,m_{\max}, \forall t = 1,..,t_{\max}, \quad (3.7)$$

$$u_{pmt} \in \{0,1\}, \qquad \forall p = 1,..,p_{\max}, \forall m = 1,..,m_{\max}, \forall t = 1,..,t_{\max}. \quad (3.8)$$

The objective consists in maximizing the overall difference between the revenues and the sum of costs. The first term in the objective function f models the revenues for fulfilling supply reservations. The costs for holding inventory are modeled by the second term. The third term refers to penalty costs for backlogged firm customer orders. The fourth and fifth terms represent variable and fixed production costs, respectively.

Constraint (3.2) represents the flow balance in every period and for every product. The inflows are the initial inventory, the production quantities, and the WIP; the outflows are the sales quantities related to firm customer orders and supply reservations, and the ending inventory. Constraints (3.3) and (3.4) relate sales quantities to market demand. Backlog is allowed only for firm customer orders. In case of supply reservations, a maximum bound is considered. The capacity restrictions for every bottleneck work center in each period are defined in constraint (3.5) with minimum and maximum utilization limits. The overall loading is calculated by taking the production quantities and the WIP of all products into account. It is assumed that $\sum_{p=1}^{p_{\max}} cc_{b0}^{pm} > 0$ to ensure that there is at least one product p such that $\sum_{k=0}^{\min\left(k_{\max},t_{\max}-t\right)} cc_{bk}^{pm} > 0$, $\forall m = 1,..,m_{\max}, \forall b = 1,..,b_{m,\max}, \forall t = 1,..,t_{\max}$. Inequality (3.6) fixes the

binary variable to 1 whenever there is a positive production for the considered product, wafer fab, and time period. On the other hand, $u_{pmt} = 0$ leads to $x_{pmt} = 0$. It makes sure that an additional facility is used only when it is necessary. Non-negativity and binary conditions are defined by constraints (3.7) and (3.8), respectively.

3.3 Computational Complexity of MPSC

In this section, the computational complexity of MPSC is investigated. Therefore, the statement in Proposition 1 will be proved.

Proposition 1: MPSC is NP-hard.

Proof: Proposition 1 is proved by reduction by considering a special case of the original problem.

First, MPSC is changed to a single-product, single-facility, multi-period problem:

$$p_{\max} = 1,\ m_{\max} = 1,\ t_{\max} > 1. \tag{3.9}$$

The indices related to products and facilities are eliminated from f. It leads to the following formulation:

$$\text{Maximize} \sum_{t=1}^{t_{\max}} \left\{ rev_t s_t^{sr} - hc_t I_t - mc_t x_t - lc_t u_t - udc_t B_t \right\}. \tag{3.10}$$

Revenues and inventory holding costs are assumed equal to zero except for non-zero holding costs in the last period:

$$rev_t = 0,\ \forall t = 1,..,t_{\max}, \tag{3.11}$$

$$hc_t = 0,\ \forall t = 1,..,t_{\max} - 1, \tag{3.12}$$

$$hc_{t_{\max}} > 0. \tag{3.13}$$

Thus, the following minimization formulation is obtained that is a special case of the problem given by (3.1)-(3.8):

$$\text{Minimize} \sum_{t=1}^{t_{\max}} \left\{ hc_t I_t + mc_t x_t + lc_t u_t + udc_t B_t \right\}. \tag{3.14}$$

Then, the capacity constraint (3.5) is modified to a single-product, single-facility, multi-period problem. Also, the manufacturing process is assumed to consist of only one bottleneck work center, and the product lead time q is set to one period:

$$b_{1,\max} = 1, \tag{3.15}$$

$$q = 1. \tag{3.16}$$

It is assumed per definition that $q = k_{\max} + 1$. Hence, the following setting holds:

$$k_{\max} = 0. \tag{3.17}$$

The parameters related to the minimum capacity bound, the WIP, and the capacity consumption are set as follows:

$$C_t^{\min} = 0,\ \forall t = 1,..,t_{\max}, \tag{3.18}$$
$$x_t^i = 0,\ \forall t = 1,..,t_{\max}, \tag{3.19}$$
$$cc = 1. \tag{3.20}$$

Given the settings described above, the capacity constraint (3.5) can be formulated as:

$$x_t \le C_t^{\max},\ \forall t = 1,..,t_{\max}. \tag{3.21}$$

Moreover, it is assumed that there is no supply reservation as well as no initial inventory and backlog:

$$d_t^{sr} = 0,\ \forall t = 1,..,t_{\max}, \tag{3.22}$$
$$I_0 = B_0 = 0. \tag{3.23}$$

Then, the backlog costs are considered constant and higher than any other costs:

$$udc_t \equiv udc,\ \forall t = 1,\ldots,t_{\max}, \tag{3.24}$$
$$lc_t < udc,\ \forall t = 1,\ldots,t_{\max}, \tag{3.25}$$
$$mc_t < udc,\ \forall t = 1,\ldots,t_{\max}. \tag{3.26}$$

Also, in the last period the following assumption holds:

$$mc_{t_{\max}} < hc_{t_{\max}}. \tag{3.27}$$

Next, the following decision version of the knapsack problem (KSP) is considered that is known to be NP-complete (cf. Garey and Johnson, 1979):

KSP: Given positive integers $a_1,\ldots,a_\kappa, A$, does there exist a subset $\mathrm{I} \subseteq \mathrm{K} := \{1,\ldots,\kappa\}$ such that $\sum_{i\in \mathrm{I}} a_i = A$ is valid?

Given an arbitrary instance of KSP, a special instance of MPSC is constructed similarly to the construction of Proposition 1 in Florian *et al.* (1980) to perform a reduction from KSP to MPSC. Therefore, the following assumptions are required:

$$t_{\max} = \kappa, \tag{3.28}$$
$$d_i^{fo} = 0,\ \forall i = 1,\ldots,t_{\max} - 1, \tag{3.29}$$
$$d_{t_{\max}}^{fo} = A, \tag{3.30}$$

$$lc_i = 1, \forall i = 1,\ldots,t_{\max}, \tag{3.31}$$

$$mc_i = (a_i - 1)/a_i, \forall i = 1,\ldots,t_{\max}, \tag{3.32}$$

$$C_i^{\max} = a_i, \forall i = 1,\ldots,t_{\max}. \tag{3.33}$$

It can be claimed that KSP has a solution if and only if there is a feasible solution of MPSC with cost at most equal to A. The production in periods $1,\ldots,t_{\max}$ has to supply the demand in period $t_{\max}$ only. Due to a demand of $d_i^{fo} = 0$ in periods $1,\ldots,t_{\max}-1$, there is no backlog in these periods. Large backlog costs always make production preferable. Because of having non-zero holding cost only in the last period, it can always be found a feasible production plan with production $\sum_{i=1}^{t_{\max}} x_i = A$ that has lower costs. In summary, only the solutions of the form given by expression (3.34) will be considered.

$$\sum_{i=1}^{t_{\max}} x_i = A, \text{ with } 0 \le x_i \le a_i, \ \forall i = 1,\ldots,t_{\max}. \tag{3.34}$$

Because backlog can only occur in the last period, the costs of such a solution can be estimated as follows:

$$\sum_{t=1}^{t_{\max}} \{mc_t x_t + lc_t u_t + udcB_t\} = \sum_{i=1}^{t_{\max}} \left(\frac{a_i - 1}{a_i} x_i + sign(x_i) \right) + udc\left(A - \sum_{i=1}^{t_{\max}} x_i \right) \ge \sum_{i=1}^{t_{\max}} x_i + \left(A - \sum_{i=1}^{t_{\max}} x_i \right) = A. \tag{3.35}$$

A feasible solution with cost at most A exists if and only if both expressions (3.36) and (3.37) are fulfilled due to the strict inequality given by expression (3.38).

$$x_i \in \{0, a_i\}, \ \forall i = 1,\ldots,t_{\max}. \tag{3.36}$$

$$\sum_{t=1}^{t_{max}} x_i = A. \tag{3.37}$$

$$\frac{a_i - 1}{a_i} x_i + 1 > x_i, \forall x_i \in [0, a_i). \tag{3.38}$$

However, this is equivalent to the existence of a solution of KSP. Therefore, it can be concluded by reduction from KSP to MPSC that MPSC is NP-hard. □

As a consequence, it is necessary to look for alternative approaches since an optimum solution procedure implies a large computational burden for large-scale problem instances.

3.4 Literature Review

Production planning problems similar to MP problems are often addressed in the literature. It can be distinguished between publications that focus on rather long-term, strategic capacity planning and other papers that are related to mid-term, tactical or even operational production planning.

Many papers applied to semiconductor manufacturing belong to the first category. For instance, Bermon and Hood (1999) suggest a linear program to optimize strategic resource allocation in semiconductor manufacturing. However, this kind of long-term planning decision does not fit with MP problems. Furthermore, MPSC considers alternative assignment and outsourcing decisions. This requirement is tackled in the following two papers: Stray *et al*. (2006) and Wu *et al*. (2012) introduce MIP models that consider semiconductor manufacturing networks. In the first publication, an enterprise-wide resource planning approach is presented, whereas the second paper focuses on the selection of assembly outsourcing sites and order allocations. However, neither paper considers mid-term planning decisions. Moreover, Stray *et al*. (2006) deal with aggregated product families that are not suitable for MP problems, since the optimization of master plans has to take place at a single-item level. In addition, Denton *et al*. (2006) suggest a MIP model and corresponding heuristics for solving both strategic and operational planning decisions for a semiconductor supply chain. For this, a very detailed representation of the operations is used, which does not fit with the level of detail required for MP problems.

Publications from the second category relate MP problems to tactical decisions. For example, Chern and Hsieh (2006) and Vieira and Ribas (2008) introduce multi-objective MP formulations solved by heuristic approaches. Nevertheless, the number of resources used to complete the production is not taken into account in the objective function. Of course, it simplifies the formulation and the execution of the model since they avoid the use of binary variables; however it remains a major criterion for assignment decisions.

Zobolas *et al*. (2007) describe a repair scheme to achieve more realistic plans when demand exceeds capacity, but their approach is not appropriate for generating entire master plans. In addition, Kallrath and Maindl (2006) study the use of the SAP® Advanced Planner Optimizer (APO) tool for production planning in semiconductor environments. Their analysis reveals that the MP problem is solved by a purely rule-based algorithm, called Capable-to-Match (CTM). It turns out that CTM ignores the specific modeling of re-entrant flows of material. Hence, the capacity representation in CTM does not seem to be appropriate for tactical and operational planning decisions in semiconductor manufacturing. In addition, no computational results are provided. As one can see, many papers tackle similar planning problems, but none of the already proposed models explicitly addresses the specifics of the researched problem.

3.5 Conclusion

This chapter introduced the problem under consideration. An appropriate MIP formulation of MPSC has been proposed, and the computation complexity of MPSC has been investigated. Finally, a review of the related literature showed that the problem specifics have not been addressed so far. Parts of this chapter are already published in Ponsignon and Mönch (2012a).

The NP-hardness of the researched problem implies that it is unlikely that an efficient algorithm exists, which is guaranteed to solve the problem to optimality. An exact solution procedure may not be practical when confronted with large-scale problem instances as frequently encountered in the context of enterprise-wide planning. Hence, the next chapter discusses appropriate heuristic solution approaches of MPSC.

4. Solution Approaches of MPSC

This chapter deals with solution approaches that can efficiently solve problem instances of MPSC with respect to solution quality and computing time. The NP-hardness of the problem is due to the usage of the binary condition (3.8). A common method to ease the computation of a MIP model consists in replacing binary variables by semi-continuous variables. A semi-continuous variable X must be either zero or any value equal or larger than a specified positive number c, i.e., $X = 0$ or $X \geq c$ (cf. Voß and Woodruff, 2006). Using a semi-continuous variable saves the declaration of a constraint that defines the relationship between a binary variable and a real valued variable, i.e., constraint (3.6) in MPSC. Even though the resulting model is simplified, the size of the branch-and-bound tree remains the same and a MIP solver still needs to branch between the two cases where the variable is either forced to be zero or constrained to be above the defined threshold. The possibility of declaring u_{pmt} as a semi-continuous variable is investigated. It turns out that such a formulation is not suitable per definition for fixed cost as only two levels of value are accepted. In the literature, semi-continuous variables are often used for modeling production quantities that are subject to a minimum lot size constraint (cf. Voß and Woodruff, 2006; Stadtler, 2012). Since the computational burden of the problem setting cannot be decreased with alternative modeling strategies, efficient heuristics are required.

In this chapter, heuristic solution approaches of MPSC are proposed that are of different natures, i.e., a product-based decomposition scheme, a rule-based assignment scheme, and a metaheuristic namely a genetic algorithm (GA). After introducing each method in detail, the performance of the approaches will be assessed in designed experiments.

4.1 Product-based Decomposition Scheme (PD-MPSC)

4.1.1 Motivation

Because of its NP-hardness, MPSC may not be solvable to optimality in reasonable time. However, it is known that decomposition approaches allow reducing the computational burden for large-scale production planning and scheduling problems (cf., for example, Pochet and Wolsey, 2006). The main idea behind it is to derive several smaller subproblems from the initial problem definition and to solve each subproblem successively. This leads to a series of subsolutions that are combined to obtain an overall solution. It is referred to as fix-and-optimize heuristic.

4.1.2 Product-based Decomposition Scheme for Solving MPSC

Since most of the parameters of MPSC are product-dependent, it is possible to build a feasible solution with good quality by choosing an appropriate product sequence. Therefore, the products are first sorted according to their criticality, i.e., products with high demand of firm customer orders and high backlog costs have high priority. The product of cost and demand is considered as criticality measure to obtain a value for comparison. Then, a bundle of products is selected from the ranked list to form a subproblem. An appropriate size of the product subsets is selected based on computational experiments (cf. Subsection 4.4.3). The reduced MPSC problem is solved by taking the demand for those products and the actual remaining capacity into account. Afterwards, the values of the decision variables are frozen, and the maximum capacity limits are decreased according to the loading that is already planned. Finally, the global objective function value is incremented. The algorithm terminates when all product subsets have been considered. The resulting approach is a product-based decomposition scheme denoted by PD-MPSC. The PD-MPSC scheme leads to Algorithm 4.1.

Algorithm 4.1: PD-MPSC scheme.

1 Initialize the objective function value $f_{global} = 0$.

2 Sort products according to the criteria CR_p in descending order with

$$CR_p := \sum_{t=1}^{t_{\max}} udc_{pt} d_{pt}^{fo}, \forall p = 1,..,p_{\max}. \tag{4.1}$$

3 Decompose product set P into $\varphi \geq 2$ disjoint subsets $P_1,...,P_\varphi$ of equal size, only the last subset might have a different size, such that products with similar CR_p values are gathered in the same subset or in consecutive subsets. The subsets are defined as:

$$\bigcup_{i=1}^{\varphi} P_i = P, \tag{4.2}$$

$$P_i \cap P_j = \varnothing, \forall i = 1,..,\varphi, \forall j = 1,..,\varphi, i \neq j. \tag{4.3}$$

4 Solve MPSC given by objective function (3.1) and constraints (3.2)-(3.8) for the current product subset P_i by taking the actual maximum capacity limits into account and by setting the minimum capacity bounds to zero.

5 Increment the global objective function value with the objective function value of the current subproblem:

$$f_{global} := f_{global} + f_{P_i}\left(x, u, I, B, s^{sr}\right). \tag{4.4}$$

6 Decrease the maximum capacity limits as follows:

$$C_{mbt}^{\max} := C_{mbt}^{\max} - \sum_{p \in P_i} \sum_{k=0}^{\min(k_{\max}, t_{\max}-t)} cc_{bk}^{pm} \left(x_{p,m,t+k} + x_{p,m,t+k}^{i}\right), \forall m = 1,..,m_{\max}, \forall b = 1,..,b_{m,\max},$$

$\forall t = 1,..,t_{\max}$. (4.5)

7 As long as any product subset has not been considered, increment the index i of the current product subset P_i and go to Step 4, else return the objective function value f_{global}.

One sees that the minimum capacity limit is ignored in the previous algorithm. Obviously, considering a minimum utilization threshold leads to an artificial increase of production quantities for products of the first subset. As a result, the remaining capacity may not be sufficient for other subsets. That is why an *a posteriori* repair loop is proposed in Algorithm 4.2.

Algorithm 4.2: Repair scheme subsequent to Algorithm 4.1

---

8 $m = 0$, $b = 0$, $t = 0$.

9 $m := m + 1$.

10 $b := b + 1$.

11 $t := t + 1$.

12 Calculate the usage of bottleneck b in wafer fab m in period t as follows:

$$Load_{mbt}^{(1)} := \sum_{p=1}^{p_{\max}} \sum_{k=0}^{\min(k_{\max}, t_{\max}-t)} cc_{bk}^{pm} \left(x_{p,m,t+k} + x_{p,m,t+k}^{i} \right). \quad (4.6)$$

13 As long as $Load_{mbt}^{(1)} < C_{mbt}^{\min}$, repair the violation with the following step-by-step approach:

- Select the next product p on a list that is sorted according to average holding costs $\overline{hc}_p = \frac{1}{t_{\max}} \sum_{s=1}^{t_{\max}} hc_{ps}$ in ascending order that satisfies both conditions $x_{pmt} > 0$ and $\sum_{k=0}^{\min(k_{\max}, t_{\max}-t)} cc_{bk}^{pm} > 0$.
- Choose the quantity $\tilde{x} \sim U\left[0, \left(C_{mbt}^{\max} - Load_{mbt}^{(1)} \right) \Big/ \left(p_{\max} \sum_{k=0}^{\min(k_{\max}, t_{\max}-t)} cc_{bk}^{pm} \right) \right]$, (4.7)

  where $x \sim U[a,b]$ denotes a realization of a random variable that is uniformly distributed over $[a,b]$ for $a,b \in \mathbb{R}$ and $a < b$.
- Increase the production quantity by $x_{pmt} := x_{pmt} + \tilde{x}$. (4.8)
- Increase the inventory levels for the current and the subsequent periods accordingly:

  $I_{ps} := I_{ps} + \tilde{x}$, $\forall s = t,..,t_{\max}$. (4.9)
- If all products have been considered and $Load_{mbt}^{(1)} < C_{mbt}^{\min}$, then select the first

product of the list that satisfies $\sum_{k=0}^{\min(k_{\max},t_{\max}-t)} cc_{bk}^{pm} > 0$ and add the quantity $\tilde{x} := \left(C_{mbt}^{\min} - Load_{mbt}^{(1)}\right) \Big/ \sum_{k=0}^{\min(k_{\max},t_{\max}-t)} cc_{bk}^{pm}$ with respect to expressions (4.8) and (4.9). There always exists at least one product with this property because of the assumption in Section 3.2.

14 If $t < t_{\max}$, then go to Step 11.

15 If $b < b_{m,\max}$, then set $t = 0$, and go to Step 10.

16 If $m < m_{\max}$, then set $b = 0$ and $t = 0$, and go to Step 9.

In Algorithm 4.2, the bottleneck usage is checked in each time period and increased in case that the minimum capacity bound is not met. The lack of production is filled in when the gap is distributed between the products. Preferably, the production quantities of a large number of products are increased rather than those of only one or a few arbitrarily chosen products since it reduces the risk of scrap stocks. However, it has to be paid attention to holding costs. It is also important to add production quantities only where fixed production costs are already counted. The setting (4.7) makes sure that the maximum limit is not exceeded, and the presence of $p_{\max}$ ensures that only small quantities are added in each iteration. Unsold goods are stored for the rest of the horizon as described by expression (4.9). The procedure stops whenever the minimum limit is reached or exceeded. It also ensures that the minimum limit is met even when none of the products satisfies the two conditions of the first substep within Step 13.

4.2 Rule-based Assignment Scheme (RA-MPSC)

The rule-based assignment scheme denoted by RA-MPSC is a rather simple allocation algorithm that provides a single solution of a given MPSC problem. It starts by randomly selecting a first period from $t = 1, \ldots, t_{\max}$. Firm customer orders are allocated with a higher priority than supply reservations. Therefore, products are sorted in descending order in two lists, i.e., L^{udc} and L^{rev}, with respect to their average backlog cost $\overline{udc}_p$ and their average revenue $\overline{rev}_p$, respectively. Average backlog cost and revenue are defined as follows:

$$\overline{udc}_p = \frac{1}{t_{\max}} \sum_{s=1}^{t_{\max}} udc_{ps}, \tag{4.10}$$

$$\overline{rev}_p = \frac{1}{t_{\max}} \sum_{s=1}^{t_{\max}} rev_{ps}. \tag{4.11}$$

RA-MPSC considers the ranking from the first list for allocating firm customer orders, and then the second list is used for allocating supply reservations. Hence, the next product on the list under consideration is selected. When allocating firm customer orders, the facility with the most remaining capacity is chosen, i.e., the difference between the maximum available

capacity and the load already planned, among the in-house locations as they have lower location costs. In the case that it cannot be decided between several sites because they have equal remaining available capacity, the fab with the lower location cost is chosen. Moreover, if the demand for the selected product in the current time period cannot be entirely assigned to one wafer fab, an additional site, either in-house location or silicon foundry, with the second most remaining available capacity is selected. Afterwards, the demand is assigned to the chosen wafer fabs with respect to constraint (3.5). This procedure minimizes the backlog. In addition, the production for one product in a given time period is assigned to no more than two wafer fabs in order to avoid high location costs. When assigning supply reservations quantities are exclusively allocated to wafer fabs where firm customer orders have already been planned. Hence, production partitioning does not generate additional costs. In the case that two wafer fabs are used for satisfying firm customer orders, the supply reservations are first allocated to the wafer fab with the lower variable manufacturing cost mc_{pmt} without exceeding the maximum capacity limit. If there is any remaining supply reservation it is assigned to the second wafer fab. Moreover, if capacity is not sufficient in the current period while allocating firm customer orders or supply reservations, the algorithm looks for available capacity in the $n_{preprod}$ previous periods, i.e., periods $\max\{1, t-n_{preprod}\},\ldots,\max\{1,t-1\}$ are considered. This leads to preproduction and stock building, but it avoids backlog. Quantities are assigned to wafer fabs where production has already been planned so that no extra location cost is generated. If no capacity is available in the five previous periods, backlog occurs. Finally, this procedure is repeated by increasing the current period t by $t := t \bmod t_{\max} + 1$ until all products and all time periods have been considered. Eventually, the production quantities are returned, and the other decision variables are derived. The approach for defining inventory level, backlog level, binary variables, and sales quantities is described in the next Subsection by formulas (4.16)-(4.20). In addition, minimum capacity limits are taken into account. If the loading is too low, it is increased *a posteriori* by means of a repair loop similar to Algorithm 4.2 used in PD-MPSC. It is notable that RA-MPSC is used as an initialization scheme in the heuristic GA-MPSC, which is presented in the next section. The RA-MPSC scheme leads to Algorithm 4.3.

Algorithm 4.3: RA-MPSC scheme

1 Sort the lists L^{udc} and L^{rev}.

2 Choose a period from $t = 1,\ldots,t_{\max}$.

3 Starting from the top, choose the next product p on the list L^{udc}.

4 Choose among the in-house facilities the wafer fab m with the most remaining capacity C_{mt}^{rem} that is given by (4.12). In case of a tie, choose the facility m with the lowest lc_{pmt}. Allocate the quantity $x_{pmt} := \min\{d_{pt}^{fo}, C_{mt}^{rem}\}$.

$$C_{mt}^{rem} := \min_{b=1,\ldots,b_{m,mxa}} \left\{ \left(C_{mbt}^{\max} - Load_{mbt}^{(1)}\right) \Big/ \sum_{k=0}^{\min(k_{\max}, t_{\max}-t)} cc_{bk}^{pm} \right\}. \tag{4.12}$$

5 If $\tilde{x} := d_{pt}^{fo} - C_{mt}^{rem} > 0$, then choose another facility m among all wafer fabs with the

second most remaining capacity C_{mt}^{rem}. Allocate the quantity $x_{pmt} := \min\left\{\tilde{x}, C_{mt}^{rem}\right\}$. Update $\tilde{x} := \tilde{x} - C_{mt}^{rem}$.

6 While $\tilde{x} > 0$ and $\max_{m=1,\ldots,m_{\max}}\left\{C_{mt}^{rem}\right\} > 0$, look for preproduction opportunities in the past time periods $s = \max\left\{1, t - n_{preprod}\right\},\ldots,\max\left\{1, t-1\right\}$, successively. Exclusively consider wafer fabs m where $x_{pms} > 0$, wherein production is preferably allocated to the wafer fabs with the lowest variable manufacturing costs. Allocate the quantity $x_{pms} := x_{pms} + \min\left\{\tilde{x}, C_{mt}^{rem}\right\}$. Update $\tilde{x} := \tilde{x} - C_{mt}^{rem}$.

7 If the bottom of the list L^{udc} is not reached and $\max_{m=1,\ldots,m_{\max}}\left\{C_{mt}^{rem}\right\} > 0$, then go to Step 3, otherwise, if $\max_{m=1,\ldots,m_{\max}}\left\{C_{mt}^{rem}\right\} = 0$ go to Step 13.

8 Starting from the top, choose the next product p on the list L^{rev}.

9 Choose a wafer fab m with $x_{pmt} > 0$. In case of a tie, choose the facility with the lowest mc_{pmt}. Allocate the quantity $x_{pmt} := x_{pmt} + \min\left\{d_{pt}^{sr}, C_{mt}^{rem}\right\}$.

10 If $\tilde{x} := d_{pt}^{sr} - C_{mt}^{rem} > 0$, and if there exists another facility m with $x_{pmt} > 0$, then allocate the quantity $x_{pmt} := x_{pmt} + \min\left\{\tilde{x}, C_{mt}^{rem}\right\}$, and update $\tilde{x} := \tilde{x} - C_{mt}^{rem}$.

11 While $\tilde{x} > 0$ and $\max_{m=1,\ldots,m_{\max}}\left\{C_{mt}^{rem}\right\} > 0$, look for preproduction opportunities as in Step 6.

12 If the bottom of the list L^{rev} is not reached and $\max_{m=1,\ldots,m_{\max}}\left\{C_{mt}^{rem}\right\} > 0$, then go to Step 8.

13 If not all time periods have been considered, then increment the time period $t := t \bmod t_{\max} + 1$, and go to Step 3.

14 Return the quantity x_{pmt}. Derive the decisions variables I_{pt}, B_{pt}, u_{pmt}, s_{pt}^{fo}, and s_{pt}^{sr}.

4.3 Genetic Algorithm (GA-MPSC)

4.3.1 Motivation and Basic Principle

It is known from the literature that GAs are powerful heuristics widely used for solving large-scale combinatorial optimization problems from manufacturing (cf. Aytug *et al*., 2003). A series of papers introduce GAs for planning or assignment problems that are similar to MPSC to a certain extent (cf., among others, Hornung and Mönch, 2008). Therefore, a GA is proposed to solve MPSC problems, even though other metaheuristics seem to be appropriate too. The resulting approach is denoted by GA-MPSC.

Different versions of GA are described in the literature (cf. Goldberg, 1989; Michalewicz, 1996; Wall, 2012). The implementation that is used in GA-MPSC is described in the following.

The scheme starts with a pool of chromosomes randomly spread over the search space. Each chromosome represents a solution of the considered problem instance. A fitness value derived from the objective function value of the problem is assigned to each single chromosome. The set of all chromosomes from the same iteration is called a population. A selector is used to pick out parent chromosomes among the current population. From two selected parents, two offspring are generated according to a crossover probability $p_{crossover}$ that states when a crossover is performed or when the parents are duplicated. Next, a random number is assigned to each newly created chromosome. When this number is smaller than a given mutation probability $p_{mutation}$, then a mutation operation takes place. Each generation the algorithm adds the offspring chromosomes to the current population, and then removes the worst members depending on their fitness to decrease the population to its original size. A replacement percentage $p_{replacement}$ determines the number $n_{replacement}$ of children to generate. An overlapping population of size ps is the consequence, and the resulting GA is a steady state one. The GA terminates whenever a maximum number of generations $gen_{\max}$ is reached, or when the diversity within the population falls below a predefined threshold $div_{\min}$. The diversity value results from the calculation of the standard deviation based upon the fitness values of the chromosomes of the current population. A diversity value of zero means that the chromosomes have the same fitness value. The GA scheme that is used in GA-MPSC leads to Algorithm 4.4.

Algorithm 4.4: GA scheme

1 Generate the chromosomes of the initial population.

2 Evaluate the fitness of the chromosomes.

3 Repeat until at least one of the termination criteria is satisfied, i.e., diversity falls below $div_{\min}$ or the maximum number of generations $gen_{\max}$ is reached.

4 For $i := 1$ to $\left\lceil \frac{n_{replacement}}{2} \right\rceil$.

5 Select two parent chromosomes $\mathbf{C}^{(1)}$ and $\mathbf{C}^{(2)}$ among the current population.

6 Choose $z_{crossover} \sim U[0,1]$.

7 If $z_{crossover} < p_{crossover}$,

- Then obtain the offspring chromosomes $\mathbf{C}^{(3)}$ and $\mathbf{C}^{(4)}$ by applying a crossover operator,
- Else duplicate both parents chromosomes, i.e., $\mathbf{C}^{(3)} := \mathbf{C}^{(1)}$ and $\mathbf{C}^{(4)} := \mathbf{C}^{(2)}$.

8 Do for each of the offspring chromosomes.

9 Choose $z_{mutation} \sim U[0,1]$.

10 If $z_{mutation} < p_{mutation}$, then apply a mutation operator to the chromosome under consideration.

11 End Do.

12 Evaluate the fitness of both offspring chromosomes.

13 End For.

14 Add the $n_{replacement}$ offspring chromosomes to the population and keep the ps fittest chromosomes.

15 End Repeat.

16 Return the best chromosome of the current population

It is crucial to design an appropriate chromosome representation, proper initialization schemes, and suitable genetic operators. Hence, in the remainder of this section important features of GA-MPSC will be presented in more detail.

4.3.2 Chromosome Representation

Each chromosome of the population of GA-MPSC represents production quantities and sales quantities of one solution of a problem instance of MPSC. A three-dimensional structure is used as chromosome $\mathbf{C}$. It is defined as follows:

$$\mathbf{C}_{pmt} := x_{pmt}, \qquad \forall p = 1,..,p_{\max}, \forall m = 1,..,m_{\max}, \forall t = 1,..,t_{\max}, \tag{4.13}$$

$$\mathbf{C}_{p,m_{\max}+1,t} := s_{pt}^{sr}, \qquad \forall p = 1,..,p_{\max}, \forall t = 1,..,t_{\max}, \tag{4.14}$$

$$\mathbf{C}_{p,m_{\max}+2,t} := s_{pt}^{fo}, \qquad \forall p = 1,..,p_{\max}, \forall t = 1,..,t_{\max}. \tag{4.15}$$

It is notable that the sales quantities related to supply reservations and firm customer orders are encoded in the $(m_{\max}+1)$-*th* and $(m_{\max}+2)$-*th* layers of chromosome $\mathbf{C}$, respectively. Figure 4.1 shows the chromosome representation used.

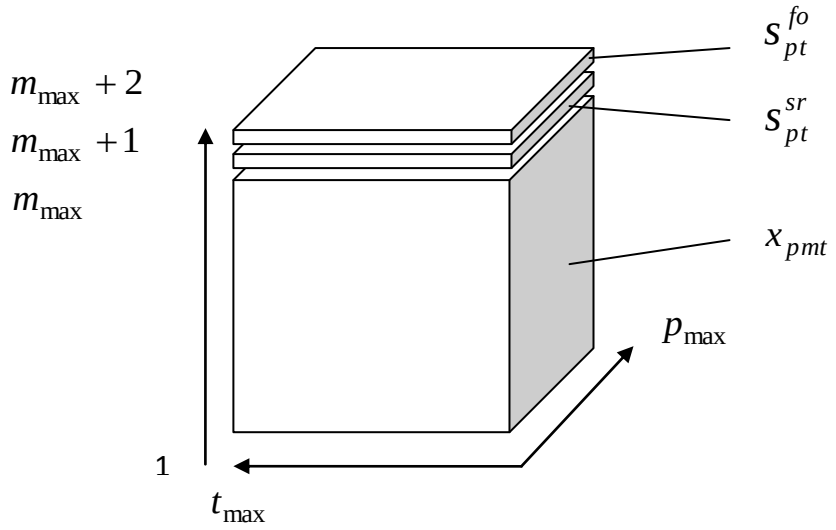

Figure 4.1: Chromosome representation.

All other decision variables are deduced from the values encoded in chromosome $\mathbf{C}$. For this, the formulas (30)-(32) are used, which are derived from constraints (3.2), (3.3) and (3.6):

$$I_{pt} := I_{pt-1} + \sum_{m=1}^{m_{\max}} x_{pmt}^{i} + \sum_{m=1}^{m_{\max}} \mathbf{C}_{pmt} - \mathbf{C}_{p,m_{\max}+1,t} - \mathbf{C}_{p,m_{\max}+2,t}, \quad \forall p = 1,..,p_{\max}, \forall t = 1,..,t_{\max}, \tag{4.16}$$

$$B_{pt} := d_{pt}^{fo} + B_{pt-1} - \mathbf{C}_{p,m_{\max}+2,t}, \quad \forall p = 1,..,p_{\max}, \forall t = 1,..,t_{\max}, \tag{4.17}$$

$$u_{pmt} := sign(\mathbf{C}_{pmt}), \quad \forall p = 1,..,p_{\max}, \forall m = 1,..,m_{\max}, \forall t = 1,..,t_{\max}. \tag{4.18}$$

4.3.3 Generating the Initial Population

The chromosomes of the initial population are formed randomly taking the capacity restriction (3.5) into account. It is not necessary to consider the other constraints of MPSC since they are not subject to maximum bounds, but it has to be ensured that constraint (3.7) is fulfilled. To achieve a heterogeneous initial population the following three constructive schemes are used to determine half of the initial population:

- Scheme 1 uses RA-MPSC scheme,
- Scheme 2 assigns random quantities to products and facilities arbitrarily chosen until the minimum utilization threshold of each bottleneck is reached in every time period, and
- Scheme 3 is similar to Scheme 2 except that bottlenecks are filled until the maximum capacity limit of each bottleneck is met in every period.

Scheme 1 generates high-quality chromosomes with respect to the objective function (3.1), while Scheme 2 and Scheme 3 produce solutions with lower objective function values.

Moreover, sales quantities in chromosomes are defined with the following heuristic procedure regardless of the initialization schemes that is used:

$$\mathbf{C}_{p,m_{\max}+2,t} = s_{pt}^{fo} := \min\left\{ d_{pt}^{fo} + B_{pt-1}, \sum_{m=1}^{m_{\max}} \mathbf{C}_{pmt} + \sum_{m=1}^{m_{\max}} x_{pmt}^{i} + I_{pt-1} \right\}, \forall p = 1,..,p_{\max}, \forall t = 1,..,t_{\max}, \quad (4.19)$$

$$\mathbf{C}_{p,m_{\max}+1,t} = s_{pt}^{sr} := \min\left\{ d_{pt}^{sr}, \sum_{m=1}^{m_{\max}} \mathbf{C}_{pmt} + \sum_{m=1}^{m_{\max}} x_{pmt}^{i} + I_{pt-1} - \mathbf{C}_{p,m_{\max}+2,t} \right\}, \forall p = 1,..,p_{\max}, \forall t = 1,..,t_{\max}. \quad (4.20)$$

The expressions (4.19) and (4.20) both lead to $B_{pt} \geq 0$ and $I_{pt} \geq 0$, $\forall p = 1,..,p_{\max}$, $\forall t = 1,..,t_{\max}$. Based on this constructive procedure, it is made sure that the initial chromosomes satisfy the capacity constraints, i.e., they represent feasible solutions to a problem instance of MPSC. It is important to determine the sales quantities related to firm customer orders in the first place to ensure that only a small amount of backlog occurs. The probabilities that Scheme 1, Scheme 2, and Scheme 3 are applied are set to 0.10, 0.45, and 0.45, respectively.

To achieve a diversified initial population, an initialization procedure is used that is based on a distance function known as heterogeneity criterion. It allows investigating a large portion of the solution space, and it enhances the optimization more than a homogenous population does. The scheme for the generation of the initial population leads to Algorithm 4.5.

Algorithm 4.5: Scheme for the generation of the initial population.

1 Generate half of the population.

2 Initialize the index h referring to newly produced chromosomes by $h = 0$.

3 Generate a new chromosome $\mathbf{C}^{(new)}$ according to one of the three schemes. The probabilities that the first, the second, and the third schemes are applied are set to 0.10, 0.45, and 0.45, respectively.

4 Measure the Euclidean distances between $\mathbf{C}^{(new)}$ and all the other chromosomes of the current population:

$$d_{new,i}\left(\mathbf{C}^{(new)}, \mathbf{C}^{(i)}\right) := \sqrt{\sum_{p=1}^{p_{\max}} \sum_{m=1}^{m_{\max}+2} \sum_{t=1}^{t_{\max}} \left(\mathbf{C}_{pmt}^{(new)} - \mathbf{C}_{pmt}^{(i)}\right)^2}, \forall i = 1,...,\lfloor ps/2 \rfloor + h. \quad (4.21)$$

5 If at least n_{chr} chromosomes from the current population can be found for which the Euclidean distance to $\mathbf{C}^{(new)}$ is higher than $d_{\min}$ that is defined as 110% of the average Euclidean distance among chromosomes of the first half of the population:

- Then add chromosome $\mathbf{C}^{(new)}$ to the population and increment $h := h + 1$,
- Else go to Step 3, unless a predefined maximum number of attempts $try_{\max}$ has been reached. After $try_{\max}$ unsuccessful trials, the chromosome with the highest distance is kept, h is incremented, and it proceeds to Step 6.

6 If $h < \lceil ps/2 \rceil$, then go to Step 3.

4.3.4 Genetic Operators

4.3.4.1 Crossover

From two given parent chromosomes $\mathbf{C}^{(1)}$ and $\mathbf{C}^{(2)}$, two offspring chromosomes $\mathbf{C}^{(3)}$ and $\mathbf{C}^{(4)}$ are obtained by applying an arithmetical crossover operator. The crossover scheme is described by Algorithm 4.6.

Algorithm 4.6: Crossover scheme.

1 Choose two values $\lambda_1, \lambda_2 \in \mathbb{R}_+$ as follows:

$$\lambda_1 \sim U[0,K_1], \lambda_2 \sim U[0,K_2] \text{ with } K_1 \geq 1, K_2 \geq 1. \tag{4.22}$$

2 Set offspring as follows:

$$\begin{cases} \mathbf{C}^{(3)}_{pmt} = \lambda \mathbf{C}^{(1)}_{pmt} + (1-\lambda)\mathbf{C}^{(2)}_{pmt}, \\ \mathbf{C}^{(4)}_{pmt} = (1-\lambda)\mathbf{C}^{(1)}_{pmt} + \lambda \mathbf{C}^{(2)}_{pmt}, \end{cases} \text{with } \lambda = \begin{cases} \lambda_1, & if\ m \leq m_{\max}, \\ \lambda_2, & if\ m = m_{\max}+1, \\ \lambda_2, & if\ m = m_{\max}+2, \end{cases} \tag{4.23}$$

$$\forall p = 1,..,p_{\max}, \forall m = 1,..,m_{\max}+2, \forall t = 1,..,t_{\max}.$$

It turns out that the two offspring chromosomes are created by exchanging the role of λ and $1-\lambda$. It can be proved that children are feasible solutions of MPSC when $\lambda_1 = \lambda_2$ and λ_1 and λ_2 are lower than or equal to 1. In fact, an arbitrary convex combination of feasible solutions of MPSC is also a feasible solution since the set determined by constraint (3.5) is convex.

Proposition 2: $\mathbf{C}^{(3)}$ is a feasible solution of MPSC if $\mathbf{C}^{(1)}$ and $\mathbf{C}^{(2)}$ are feasible solutions, $\lambda_1 = \lambda_2$, and $K_1 = K_2 = 1$.

Proof: Proposition 2 is proved by showing that constraints (3.4), (3.5), and (3.7) hold for $\mathbf{C}^{(3)}$.

First, given that $\mathbf{C}^{(1)}$ and $\mathbf{C}^{(2)}$ are feasible solutions, constraint (3.4) is fulfilled for the parent chromosomes. The expression (4.24) shows that it is also fulfilled for $\mathbf{C}^{(3)}$:

$$\mathbf{C}^{(3)}_{p,m_{\max}+1,t} := s^{sr,(3)}_{pt} = \lambda s^{sr,(1)}_{pt} + (1-\lambda)s^{sr,(2)}_{pt} \leq \lambda d^{sr}_{pt} + (1-\lambda)d^{sr}_{pt} = d^{sr}_{pt},$$
$$\forall p = 1,..,p_{\max}, \forall t = 1,..,t_{\max}. \tag{4.24}$$

Then, the capacity constraint (3.5) is simplified by replacing the terms related to the minimum bound, maximum bound, WIP, and capacity consumption by the constants A_1, A_2, A_3, and a, respectively. Hence, the capacity constraints related to $\mathbf{C}^{(1)}$ and $\mathbf{C}^{(2)}$ are given by:

$$A_1 \leq a\mathbf{C}^{(1)}_{pmt} + A_3 \leq A_2,\ A_1 \leq a\mathbf{C}^{(2)}_{pmt} + A_3 \leq A_2,$$
$$\forall p = 1,..,p_{\max}, \forall m = 1,..,m_{\max}, \forall t = 1,..,t_{\max}. \tag{4.25}$$

By pursuing the simplification, the capacity constraints can be formulated as follows:

$$\tilde{A}_1 \le \mathbf{C}^{(1)}_{pmt} \le \tilde{A}_2,\ \tilde{A}_1 \le \mathbf{C}^{(2)}_{pmt} \le \tilde{A}_2,$$
$$\forall p = 1,..,p_{\max}, \forall m = 1,..,m_{\max}, \forall t = 1,..,t_{\max}, \quad (4.26)$$

where the quantities $\tilde{A}_1, \tilde{A}_2$ are appropriate constants. When applying the crossover operator to $\mathbf{C}^{(1)}$ and $\mathbf{C}^{(2)}$, the following expressions hold:

$$\lambda\tilde{A}_1 \le \lambda\mathbf{C}^{(1)}_{pmt} \le \lambda\tilde{A}_2,\ \forall p = 1,..,p_{\max}, \forall m = 1,..,m_{\max}, \forall t = 1,..,t_{\max}, \quad (4.27)$$
$$(1-\lambda)\tilde{A}_1 \le (1-\lambda)\mathbf{C}^{(2)}_{pmt} \le (1-\lambda)\tilde{A}_2,\ \forall p = 1,..,p_{\max}, \forall m = 1,..,m_{\max}, \forall t = 1,..,t_{\max}. \quad (4.28)$$

The sum of expressions (4.27) and (4.28) leads to:

$$\tilde{A}_1 \le \mathbf{C}^{(3)}_{pmt} \le \tilde{A}_2,\ \forall p = 1,..,p_{\max}, \forall m = 1,..,m_{\max}, \forall t = 1,..,t_{\max}. \quad (4.29)$$

Thus, it confirms that an arbitrary convex combination of feasible solutions of MPSC is also a feasible solution due to the convexity of the set determined by constraint (3.5).

Finally, the validity of non-negativity constraint (3.7) is investigated. Given the construction of the crossover operator as defined by expressions (4.22) and (4.23), the quantities encoded in $\mathbf{C}^{(3)}$ are positive. Hence, the following equations hold:

$$\mathbf{C}^{(3)}_{pmt} := x^{(3)}_{pmt} \ge 0,\ \forall p = 1,..,p_{\max}, \forall m = 1,..,m_{\max}, \forall t = 1,..,t_{\max}, \quad (4.30)$$
$$\mathbf{C}^{(3)}_{p,m_{\max}+1,t} := s^{sr,(3)}_{pt} \ge 0,\ \mathbf{C}^{(3)}_{p,m_{\max}+2,t} := s^{fo,(3)}_{pt} \ge 0,\ \forall p = 1,..,p_{\max}, \forall t = 1,..,t_{\max}. \quad (4.31)$$

The non-negativity of inventory and backlog levels that are derived from $\mathbf{C}^{(3)}$ is proved by expressions (4.32) and (4.33) by referring to constraints (3.2) and (3.3), respectively. Per definition, inventory and backlog levels derived from the parent chromosomes are non-negative, i.e., $I^{(1)}_{pt}, I^{(2)}_{pt}, B^{(1)}_{pt}, B^{(2)}_{pt} \ge 0$.

$$I^{(3)}_{pt} := I^{(3)}_{pt-1} - \mathbf{C}^{(3)}_{p,m_{\max}+2,t} - \mathbf{C}^{(3)}_{p,m_{\max}+1,t} + \sum_{m=1}^{m_{\max}} \mathbf{C}^{(3)}_{pmt} + \sum_{m=1}^{m_{\max}} x^{i,(3)}_{pmt} = \lambda I^{(1)}_{pt} + (1-\lambda) I^{(2)}_{pt} \ge 0, \forall p = 1,..,p_{\max}, \forall t = 1,..,t_{\max}, \quad (4.32)$$

$$B^{(3)}_{pt} := d^{fo}_{pt} + B^{(3)}_{pt-1} - \mathbf{C}^{(3)}_{p,m_{\max}+2,t} = \lambda B^{(1)}_{pt} + (1-\lambda) B^{(2)}_{pt} \ge 0, \forall p = 1,..,p_{\max}, \forall t = 1,..,t_{\max}. \quad (4.33)$$

It can be concluded from the expressions (4.24), (4.29), and (4.30)-(4.33) that Proposition 2 is valid. □

However, it is unlikely that an arithmetical crossover operator with λ values smaller than 1 can increase the objective function value because parts of the objective function (3.1) are linear (except the first and the last term) and therefore successive generations are within the convex hull of the initial population. Hence, the offspring cannot outperform the best solution from previous generations. In addition, the last term, i.e., the location costs, can only decrease the objective function value by generating more fixed production costs in the case that parent chromosomes use different wafer fabs. In order to avoid this disadvantage by design, one uses

the crossover operator presented above with λ values potentially larger than 1 in order to let the child chromosomes improve their performance and to expedite the search.

On the other hand, children obtained from expressions (4.22) and (4.23) can be infeasible solutions of MPSC as the combination is no longer convex, i.e., capacity restriction (3.5) can be unfulfilled and I_{pt}, B_{pt}, and $\mathbf{C}_{pmt}$ can be negative. For this reason a penalty function is used as suggested by Deb (2000) to make sure that any infeasible chromosome has an objective function value smaller than any of the feasible chromosomes. Thereby, the probability of being selected for the next generation and for reproduction is low for infeasible solutions. Thus, to calculate the objective function value of a chromosome $\mathbf{C}$ a function f_{pen} is used that verifies if minimum and maximum capacity restrictions (3.5) and restriction (3.7) are fulfilled. The function f_{pen} is defined as follows:

$$f_{pen}(\mathbf{C}) := \begin{cases} f(\mathbf{C}), & \text{if no violation} \\ f_{\min} - \sum_{m=1}^{m_{\max}} \sum_{b=1}^{b_{m,\max}} \sum_{t=1}^{t_{\max}} \max\left\{0, Load_{mbt}^{(2)} - C_{mbt}^{\max}\right\} - \sum_{m=1}^{m_{\max}} \sum_{b=1}^{b_{m,\max}} \sum_{t=1}^{t_{\max}} \max\left\{0, C_{mbt}^{\min} - Load_{mbt}^{(2)}\right\} & \\ \quad - \sum_{p=1}^{p_{\max}} \sum_{t=1}^{t_{\max}} \left(\left|\min\left\{0, I_{pt}\right\}\right| + \left|\min\left\{0, B_{pt}\right\}\right| + \sum_{m=1}^{m_{\max}} \left|\min\left(0, \mathbf{C}_{pmt}\right)\right| \right), & \text{otherwise} \end{cases} \quad (4.34)$$

$$\text{with } Load_{mbt}^{(2)} := \sum_{p=1}^{p_{\max}} \sum_{k=0}^{\min(k_{\max}, t_{\max}-t)} cc_{bk}^{pm} \left(\mathbf{C}_{p,m,t+k} + x_{p,m,t+k}^{i}\right). \quad (4.35)$$

If necessary, the summation of constraint violations is subtracted from the lowest objective function value $f_{\min}$ of all feasible solutions in the current population. As found in the computational experiments, there are feasible solutions within each single generation. Otherwise, if there are no constraint violations, the objective function f is used as defined in equation (3.1). Finally, the fitness of chromosome $\mathbf{C}$ is derived from its objective function value by means of a scaling function. The GA-MPSC scheme uses sigma truncation as scaling function because it is a method that accepts negative fitness values. It may occur with respect to expression (4.34) when the sum of costs is higher than the revenues or when the summation of constraint violations is higher than $f_{\min}$. The roulette wheel method is chosen as selector, i.e., the higher the fitness, the more likely a chromosome is selected as a parent (cf. Goldberg, 1989).

4.3.4.2 Mutation

A dynamic mutation operator is implemented that allows continuously decreasing the impact of mutation while the number of generations increases (cf. Michalewicz, 1996). A similar technique is used by Hornung and Mönch (2008). It starts by determining a gene of chromosome $\mathbf{C}$ that has to be mutated. Then, in order to ensure the feasibility of the solution, an upper bound ub for the value of the gene is calculated. Infeasible chromosomes that result from the crossover operator are excluded from the mutation. Depending on the position of the gene to mutate, ub represents either the maximum remaining capacity or the maximum quantity that can be sold. The mutation scheme is described by Algorithm 4.7.

Algorithm 4.7: Mutation scheme.

1 Determine randomly a gene position (p^*, m^*, t^*) in chromosome $\mathbf{C}$.

2 Calculate ub depending on the gene position:

- If $m^* = m_{max} + 2$, then $ub := \min\left\{ d^{fo}_{p^*t^*} + B_{p^*t^*-1}, \sum_{m=1}^{m_{\max}} \mathbf{C}_{p^*mt^*} + \sum_{m=1}^{m_{\max}} x^i_{p^*mt^*} + I_{p^*t^*-1} \right\}$, (4.36)

- If $m^* = m_{max} + 1$, then $ub := \min\left\{ d^{sr}_{p^*t^*}, \sum_{m=1}^{m_{\max}} \mathbf{C}_{p^*mt^*} + \sum_{m=1}^{m_{\max}} x^i_{p^*mt^*} + I_{p^*t^*-1} \right\}$, (4.37)

- If $1 \le m^* \le m_{max}$, then $ub := \min_{b=1,..,b_{m,\max}} \left\{ \left(C^{\max}_{m^*bt^*} - Load^{(2)}_{m^*bt^*} \right) \Big/ \sum_{k=0}^{\min(k_{\max}, t_{\max}-t^*)} cc^{p^*m^*}_{bk} \right\}$. (4.38)

3 Calculate lb depending on the gene position:

- If $m^* \ge m_{\max} + 1$, then $lb := 0$, (4.39)

- If $1 \le m^* \le m_{\max}$, then $lb := \min_{b=1,..,b_{m,\max}} \left\{ \left(Load^{(2)}_{m^*bt^*} - C^{\min}_{m^*bt^*} \right) \Big/ \sum_{k=0}^{\min(k_{\max}, t_{\max}-t^*)} cc^{p^*m^*}_{bk} \right\}$. (4.40)

4 Choose $\mu \sim U[0,1]$ and $r \sim U[0,1]$. (4.41)

5 Determine the new value of the selected gene by:

$$\mathbf{C}^{(mut)} := \begin{cases} \mathbf{C}_{p^*m^*t^*} + \Delta(\eta, ub - \mathbf{C}_{p^*m^*t^*}), if\ \mu \le 0.5 \\ \mathbf{C}_{p^*m^*t^*} - \Delta(\eta, \mathbf{C}_{p^*m^*t^*} - lb), otherwise \end{cases}, \text{ where } \Delta(\eta, \theta) := \theta \cdot r \cdot \left(1 - \frac{\eta}{\tau}\right)^b . \quad (4.42)$$

6 $\mathbf{C}_{p^*m^*t^*} := \mathbf{C}^{(mut)}$. (4.43)

7 Update other related variables depending on the gene position:

- If $m^* = m_{\max} + 2$, then update $\mathbf{C}_{p^*,m_{\max}+1,t^*}$ based on equation (4.20),

- If $m^* = m_{\max} + 1$, then update $\mathbf{C}_{p^*,m_{\max}+2,t^*}$ as follows:

$$\mathbf{C}_{p^*,m_{\max}+2,t^*} = s^{fo}_{p^*t^*} := \min\left\{ d^{fo}_{p^*t^*} + B_{p^*t^*-1}, \sum_{m=1}^{m_{\max}} \mathbf{C}_{p^*mt^*} + \sum_{m=1}^{m_{\max}} x^i_{p^*mt^*} + I_{p^*t^*-1} - \mathbf{C}_{p^*,m_{\max}+1,t^*} \right\}, \quad (4.44)$$

- If $1 \le m^* \le m_{\max}$ then update $\mathbf{C}_{p^*,m_{\max}+1,t^*}$ and $\mathbf{C}_{p^*,m_{\max}+2,t^*}$ based on expressions (4.19) and (4.20).

As one can see, equations (4.36) and (4.37) are similar to expressions (4.19) and (4.20), respectively, except that the sales quantities referring to supply reservations are defined independently from the sales quantities related to firm customer orders so that it increases the possible room for improvement obtained by the mutation. In fact, it can be advantageous

to decrease s_{pt}^{fo} in favor of s_{pt}^{sr}, especially if the revenues are higher than the backlog costs. Equation (4.38) allows finding the bottleneck with the lowest remaining capacity. For this, the sum of load for all products as defined by formula (4.35) is subtracted from the maximum capacity limit, and the capacity is converted into pieces by means of the capacity consumption. In the same way, a lower bound lb is defined. If the gene to be mutated refers to a sales quantity, lb is derived from constraint (3.7), whereas lb ensures that the minimum capacity threshold as defined in constraint (3.5) is respected when a production quantity has to be changed (cf. Step 3 in Algorithm 4.7). Based on Steps 2 and 3, an upper bound ub and a lower bound lb are obtained for the value of the gene $\mathbf{C}_{p^*m^*t^*}$. The quantity τ denotes a predefined maximum number of generations, i.e., $\tau := gen_{\max}$, and η refers to the current generation. The parameter b is a calibration factor. The function $\Delta(\eta,\theta)$ returns a value in the range $[0,\theta]$ such that the probability of $\Delta(\eta,\theta)$ being close to 0 increases as the number of completed generations η goes up. Hence, it ensures that the mutation operator searches the space uniformly at the beginning of GA-MPSC and very locally at later stages. A higher value of b leads to a lower diversifying effect of the dynamic mutation on the search space with an increasing number of generations (cf. Steps 4-6 in Algorithm 4.7). Since the production quantities and the sales amount encoded in chromosome $\mathbf{C}$ are interlinked, changing the value of a gene also affects the other variables. Thus, a final step is required to guarantee the feasibility of the mutated solution (cf. Step 7 in Algorithm 4.7).

4.3.5 Improving Performance by Using Local Search

In order to enhance the intensification of the search, a local search algorithm is implemented. It takes place for LS_{init} chromosomes after generating the initial population of GA-MPSC. It also occurs regularly during the GA, i.e., every LS_{gen} generations for the LS_{chr} fittest chromosomes. The frequency and extent of the local search scheme have to be appropriately chosen to find the best trade-off between computational effort and speed of convergence.

The local search routine that is implemented consists of swapping allocated production quantities to satisfy supply reservations rather than firm customer orders whenever it improves the objective function value. It is assumed that large revenues lead to large backlog costs. A subset P_1 of products with large revenues and at the same time large backlog costs is considered as well as a subset P_2 of products with low revenues and low backlog costs. The size of the subsets P_1 and P_2 is $\lfloor p_{\max}/2 \rfloor$ and $\lceil p_{\max}/2 \rceil$, respectively. At the beginning of GA-MPSC two ranked lists, i.e., $L^{rev,mc}$ and $L^{udc,mc}$, are established such that the products of subset P_1 are sorted according to descending $\overline{rev}_p$ values and ascending $\overline{mc}_p$ values in $L^{rev,mc}$, while the products of subset P_2 are sorted according to ascending $\overline{udc}_p$ values and descending $\overline{mc}_p$ values in $L^{udc,mc}$. Average variable manufacturing costs that are given by expression (4.45) are used as a tie breaker in case of identical revenues or backlog costs.

$$\overline{mc}_p = \frac{1}{m_{\max} t_{\max}} \sum_{m=1}^{m_{\max}} \sum_{s=1}^{t_{\max}} mc_{pms} . \tag{4.45}$$

If two products cannot be sorted with respect to these criteria in either of the lists, their ranking order is randomly chosen. The sorted lists help to select appropriate products for the quantity swapping.

The local search procedure performs a number of attempts $\theta_{LS} = \lceil p_{\max}/5 \rceil$ for each wafer fab m and each period t. At each trial, two products p_1 and p_2 are randomly selected among the first half of the first and the second list, respectively. It is not allowed for products to be chosen more than once for the same combination of m and t. The quantity to swap $\tilde{x}$ is defined in such as way that the quantities of the chromosome are not changed to a large extent. It is chosen as follows:

$$\tilde{x} \sim U\left[0, 0.1 min\left(\mathbf{C}_{p_2 mt}, \mathbf{C}_{p_2,m+2,t}, d^{sr}_{p_1 t} - \mathbf{C}_{p_1,m+1,t}\right)\right]. \tag{4.46}$$

The quantities $\mathbf{C}_{p_1 mt}$ and $\mathbf{C}_{p_1,m+1,t}$ are increased by $\tilde{x}$, and the quantities $\mathbf{C}_{p_2 mt}$ and $\mathbf{C}_{p_2,m+2,t}$ are reduced by $\tilde{x}$. Given the presence of the term $d^{sr}_{p_1 t} - \mathbf{C}_{p_1,m+1,t}$ in expression (4.46), it is ensured that the constraint (3.4) is fulfilled even after the quantity swap. If the modification improves the objective function value, it is accepted, otherwise it is discarded.

Additionally, a criterion based on threshold accepting (cf. Dueck and Scheuer, 1990; Moscato and Fontanari, 1990) is used to avoid obtaining only a local maximum. Therefore, a modification decreasing the current objective function value can be accepted if the relative difference with the current objective function value is smaller than a predefined threshold ε_{LS}. At the beginning of GA-MPSC, one allows that the new objective function value is at most by 2% smaller than the incumbent one, i.e., $\varepsilon_{LS} = \varepsilon_{init} = 0.02$. Within each generation of the GA the threshold from the previous generation is decreased, i.e., $\varepsilon_{LS} := \varepsilon_{LS} - \varepsilon_{step}$ with $\varepsilon_{step} = 1.5 \times 10^{-5}$. Given the setting $gen_{\max}$ showed in Subsection 4.4.3, it is ensured that $\varepsilon_{LS} > 0$. Modifications that violate any of the constraints (3.2)-(3.8) are rejected. Infeasible chromosomes resulting from the crossover operator are excluded from the local search algorithm. The local search scheme for a given feasible chromosome $\mathbf{C}$ leads to Algorithm 4.8.

The possibility to swap allocated quantities from earlier time periods has also been investigated. But due to the rather narrow difference between average backlog costs and average revenues, it is unlikely that preproduction and stock building several periods ahead may improve the objective function value. Therefore, quantities are only swapped within the same time period.

Algorithm 4.8: Local search scheme.

1 Build the subsets of products P_1 and P_2.

2 Sort the products of subsets P_1 and P_2 into the lists $L^{rec,mc}$ and $L^{udc,mc}$, respectively.

3 $m = 0$, $t = 0$.

4 $m := m + 1$.

5 $t := t + 1$.

6 Set the selection flags of the products of both lists to FALSE.

7 For $i := 1$ to θ_{LS}.

8 Choose two products p_1 and p_2 whose selection flag is equal to FALSE among the first half of $L^{rec,mc}$ and $L^{udc,mc}$, respectively.

9 Set the selection flags of p_1 and p_2 to TRUE.

10 Choose the quantity to be swapped $\tilde{x}$ according to expression (4.46).

11 Set the following quantities:

$$\hat{\mathbf{C}}_{p_1mt} := \mathbf{C}_{p_1mt} + \tilde{x},\ \hat{\mathbf{C}}_{p_1,m+1,t} := \mathbf{C}_{p_1,m+1,t} + \tilde{x},\ \hat{\mathbf{C}}_{p_2mt} := \mathbf{C}_{p_2mt} - \tilde{x},$$
$$\hat{\mathbf{C}}_{p_2,m+2,t} := \mathbf{C}_{p_2,m+2,t} - \tilde{x}. \qquad (4.47)$$

12 Derive the related decision variables $\hat{u}_{p_1mt}$, $\hat{\mathrm{B}}_{p_2t}$. The inventory level stays as-is.

13 If $f_{pen}(\hat{\mathbf{C}}) > f_{pen}(\mathbf{C})$ or $1 - \left| f_{pen}(\hat{\mathbf{C}}) \right| / f_{pen}(\mathbf{C}) \le \varepsilon_{LS}$ and $f_{pen}(\hat{\mathbf{C}}) > 0$, then perform the quantity swap as follows:

$$\mathbf{C}_{p_1mt} := \hat{\mathbf{C}}_{p_1mt},\ \mathbf{C}_{p_1,m+1,t} := \hat{\mathbf{C}}_{p_1,m+1,t},\ \mathbf{C}_{p_2mt} := \hat{\mathbf{C}}_{p_2mt},\ \mathbf{C}_{p_2,m+2,t} := \hat{\mathbf{C}}_{p_2,m+2,t},$$
$$u_{p_1mt} := \hat{u}_{p_1mt},\ \mathrm{B}_{p_2t} := \hat{\mathrm{B}}_{p_2t}. \qquad (4.48)$$

14 End For.

15 If $t < t_{\max}$, then go to Step 5.

16 If $m < m_{\max}$, then set $t = 0$ and go to Step 4.

4.4 Static Performance Assessment of the Heuristic Solution Approaches

First, the assessment methodology is explained, and then the implementation of the different methods is described. Afterwards, details of the parameter settings are given and the scheme used to generate the problem instances is introduced. Finally, the results of computational experiments are presented and discussed.

4.4.1 Assessment Methodology

The performance of the suggested heuristic approaches PD-MPSC and GA-MPSC are assessed by comparing the corresponding results with the results of a commercial MIP solver after a predefined computing time (e.g., 30 minutes, 2 hours) for different randomly generated problem instances. The MIP solver is based on a branch-and-bound algorithm. It is called BB-MPSC throughout the rest of the paper when it is used for solving instances of MPSC. The comparison between the three methods, i.e., PD-MPSC, GA-MPSC, and BB-MPSC, is performed with respect to both solution quality and computing time.

4.4.2 Implementation of the Solution Approaches

The BB-MPSC and the PD-MPSC procedures are performed by using the commercial MIP solver ILOG CPLEX 11.1. The GA-MPSC scheme is implemented by means of the object-oriented framework GALib 2.47 using the C++ programming language (cf. Wall, 2012). The chromosome representation presented in Subsection 4.3.2 is coded by using the template class GA3DArrayGenome. All algorithms are tested on a computer equipped with a 1.7 GHz Intel Pentium M processor and 1.0 GB memory.

4.4.3 Parameter Settings

Default settings of ILOG CPLEX 11.1 are used for solving problem instances with BB-MPSC and PD-MPSC. Extensive computational experiments in combination with a trial and error strategy are carried out to select the best parameter combination for GA-MPSC. The parameter settings of GA-MPSC and PD-MPSC are summarized in Table 4.1.

Table 4.1: Parameter settings of GA-MPSC and PD-MPSC.

Parameter Settings for GA-MPSC	
Population size	$ps = 200$
Maximum number of generations	$gen_{\max} = 1000$
Diversity threshold	$div_{\min} = 0.001$
Crossover probability	$p_{crossover} = 0.80$
Mutation probability	$p_{mutation} = 0.07$
Replacement probability	$p_{replacement} = 0.50$
Upper bounds for arithmetical crossover	$K_1 = 1.25$, $K_1 = 1.50$
Calibration factor for dynamic mutation	$b = 1.50$
Maximum attempts for the heterogeneity criterion	$try_{\max} = 12$
Minimum number of chromosomes with a large distance to current population	$n_{chr} = 7$
Occurrence of the local search procedure (in generations)	$LS_{gen} = 100$
Number of chromosomes from the initial population concerned by the local search	$LS_{init} = 100$
Number of fittest chromosomes concerned by the local search during GA-MPSC	$LS_{chr} = 10$
Initial deterioration threshold for the local search	$\varepsilon_{init} = 0.02$
Step for decrementing the deterioration threshold	$\varepsilon_{step} = 1.5 \times 10^{-5}$
Parameter Settings for RA-MPSC	
Maximum number of past time periods considered for preproduction	$n_{preprod} = 5$
Parameter Settings for PD-MPSC	
Number of products per subproblem	4

A rather large value is chosen for the mutation probability in GA-MPSC. This is necessary to ensure that the diversity within the population does not decrease too quickly and to avoid being able to find only a local maximum. The number of products per subproblem in PD-MPSC is determined by searching the factor combinations from which the MIP solver can no longer find an optimal solution in a small amount of time, i.e., less than 2 minutes. It turns out that only instances with at maximum four products can be optimally solved within the given amount of time (cf. Table 4.7).

4.4.4 Design of Experiments

The used design of experiments is summarized in Table 4.2.

Table 4.2: Design of experiments (I).

Factor	Notation	Level	Number
Number of time periods	$t_{\max}$	26	1
Number of products	$p_{\max}$	50, 100, 200	3
Number of wafer fabs	$m_{\max}$	8, 10, 10, 12	
- In-house locations	$ih_{\max}$	6, 8	2
- Silicon foundries	$sf_{\max}$	2, 4	2
Number of bottlenecks per location	$b_{m,\max}$	1	1
Firm customer orders	d_{pt}^{fo}	$U[200m_{\max}, 300m_{\max}]$	1
Supply reservations	d_{pt}^{sr}	$U[200m_{\max}, 300m_{\max}]$	1
Initial inventory	I_{p0}	$500m_{\max} / p_{\max}$	1
Initial backlog	B_{p0}	$250m_{\max} / p_{\max}$	1
WIP - In-house locations - Silicon foundries	x_{pmt}^{i}	$400/p_{\max}$ $200/p_{\max}$	1
Maximum available capacity - In-house locations - Silicon foundries	$C_{mbt}^{\max}$	6720 hours 2000 wafers	1
Minimum capacity limit	$C_{mbt}^{\min}$	$C_{mbt}^{\min} = 0.80C_{mbt}^{\max}$	1
Capacity consumption - In-house locations - Silicon foundries	cc_{bk}^{pm}	2 hours/wafer 1 wafer	1
Product lead time	$q = k_{\max} + 1$	6 periods	1
Variable manufacturing cost - In-house locations - Silicon foundries	mc_{pmt}	Each range is for 50% of the products $U[10,20]$, $U[20,40]$ $U[30,40]$, $U[40,60]$	1
Fixed location cost - In-house locations - Silicon foundries	lc_{pmt}	$U[375,625]$ $U[625,1250]$	1
Inventory holding cost	hc_{pt}	$U[5,10]$	1
Cost due to unmet demand	udc_{pt}	Each range is for 50% of the products $U[200,240]$, $U[300,400]$	1
Revenue for fulfilling supply reservation	rev_{pt}	Each range is for 50% of the products $U[80,120]$, $U[150,200]$	1
Total parameter combinations			12
Number of problem instances per combination			20
Total number of problem instances			240

A factorial design with two factors is used for the purpose of generating problem instances to compare the different algorithms. The number of products $p_{\max}$ and the number of wafer fabs $m_{\max}$, i.e., the sum of the number of in-house locations $ih_{\max}$ and the number of silicon foundries $sf_{\max}$, are selected as factors. Each of these factors is varied at three and four levels, respectively. Thus, twelve possible factor combinations are obtained. For each factor combination, twenty independent instances are generated and solved by the MIP solver and

the heuristics PD-MPSC and GA-MPSC. Thus, each method is applied to 240 problem instances. In addition, GA-MPSC is performed ten times for each instance with different seed values and also ten independent runs of the repair loop are carried out for PD-MPSC.

The levels of the MPSC factors are empirically determined based on the example of Infineon Technologies. However, these levels are not real-world values because of the complex data extraction and data privacy, but they reliably reflect the network structure, demand levels, available capacity, and cost configuration of a medium-sized semiconductor manufacturer like Infineon. Although the costs in MPSC are defined as time-dependent, for simplicity reasons it is assumed that they are constant over time for the computational experiments.

4.4.5 Results of Computational Experiments

4.4.5.1 Presentation of Results

The solution quality is measured as the ratio of the objective function value for both heuristic algorithms and the corresponding objective function value obtained with BB-MPSC. The measure assessing the performance of PD-MPSC is called PD/BB ratio, whereas the measure related to GA-MPSC is named GA/BB ratio. The ratio values represent the decrease or increase in percent of the objective function value by using a heuristic algorithm instead of the optimum solution procedure. All result ratios are grouped according to the factor levels of the design of experiments. The best, average, and worst ratio values as well as the confidence intervals denoted by CI for the average are showed in Table 4.3. The corresponding computing times are presented in Table 4.6.

The branch-and-bound algorithm is exact if the enumeration is complete, and provides only an approximate solution if the enumeration is truncated, e.g., after a certain amount of time. In the latter case, the quality of the solution is measured by the so-called MIP gap. It is the difference between 1 and the ratio of the objective function value of the best feasible solution found during the branch-and-bound procedure and the objective function value of the best node among the remaining solutions not necessary feasible that have not been explored yet. This percentage indicates the maximum relative deviation from optimality of the best feasible solution found (cf. Pochet and Wolsey, 2006). The average values of the MIP gap given by the MIP solver are also showed in Tables 4.3, 4.7, and 4.8.

4.4.5.2 Comparison of Solution Quality

Table 4.3 presents the grouped best, average, and worst result ratios for solving problem instances of MPSC. The average MIP gap is around 2.30% and 10.96% for problem instances with 50 and 100 products, respectively. In both cases, the product-based decomposition scheme achieves results that are very close to the results obtained by BB-MPSC. In fact, the average PD/BB ratio shows an average decrease of the objective function value of around 2.23%. In addition, the results of PD-MPSC are highly homogenous since the differences between worst and best objective function values are never higher than 3.50%. The GA reaches on average 92.87% and 90.66% of the objective function values obtained by BB-MPSC for problem instances with 50 and 100 products, respectively. The best average GA/BB ratio is achieved for instances with 50 products, six in-house locations, and four silicon foundries by a value of 96.82%. The results of GA-MPSC have a mean span of 7.47% between worst and best outcomes, and the maximum difference reaches 12.87%.

The average MIP gap for instances with 200 products is 57.57%. This indicates that the MIP solver is unable to deliver high-quality solutions within the given amount of time due to

the increased problem size. In fact, PD-MPSC on average outperforms BB-MPSC for large-scale problem instances with 200 products. The PD-MPSC scheme reaches a mean performance of 111.96%, whereas the mean GA/BB ratio is around 99.75%, i.e., GA-MPSC and BB-MPSC obtain on average very similar results.

Columns 4 and 7 in Table 4.3 show the 95% confidence intervals for the mean of PD/BB and GA/BB ratios, respectively. The rather narrow intervals indicate that differences between both heuristics are systematic, given the chosen level of confidence, since no overlapping occurs. The results of the Wilcoxon signed-rank test (cf. Wilcoxon, 1945) with a significance level of 1% are showed in Table 4.4. With respect to the objective function values achieved by the different algorithms, one observes that BB-MPSC > PD-MPSC > GA-MPSC for instances with 50 and 100 products, and PD-MPSC > BB-MPSC > GA-MPSC for instances with 200 products.

Table 4.3: Average ratio values and confidence intervals of PD/BB and GA/BB with a level of confidence of 95%, minimum and maximum ratio values, and average MIP gaps for different factor levels as defined in Table 4.2.

Products	Locations		PD/BB Ratio			GA/BB Ratio			Average
p_{max}	ih_{max}	sf_{max}	CI	Min	Max	CI	Min	Max	MIP Gap
50	6	2	0.9775 ±0.0028	0.9656	0.9877	0.9151 ±0.0038	0.8816	0.9476	0.0318
50	6	4	0.9782 ±0.0026	0.9701	0.9897	0.9682 ±0.0025	0.9489	0.9843	0.0216
50	8	2	0.9774 ±0.0027	0.9656	0.9883	0.9164 ±0.0028	0.8802	0.9462	0.0200
50	8	4	0.9753 ±0.0037	0.9590	0.9895	0.9152 ±0.0027	0.8776	0.9518	0.0185
100	6	2	0.9824 ±0.0041	0.9643	0.9984	0.9023 ±0.0030	0.8823	0.9300	0.1441
100	6	4	0.9743 ±0.0026	0.9656	0.9875	0.9044 ±0.0044	0.8689	0.9609	0.1099
100	8	2	0.9807 ±0.0025	0.9702	0.9884	0.9190 ±0.0036	0.8821	0.9696	0.1056
100	8	4	0.9762 ±0.0030	0.9640	0.9869	0.9007 ±0.0061	0.8087	0.9374	0.0786
200	6	2	1.1343 ±0.0156	1.0827	1.1968	1.0137 ±0.0054	0.9728	1.0718	0.7361
200	6	4	1.1259 ±0.0079	1.0958	1.1659	1.0100 ±0.0088	0.9850	1.0413	0.6879
200	8	2	1.1092 ±0.0078	1.0900	1.1487	0.9828 ±0.0038	0.9536	1.0191	0.4781
200	8	4	1.1090 ±0.0093	1.0612	1.1433	0.9836 ±0.0041	0.9650	1.0146	0.4008
Overall			1.0250 ±0.0105	1.0045	1.0476	0.9443 ±0.0033	0.9089	0.9812	0.2361

Table 4.4: Results of the Wilcoxon signed-rank test with the 1% significance level.

Products	Locations		PD-MPSC vs. BB-MPSC	GA-MPSC vs. BB-MPSC	PD-MPSC vs. GA-MPSC
p_{max}	ih_{max}	sf_{max}			
50	6	2	<	<	>
50	6	4	<	<	>
50	8	2	<	<	>
50	8	4	<	<	>
100	6	2	<	<	>
100	6	4	<	<	>
100	8	2	<	<	>
100	8	4	<	<	>
200	6	2	>	<	>
200	6	4	>	<	>
200	8	2	>	<	>
200	8	4	>	<	>

Remark: '>' indicates that the first algorithm mentioned in the first row performs significantly better than the second mentioned algorithm; '<' indicates the opposite.

An analysis of variance (ANOVA) is performed to investigate the influence of the design factors, i.e., the number of products, the number of wafer fabs, and the planning heuristics denoted by $Algo$, on the solution quality relatively to the BB-MPSC values. Normal probability plots of residuals and plots of residuals versus fitted values were used to check the model adequacy as suggested in Montgomery (2008). Departures from normality and from constant variance are moderate.

Table 4.5 shows the main and two-way interaction effects of the independent variables $p_{\max}$, $m_{\max}$, and $Algo$ at a 5% significance level. The column DF gives the number of degrees of freedom. The sum of squares SS, the mean square MS, and the F value are indicated as well as the probability that the factor has no effect, i.e., Pr>F. It turns out that both the number of products and the heuristics have an impact on the solution quality. Also, there is an interaction between the number of products and the heuristics.

Furthermore, an instance factor is included as a nested effect within the problem factors to account for blocking on problem instances. This nested factor cannot interact with the problem characteristics, because it is not comparable across different problem levels. It is assumed that it does not interact with both planning heuristics in order to have an estimate of random error (cf. Rardin and Uzsoy, 2001).

Table 4.5: Main and two-way interaction effects as provided by the ANOVA procedure (I).

Source	DF	SS	MS	F	Pr>F
$p_{\max}$	2	1.3200	0.6600	286.28	2.1E-82
$m_{\max}$	3	0.0197	0.0066	1.30	0.2731
$Algo$	1	0.7827	0.7827	228.53	1.8E-42
Inst($p_{\max}$ * $m_{\max}$)	114	1.0781	0.0095	4.17	0.0006
$Algo$ * $p_{\max}$	2	0.1135	0.0567	132.16	2.4E-46
$Algo$ * $m_{\max}$	3	0.0100	0.0033	0.98	0.4029
$p_{\max}$ * $m_{\max}$	6	0.0308	0.0051	2.29	0.0344
Error	348	2.3705	0.0068	-	-

4.4.5.3 Comparison of Computing Times

Besides the solution quality, the performance of both heuristic approaches is assessed with respect to computing time. One can argue that time is not a critical parameter for mid-term planning activities that are weekly executed; on the contrary it is expected that short computing times increase the acceptance of supply chain managers and planners for those methods, and the optimization algorithms can be more easily implemented in planning processes when they quickly provide results.

As showed in Table 4.6, the predefined maximum computing times are 30 minutes for BB-MPSC and 10, 15 and 30 minutes for PD-MPSC for instances with 50, 100 and 200 products, respectively. On the contrary, as explained in Subsection 4.3.1 GA-MPSC terminates whenever one of the termination criteria is fulfilled. As one can see GA-MPSC is the fastest algorithm as it is performed in 6.58 minutes on average. For large-scale problem instances, it needs only one third of the time required by the other methods. The computing time of GA-MPSC increases with the number of products. In fact, the extended chromosome structure raises the time consumption of initialization and evaluation procedures.

Table 4.6: Average computing times (in minutes) of PD-MPSC, GA-MPSC, and BB-MPSC for different factor levels as defined in Table 4.2.

Products	Locations		Avg. Computing Time (in Minutes)		
p_{max}	ih_{max}	sf_{max}	PD-MPSC	GA-MPSC	BB-MPSC
50	6	2	10	3	30
50	6	4	10	3	30
50	8	2	10	4	30
50	8	4	10	4	30
100	6	2	15	6	30
100	6	4	15	6	30
100	8	2	15	7	30
100	8	4	15	7	30
200	6	2	30	9	30
200	6	4	30	10	30
200	8	2	30	10	30
200	8	4	30	10	30

4.4.5.4 Additional Experiments

In order to further assess the performance of both heuristics, additional experiments are carried out for a reduced number of small- and large-scale problem instances. Small-size instances are generated with respect to the design of experiments summarized in Table 4.2. The maximum available capacity is reduced according to the number of products, i.e., 350 and 700 in-house hours as well as 100 and 200 maximum outsourced wafers, respectively, for instances with 1 to 5 products, and 10 products. For small- and large-scale problems, ten and twenty independent instances are solved respectively for each factor combination. The average values are presented and discussed in the following.

Table 4.7 shows that instances with a maximum of four products can be optimally solved in a few seconds by the MIP solver. One can see an increase of hardness for instances with at least five products where PD-MSC achieves high-quality results in much less time than BB-MPSC, i.e., the average computing times of PD-MPSC are below one minute, while BB-MPSC runs for 30 minutes. For small-size problem instances GA-MPSC cannot outperform the MIP solver in terms of time, but it does when the problem size increases. Its average solution quality is also high, i.e., 0.9922.

Table 4.7: Solution quality and average computing times (in seconds) of PD-MPSC and GA-MPSC, and average MIP gaps for small-size problem instances.

Products	Locations		Average	Average	Average	Avg. Computing Times (in Seconds)		
p_{max}	ih_{max}	sf_{max}	PD/BB Ratio	GA/BB Ratio	MIP Gap	PD-MPSC	GA-MPSC	BB-MPSC
1	6	2	1.0000	0.9996	0.0000	5	22	5
1	8	4	1.0000	0.9991	0.0000	5	27	5
2	6	2	1.0000	0.9988	0.0000	5	39	5
2	8	4	1.0000	0.9982	0.0000	5	41	5
3	6	2	1.0000	0.9972	0.0000	6	52	6
3	8	4	1.0000	0.9979	0.0000	6	49	6
4	6	2	1.0000	0.9905	0.0000	7	61	7
4	8	4	1.0000	0.9901	0.0000	7	58	7
5	6	2	0.9992	0.9856	0.0022	32	70	1800
5	8	4	0.9993	0.9878	0.0027	32	72	1800
10	6	2	0.9951	0.9804	0.0056	54	119	1800
10	8	4	0.9946	0.9811	0.0067	54	127	1800
Overall Average			0.9990	0.9922	0.0014	18.17	61.42	603.83

Table 4.8 presents the PD/BB and GA/BB ratio values for a reduced number of large-scale problem instances, i.e., with 200 products as described in Table 4.2, when the BB-MPSC runs two hours. Although the average MIP gap is still high, the extended computing time allows the MIP solver to achieve slightly better objective function values. Thus, the performance of both heuristics decreases compared to Table 4.3, but the average solution quality is still high, i.e., 1.0633 and 0.9549 for PD-MPSC and GA-MPSC, respectively.

Table 4.8: Solution quality of PD-MPSC and GA-MPSC, and average MIP gaps for large-scale problem instances.

Products	Locations		PD/BB Ratio			GA/BB Ratio			Average
p_{max}	ih_{max}	sf_{max}	Max	Avg.	Min	Max	Avg.	Min	MIP Gap
200	6	2	1.1601	1.0788	1.0322	1.0577	0.9701	0.9234	0.3257
200	6	4	1.0622	1.0401	1.0045	0.9701	0.9377	0.9081	0.2889
200	8	2	1.0855	1.0687	1.0399	0.9899	0.9582	0.9256	0.2194
200	8	4	1.0898	1.0655	1.0401	0.9709	0.9534	0.9408	0.2077
Overall Average			1.0994	1.0633	1.0292	0.9972	0.9549	0.9245	0.2604

Remark: Computing time of BB-MPSC: 2 hours; Computing times of PD-MPSC and GA-MPSC: cf. Table 4.6.

4.5 Conclusion

As a consequence of the NP-hardness of MPSC, heuristic solution approaches were proposed, i.e., a product-based decomposition scheme (PD-MPSC), a rule-based assignment scheme (RA-MPSC), and a meta-heuristic (GA-MPSC). The RA-MPSC scheme is used as an initialization scheme in GA-MPSC. Computational experiments were accomplished to assess the performance of PD-MPSC and GA-MPSC. An exact solution procedure (BB-MPSC) that is based on a branch-and-bound algorithm serves as a benchmark approach. It can be concluded that PD-MPSC provides excellent results in term of solution quality compared to BB-MPSC, whereas GA-MPSC achieves a reasonable solution quality and clearly outperforms the other methods with respect to computing time. Parts of the results presented in this chapter are already published in Ponsignon and Mönch (2012a).

5. Simulation-based Performance Assessment of the Heuristic Solution Approaches of MPSC

The performance of the heuristic solution approaches of MPSC described in the previous chapter has been assessed so far by means of single problem instances. However, given the uncertainty that is typical for the semiconductor industry, there is a need for incorporating different sources of disruptions into the evaluation of the approaches. In addition, the master plan is re-planned on a weekly basis in real-world situations. It allows for taking the current state of the input parameters into account. The investigation of the resulting rolling plans may provide further insights into the performance of the planning approach used.

The intention of this chapter is to propose an appropriate simulation-based framework for the performance assessment of MP approaches in semiconductor manufacturing networks and to apply the framework to compare GA-MPSC and RA-MPSC in a rolling horizon setting taking both demand uncertainty and disruptions in a wafer fab network into account.

5.1 Literature Review

Simulation-based frameworks are often used to investigate the performance of production control approaches (cf. Mönch *et al*., 2003; Mönch, 2007). The evaluation of scheduling and dispatching strategies is in the focus of many researchers using discrete-event simulation. Some approaches involve a rolling horizon setting. However, the interaction between the planning algorithms and the execution in the production system was predominantly neglected so far. This section discusses previous work related to the performance assessment of planning approaches.

One stream of research refers to the performance assessment of planning algorithms based on the analysis of single problem instances. The papers by Barahona *et al*. (2005) and Zobolas *et al*. (2008) belong to this category. Usually, this approach does not adequately capture the dynamic behavior of the market demand and the execution system.

Another stream of research consists in embedding the planning approach in a rolling horizon setting by means of simulation. System Dynamics (cf. Kleijnen, 2005) and discrete-event simulation (cf. Horiguchi *et al*., 2001; Chong *et al*., 2006) are widely used techniques for this. Both techniques allow for modeling the time-dependent behavior of the execution

system. Venkateswaran and Son (2005) introduce a hierarchical production planning model that explicitly differentiates between the planning and control algorithms using System Dynamics and discrete-event simulation, respectively. The High-Level Architecture (HLA) ensures the synchronization between both simulation models. Nevertheless, none of these papers provides a framework that interlinks the planning, control, and execution level since the planning algorithms are implemented directly in the simulation software.

A series of papers focuses on the impact of uncertain inputs on the planning algorithm. Spitter (2004) incorporates supply chain operations planning approaches in a rolling horizon environment using discrete-event simulation. The influence of lead times on the performance of mathematical programming models while considering demand uncertainty is discussed. Mula *et al*. (2006) review production planning approaches under environment- and system-related uncertainties along with the modeling techniques. Lin (1989), Russell and Urban (1993), Venkataraman and Nathan (1999), Tang and Grubbström (2002), Xie *et al*. (2004), Brandimarte (2006), Huang *et al*. (2007), and Robinson *et al*. (2007) investigate the influence of forecasting errors on rolling production plans using simulation for different lengths of the planning horizon, different lengths of the frozen interval (cf. Subsection 5.2.1.3), or different re-planning frequencies. Genin *et al*. (2007) suggest a re-planning policy to achieve stable and robust tactical decisions in the context of APS. Kimms (1998) and Zhao *et al*. (2001) address the question of stability of planning approaches in a rolling horizon environment. However, the models used do not incorporate the execution level. As a result, the performance measurement is limited to the outcomes of the planning algorithm. However, there is a strong need for a broader approach that allows for simulating the realized production planning results.

Mönch (2007) proposes a simulation-based framework for the assessment of production control approaches. The execution system is represented by a simulation model. The framework is extended by Mönch (2008) to the assessment of production planning approaches. In this chapter, an extended and refined version of the latter framework is proposed, especially with respect to the generation of uncertain demand, the consideration of disruptions in the production system, and the performance assessment methodology.

5.2 Simulation-based Framework

This section starts with an overview of the proposed simulation-based framework. Then, its components are described. Important steps for the application of the framework are presented. Finally, the implementation of the framework is discussed.

5.2.1 Overview of the Framework

5.2.1.1 Structural Design of the Framework

The framework is structured into the planning, control, and base level. At the base level, it is differentiated between the base system and the base process. On the one hand, the base system comprises static objects from the manufacturing system that describe the resources, e.g., machines and work centers. On the other hand, the base process states how the resources are used by working objects such as the lots. The manufacturing routes in the base process describe the sequences of process steps to go through to achieve finished products. The control process specifies the rules related to the production scheduling and dispatching to sequence the lots on a machine. The control instructions *cs* are a result of the control process.

The control process is executed by the control system. The corresponding decisions only affect objects that are already released into the base system. The base system and the control system form the execution system. The planning process establishes both due quantities and planned completion dates in the form of production requests *mp*, i.e., a master plan. The requests are provided to the control process that releases working objects into the base system. The planning process is executed by the planning system. The planning level deals with objects that are not yet in the base system. The interactions between the planning, control, and base level are summarized in Figure 5.1 (cf. Mönch, 2008).

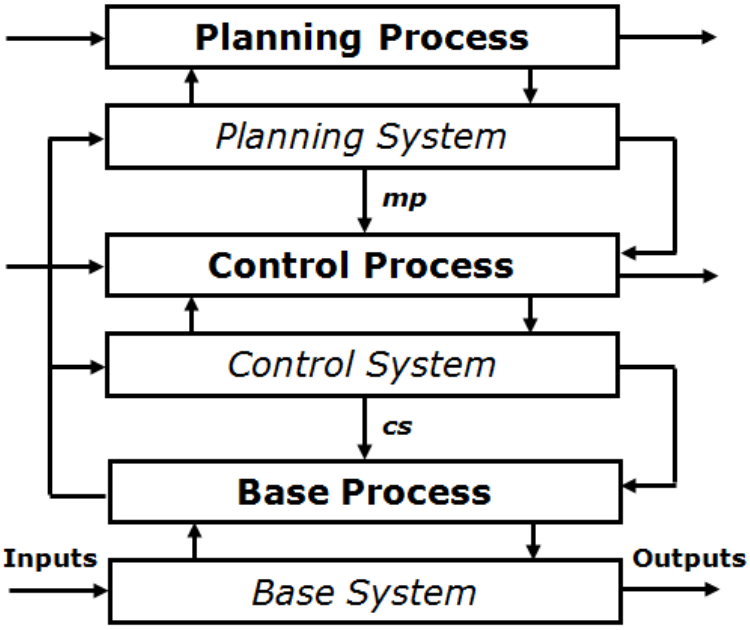

Figure 5.1: Interactions between the planning, control, and base level (cf. Mönch, 2008).

5.2.1.2 Architecture of the Framework

A set of MP algorithms $PA := \{PA_\gamma \mid \gamma = 1,\ldots,\gamma_{\max}\}$ is considered, which are used within the respective planning process PP_γ, and whose performance has to be assessed. The performance measure values depend on the quantities resulting from the planning and base level. A fixed control algorithm used within a fixed control process is assumed. To investigate the influence of the base level on the planning level, several configurations of the base system are considered.

Figure 5.2 shows the architecture of the proposed framework. It consists of a production planning module, a production control module, and a simulation model that is embedded in a simulator. The three components represent the planning, control, and base level, respectively. Further modules are required. A demand generator provides different types of demand. The MP algorithm PA_γ that is implemented in the planning module is evaluated by means of the performance assessment module. A demand fulfillment module is used to convert produced quantities into sold quantities and to update the inventory and backlog values. Finally, a black-board-like data layer allows for data exchange between the modules. The components of the framework are described in more detail in Subsection 5.2.2.

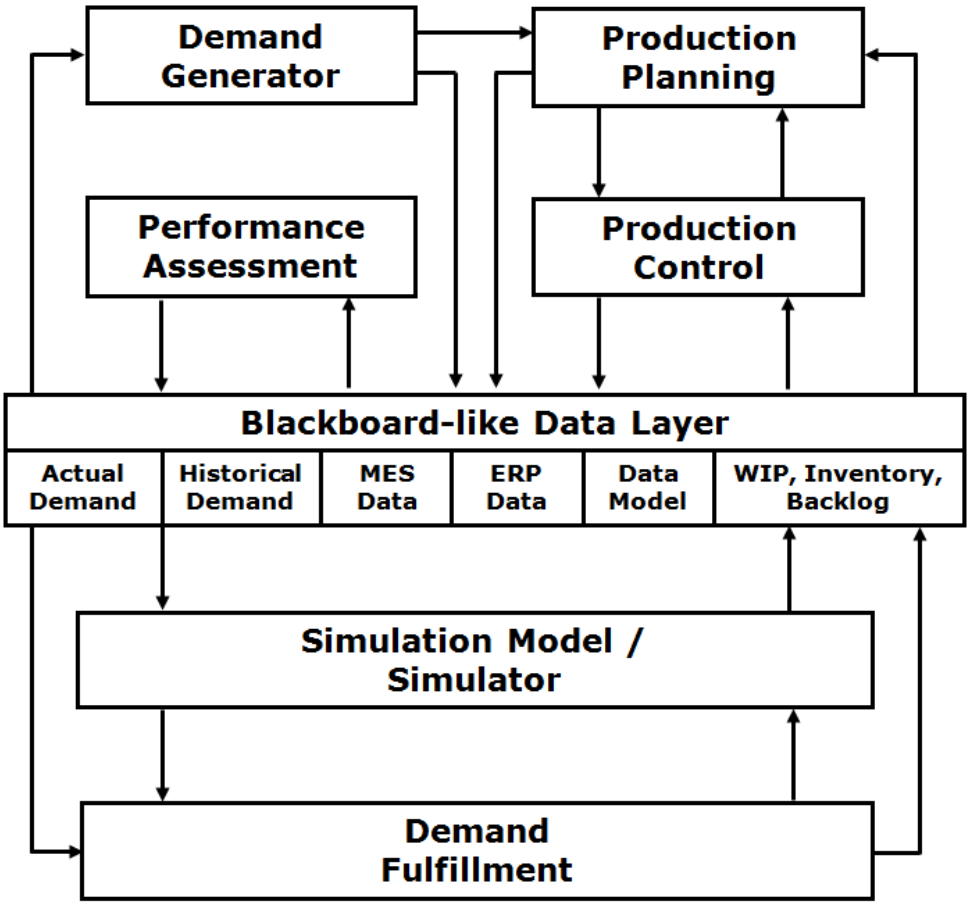

Figure 5.2: Architecture of the proposed simulation-based framework.

5.2.1.3 Rolling Horizon Setting

The framework allows for applying the planning algorithm in a rolling horizon setting. In the following, the related notation is introduced. The simulation time index is denoted by $t_s = 1,\ldots,t_{s,\max}$ with $t_{s,\max}$ being the length of the simulation horizon. The planning system runs regularly along the simulation timeline. After each planning occurrence, a master plan is provided with due quantities for the periods $1,\ldots,t_{\max}$. The planning horizon $t_{\max}$ is assumed to be shorter than the simulation horizon, i.e., $t_{\max} < t_{s,\max}$. The time between two successive planning occurrences is called the re-planning interval, and it is represented by Δt. Overlapping master plans are obtained when $1 \leq \Delta t < t_{\max}$. Throughout this chapter, Δt is considered to be equal to one planning period, i.e., $\Delta t = 1$. Since the lead times are usually longer than one planning period, the production has to be initiated ahead of the planned completion date in a timely manner. As stated in Subsection 3.1.2, MPSC assumes that all products have the same lead time of q planning periods with $1 \leq q < t_{\max}$. The due quantities for the first q periods of each master plan state the WIP in the base system. The frozen interval is defined as the time fence at the beginning of the planning horizon where changes of planned production quantities are not permitted (cf. Robinson *et al.*, 2007). In the considered problem setting, the length of the frozen interval is assumed to be equal to the product lead time. The planning algorithm determines due quantities in the planning interval, i.e., for $t = q+1,\ldots,t_{\max}$. The Δt periods that newly enter the frozen interval between two successive planning occurrences constitute the execution interval. The due quantities in the execution

interval are being assigned a release date into the base system, and the production is initiated accordingly. The index $n = 1,\ldots,n_{\max}$ denotes the planning occurrences with $n_{\max}$ being the total number of master plans. The first period of the n-th master plan is denoted by t_n, e.g., $t_1 = 1, t_2 = 1+\Delta t,\ldots,t_n = 1+(n-1)\Delta t$. It is assumed that the last master plan generation occurs when $t_n = t_{s,\max} - q$. Figure 5.3 summarizes the time intervals in a rolling horizon setting as adopted and modified from Lin (1989).

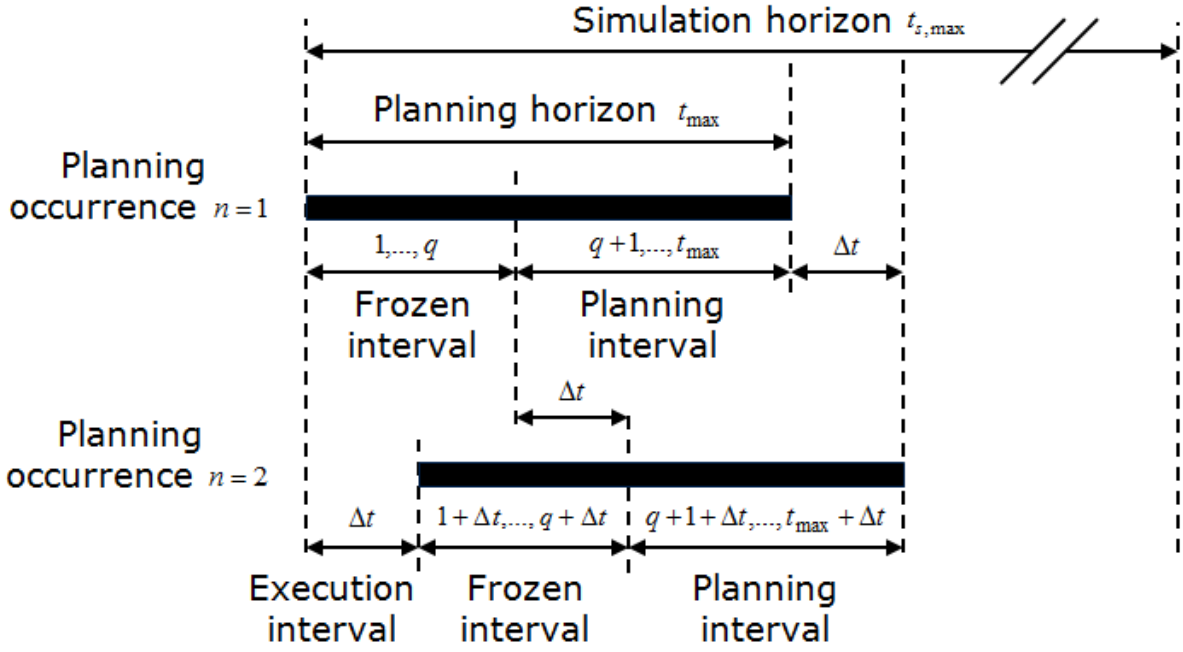

Figure 5.3: Time intervals in a rolling horizon setting.

5.2.2 Components of the Framework

5.2.2.1 Production Planning Module

The MP algorithm is implemented in the production planning module. Once the planning algorithm is applied, production requests are provided to the production control module in the form of finished products that are due at the end of the periods of the planning horizon. The planned quantities are based on current demand, backlog, WIP, and inventory on hand as available in the data layer at the time of the planning occurrence. The planning algorithm uses an aggregated representation of the finite capacity of the base system. A bottleneck mapping specifies the location of the $b_{\max}$ critical resources. A manufacturing time per planning period, i.e., the sum of the available machine-hours, is assigned to each bottleneck work center. The matrices $\tilde{C}$ are used for the calculation of the capacity consumption (cf. Subsection 3.1.2). Based on the manufacturing routes of the products and the fixed product lead time, the matrices can be determined using the flow factor that corresponds to the given lead time. The flow factor is the ratio of the average cycle time and the raw processing time.

5.2.2.2 Production Control Module

The production control algorithm implemented in the framework transforms the production plan that is provided by the planning module into a lot release schedule. The production control module exclusively considers the due quantities for the periods in the execution interval (cf. Figure 5.3). It splits the requested quantities into lots according to the standard lot size ls. The algorithm determines the production release period of each lot by subtracting the product lead time from the due period. The start dates of the lots to be released within the same period are equally spread over the period. After having assigned the start dates, the lots are hold in a virtual lot pool until their release into the base system.

To control the production in the base system, the Earliest Due Date (EDD) dispatching rule is used whose priority index l is expressed as $l_j = d_j$ with d_j the due date of lot $j \in J$ and J the set of lots. The lot with the lowest index value is selected as the next job to process. The First-in, first-out (FIFO) rule is applied as a tie breaker. The corresponding priority index is defined as $l_j = r_j$ with r_j the release time of lot $j \in J$. The lot with the lowest index value is given a higher priority. Hence, the progress of the production in the base system tends to follow the requests from the planning and control systems.

5.2.2.3 Simulation Model

The simulation model represents the execution level. It triggers the production start of the lots according to their release dates as provided by the control module, and it carries out the processing of the lots. The simulation model includes information related to the base system and process. It typically contains a representation of the resources and their characteristics, e.g., the number of parallel machines. The product-machine assignment is given by the manufacturing routes. The framework allows for investigating the influence of the planning level on the base level. Therefore, the model may include stochastic parameters for simulating the impact of disruptions that may occur on the shop-floor such as machine breakdowns.

With respect to the scope of the researched problem, the simulation model reflects the base system of an entire manufacturing network. Since using simulation usually implies a significant effort for creating, maintaining, and running the simulation model (cf. Law and Kelton, 2000; Banks *et al*., 2010), an appropriate trade-off between the level of detail and the computational burden has to be found. A reduction approach of the simulation model is presented in Subsection 5.3.2.

5.2.2.4 Data Layer

The core of the framework is a blackboard-like data layer that plays the role of an interface between the components (cf. Mönch, 2007). It comprises a mirror image of the objects of the base system. The status of the objects is updated by event-driven notifications from the simulator. For instance, as soon as a lot completes a process step and moves to the next one, its position on the manufacturing route is changed accordingly in the data layer. It allows providing the current state of the base system to the planning and control system. The data layer is a virtual representation of the operational information system of the company. It includes data that is typically stored in the Manufacturing Execution System (MES), e.g., manufacturing routes, bottleneck mapping, WIP, and in the ERP system, e.g., the data aggregation structures, the aggregated capacity representation, the current and historical demand, and the current levels of inventory and backlog. Besides the data exchange, it allows keeping track of the outcomes of other modules such as performance measure values and demand fulfillment quantities.

5.2.2.5 Demand Generator

The demand generator provides demand information to the planning module with respect to the following demand types:

- the firm customer order denoted by d^{fo},
- the supply reservations denoted by d^{sr}, and
- the definitive expectations of the customers called final demands and denoted by d^{fi}.

It is assumed that there is no supply reservation in the first planning period.

Similar to the scheme used in Zhao *et al*. (2001) and Xie *et al*. (2004), the final demands are generated as follows:

$$d^{fi}_{pt_s} := \frac{DL}{p_{\max}}\left(1+R_1\right), \forall p = 1,\ldots, p_{\max}, \forall t_s = 1,\ldots, t_{s,\max}, \tag{5.1}$$

with DL being the overall demand level in one period and R_1 a normally distributed random variate such that $R_1 \sim N\left(0,\sigma_1^2\right)$. where σ_1^2 is the variance. The product mix in the base system can be taken into account by replacing $p_{\max}$ in expression (5.1) by a product-dependent percentage. Because of $\Delta t = 1$, the simulation time index t_s in expression (5.1) can be substituted by the planning occurrence index *n* for simplification. The resulting expression is as follows:

$$d^{fi}_{pn} := \frac{DL}{p_{\max}}\left(1+R_1\right), \forall p = 1,\ldots, p_{\max}, \forall n = 1,\ldots, n_{\max}. \tag{5.1'}$$

As the impact of demand inaccuracy on the performance of the planning algorithm is of interest, the parameters ε and η are used to describe the demand bias and demand volatility, respectively, both for firm customer orders and supply reservations. Expressions (5.2) and (5.3) provide firm customer orders and supply reservations for product p in planning period t while performing the n-th planning occurrence based on the previously generated final demands:

$$d^{fo}_{pnt} := \begin{cases} d^{fi}_{pn}\left(1-\dfrac{t-1}{t_{\max}-1}\right)\left(1+\varepsilon^{fo}\right)\left(1+\eta^{fo}\cdot t\cdot R_2\right), & \forall p = 1,\ldots, p_{\max}, \forall n = 1,\ldots, n_{\max}, \forall t = 2,\ldots, t_{\max}, \\ d^{fi}_{pn}, & \forall p = 1,\ldots, p_{\max}, \forall n = 1,\ldots, n_{\max}, t = 1, \end{cases} \tag{5.2}$$

$$d^{sr}_{pnt} := \begin{cases} d^{fi}_{pn}\dfrac{t-1}{t_{\max}-1}\left(1+\varepsilon^{sr}\right)\left(1+\eta^{sr}\cdot t\cdot R_3\right), & \forall p = 1,\ldots, p_{\max}, \forall n = 1,\ldots, n_{\max}, \forall t = 2,\ldots, t_{\max}, \\ 0, & \forall p = 1,\ldots, p_{\max}, \forall n = 1,\ldots, n_{\max}, t = 1. \end{cases} \tag{5.3}$$

The firm customer order in the first planning period is equal to the final demand, and the supply reservation is equal to zero. The quantities R_2 and R_3 are random variates of a normal distribution, i.e., $R_2 \sim N(0,\sigma_2^2)$ and $R_3 \sim N(0,\sigma_3^2)$, where σ_2^2 and σ_3^2 are the corresponding variances. Using the ratio of the planning period index to the planning horizon in expressions (5.2) and (5.3) gives a trend to the demand profile according to the demand type, i.e., firm customer orders are decreasing while supply reservations are increasing along the planning horizon. This mimics the role of a placeholder for the forecasts. The terms that are related to the demand volatility in expressions (5.2) and (5.3) include the planning period index, i.e., the further the targeted periods, the higher the demand fluctuations. An exemplary evolution of the demand over the planning horizon is showed in Figure 5.4 for a single product and a single planning occurrence, i.e., $p_{\max} = 1$ and $n_{\max} = 1$. The following parameters are used: $t_{\max} = 26$, $DL = 100$, $\varepsilon^{fo} = \varepsilon^{sr} = 0.10$, and $\eta^{fo} = \eta^{sr} = 0.05$.

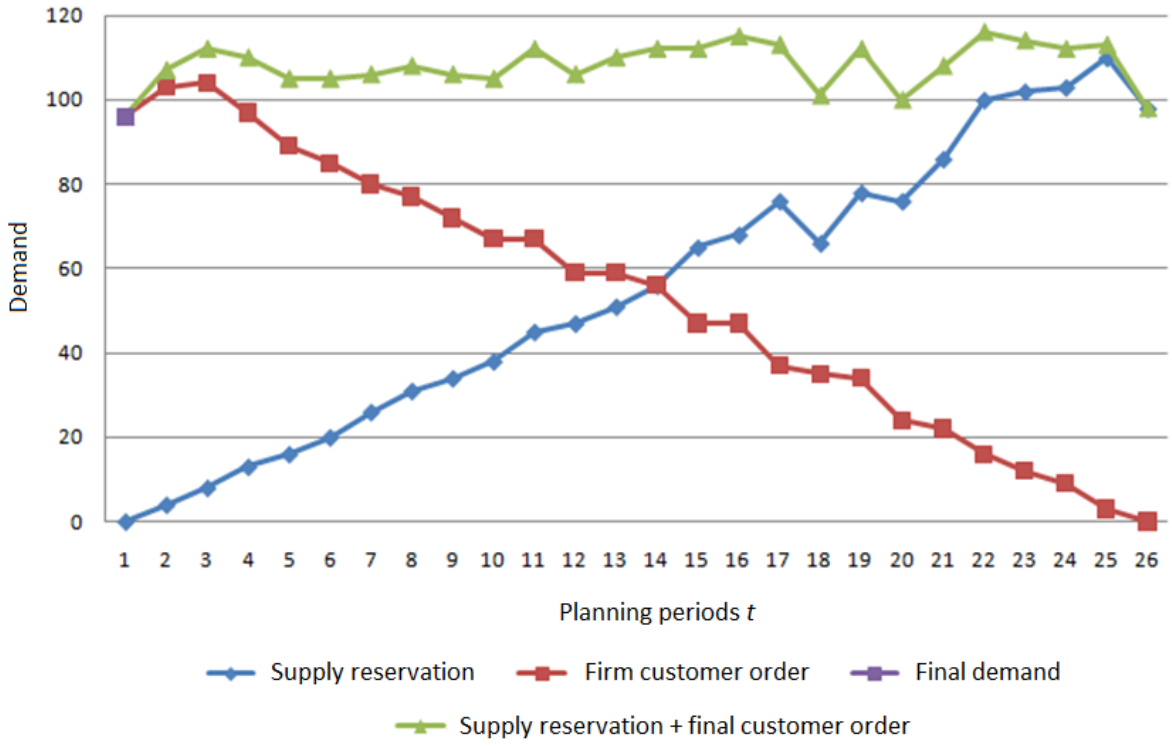

Figure 5.4: Evolution of the demand over the planning horizon for a single product and a single planning occurrence.

Algorithm 5.1 is used to generate the demand. The first loop calculates the final demands as described in formula (5.1'). The second loop computes firm customer orders and supply reservations as given by expressions (5.2) and (5.3). The scheme is carried out for all products. Algorithm 5.1 is executed prior to starting the simulation. The resulting demand quantities are rounded to the nearest integer, and they are stored in the data layer. The current demand information is provided to the planning module at each planning occurrence.

Algorithm 5.1: Scheme for the generation of the demand.

1 $p = 0, n = 0$.

2 $p := p + 1$.

3 $n := n + 1$.

4 Generate final demand d_{pn}^{fi} according to expression (5.1').

5 If $n < n_{\max}$, then go to Step 3.

6 $n = 0$, $t = 0$.

7 $n := n + 1$.

8 Retrieve final demand d_{pn}^{fi}.

9 $t := t + 1$.

10 Generate firm customer order d_{pnt}^{fo} according to expression (5.2).

11 Generate supply reservation d_{pnt}^{sr} according to expression (5.3).

12 If $t < t_{\max}$, then go to Step 9.

13 If $n < n_{\max}$, then set $t = 0$ and go to Step 7.

14 If $p < p_{\max}$, then set $n = 0$ and go to Step 2.

5.2.2.6 Demand Fulfillment Module

The demand fulfillment module converts produced quantities into sold quantities. As showed in formula (5.4) the sold quantity s'_{pt_s} results from a comparison of final demand $d_{pt_s}^{fi}$ and remaining backlog B'_{p,t_s-1} with current inventory on hand I'_{p,t_s-1} and the quantities x'_{pmt_s} of product p that left the facility m at the end of the simulation period t_s.

$$s'_{pt_s} := \min\left\{ d_{pt_s}^{fi} + B'_{p,t_s-1}, \sum_{m=1}^{m_{\max}} x'_{pmt_s} + I'_{p,t_s-1} \right\}, \forall p = 1,\ldots, p_{\max}, \forall t_s = 1,\ldots, t_{s,\max} . \qquad (5.4)$$

The final demands are provided by the demand generator, and the simulator issues information on production outcomes. When the processing of a lot is completed, the lot leaves the base system, a notification is triggered by the simulator, and its completion time is stored in the data layer. Subsequently, the current levels of inventory and backlog, respectively, are updated as follows:

$$I'_{pt_s} := I'_{p,t_s-1} + \sum_{m=1}^{m_{\max}} x'_{pmt_s} - s'_{pt_s}, \forall p = 1,..., p_{\max}, \forall t_s = 1,..., t_{s,\max}, \quad (5.5)$$

$$B'_{pt_s} := d^{fi}_{pt_s} + B'_{p,t_s-1} - s'_{pt_s}, \forall p = 1,..., p_{\max}, \forall t_s = 1,..., t_{s,\max}. \quad (5.6)$$

It is assumed that the initial parameters I'_{p0} and B'_{p0} are given. A similar approach is used in GA-MPSC to initialize the chromosomes with the difference that there is here no distinction between sold quantities related to firm customer orders and supply reservations since the demand fulfillment exclusively deals with final demands. The quantities given by expressions (5.4)-(5.6) are computed after each single period is simulated. The updated inventory and backlog levels are provided to the planning algorithm via the data layer as input parameters for the next plan. Consequently, there exists a feedback coupling between the simulation model and the planning module.

5.2.2.7 Performance Assessment Module

Due to the modeling complexity, the planning and control systems usually do not fully capture neither the behavior of the base system nor the behavior of the market. Disturbances on the shop-floor such as machine breakdowns may influence the level of WIP in the base system and thus the outcomes of the planning system. In the same way, fluctuations in the inputs of the planning algorithm such as new arrival of firm customer orders, cancelation of firm customer orders, and a lack of demand accuracy contribute to the gap between two given production plans released at two different points in time. Hence, the performance of the planning algorithm can be assessed by measuring the discrepancies between the successive production plans. This measure is known as planning stability. The lower the measure value, the more stable is the planning algorithm. After each new plan calculation, the absolute differences of the planned production outcomes compared to the plan that has been previously released are calculated. The deviations are nonlinearly weighted according to the respective planned completion date. A weight $\lambda_S \in (0,1)$ is used. With a λ_S value close to zero the discrepancies that are close to their realization dates are stronger penalized. After considering all periods, all plans, all facilities, and all products the following stability measure value is obtained:

$$S := \frac{1}{m_{\max} p_{\max} (n_{\max} - 1)} \sum_{p=1}^{p_{\max}} \sum_{m=1}^{m_{\max}} \sum_{n=2}^{n_{\max}} \sum_{t=q+1}^{t_{\max}-1} \lambda_S^{t-q} \left| x^n_{pmt} - x^{n-1}_{p,m,t+1} \right|, \quad (5.7)$$

where x^n_{pmt} are the planned quantities of product p to be completed in facility m due at the end of planning period t according to plan n. A similar measure is proposed by Sridharan *et al*. (1988). Given the rolling horizon setting, $n_{\max} > 1$ is ensured. It is also assumed that $t_{\max} > q+1$ is fulfilled. As the simulation horizon does not limit the time index in the S measure, planning periods that go beyond $t_{s,\max}$ are taken into account in the stability calculation.

Besides measuring the fluctuations between successive plans for which a simulation model is not required, the deviations between quantities planned in the planning level and quantities realized in the base level are quantified. It allows evaluating the interactions between both levels. Measure M_1 compares planned quantities $x^n_{pmt_s}$ contained in plan n with actually produced quantities x'_{pmt_s} as follows:

$$M_1 := \frac{\sum_{p=1}^{p_{\max}} \sum_{m=1}^{m_{\max}} \sum_{t_s=q+1}^{t_{s,\max}} \left| x_{pmt_s}^{t_s-q} - x'_{pmt_s} \right|}{\sum_{p=1}^{p_{\max}} \sum_{m=1}^{m_{\max}} \sum_{t_s=q+1}^{t_{s,\max}} x'_{pmt_s}}. \tag{5.8}$$

Given the rolling horizon setting, the quantities that are planned immediately before their production start are of interest. Hence the setting $n = t_s - q$ is used. It is the comparison of the quantities from the $(t_s - q)$-th plan with x'_{pmt_s}. M_1 evaluates the ability of the base level to fulfill the requirements from the planning level. A low M_1 value is desirable.

Measure M_2 is similar to M_1, except that it deals, on the one side, with planned sales quantities that come from the rolling production plans, on the other side, with sold quantities that come from the demand fulfillment module. It is defined as follows:

$$M_2 := \frac{\sum_{p=1}^{p_{\max}} \sum_{t_s=q+1}^{t_{s,\max}} \left| s_{pt_s}^{t_s-q} - s'_{pt_s} \right|}{\sum_{p=1}^{p_{\max}} \sum_{t_s=q+1}^{t_{s,\max}} s'_{pt_s}}. \tag{5.9}$$

The demand fulfillment logic does not discriminate the demand types, thus the planned sales quantities related to firm customer orders and supply reservations are summed up as follows:

$$s_{pt_s}^{t_s-q} := s_{pt_s}^{fo,t_s-q} + s_{pt_s}^{sr,t_s-q}, \forall p = 1,\ldots,p_{\max}, \forall t_s = q+1,\ldots,t_{s,\max}. \tag{5.10}$$

M_2 estimates the delivery reliability since it assesses the deviations between the ultimate commitments of the planning level towards the customers and the actually sold quantities. A low M_2 value shows the ability of the base system to stick to the sales commitments. The production strategies of the planning algorithm such as preproduction are reflected in the planned sales quantities. M_2 allows for evaluating these strategies.

Measure M_3 compares the final demands including backlog that is remaining from the previous period with the quantities sold. It evaluates the ability of the planning system and the base system to fulfill the customer requirements, i.e., a low M_3 value is desirable. It does not explicitly take any result of the planning algorithm into account. It serves as a measure of the delivery performance, also called customer service level. M_3 is defined as follows:

$$M_3 := \frac{\sum_{p=1}^{p_{\max}} \sum_{t_s=q+1}^{t_{s,\max}} \left| s'_{pt_s} - \left(d_{pt_s}^{fi} + B'_{p,t_s-1} \right) \right|}{\sum_{p=1}^{p_{\max}} \sum_{t_s=q+1}^{t_{s,\max}} \left(d_{pt_s}^{fi} + B'_{p,t_s-1} \right)}. \tag{5.11}$$

It is ensured that the denominators in expressions (5.8), (5.9), and (5.11) are strictly positive.

It is notable that other measures are proposed in the literature to assess delivery reliability and performance. They involve a binary measurement approach, i.e., an on-time delivery is rated with 1 if it occurs within a given delivery window, otherwise it is rated with zero (cf.

Gunasekaran *et al*., 2004; Bhagwat and Sharma, 2007). While this approach is suitable for evaluating an order management system, the usage of more expressive measures as showed in expressions (5.8), (5.9), and (5.11) is preferred in the context of MP decisions that deal with aggregated quantities.

As showed in Subsection 5.2.2.6, the sold quantities as well as the current levels of inventory and backlog are computed based on the realized production quantities. By considering the revenues, inventory holding costs, costs for unmet orders, variable manufacturing costs, and fixed location costs, one can compute the realized objective function value over the entire simulation horizon as follows:

$$f' := \sum_{p=1}^{p_{\max}} \sum_{t_s=q+1}^{t_{s,\max}} \left(rev_{pt_s} s'_{pt_s} - hc_{pt_s} I'_{pt_s} - udc_{pt_s} B'_{pt_s} - \sum_{m=1}^{m_{\max}} mc_{pmt_s} x'_{pmt_s} - \sum_{m=1}^{m_{\max}} lc_{pmt_s} sign\left(x'_{pmt_s}\right) \right). \quad (5.12)$$

The expression f' is derived from the objective function f in the MIP formulation of MPSC with the difference that the time index in f' refers to simulation periods, and f' considers realized values.

Using a simulator allows monitoring start and completion dates of the lots. The control algorithm determines the production start of a lot based on its planned completion date as provided by the planning algorithm and by considering the lead time. Whenever the cycle time deviates from the planned lead time, the tardiness and earliness of the lot can be measured. This measure is used as a performance indicator of the base level to meet the requests from the planning algorithm. The earliness and tardiness measure L of the set of completed lots J is computed as follows:

$$L := \sum_{j \in J} \left| c_j - d_j \right|, \quad (5.13)$$

where c_j is the completion time and d_j is the planned completion time or due date of lot j.

The performance measures S, M_1, M_2, M_3, f', and L are implemented in the performance assessment module.

5.2.3 Application of the Framework

The application of the framework is described by Algorithm 5.2. The repeat loop allows simulating the base level over the entire simulation horizon. The simulator is stopped at each planning occurrence. Flow statistics such as the cycle times of the completed lots are stored in the data layer. The demand fulfillment module is executed, and it calculates the respective sold quantities. The inventory and backlog levels are updated. The data layer then provides the demands for the planning horizon to the production planning module along with the current state of the base system, e.g., WIP. Next, the planning algorithm is performed. The resulting production requests with due quantities and planned completion dates are transferred to the control module. The production control algorithm assigns release dates to the lots. Finally, the simulation proceeds. Whenever a release date of a lot is reached during the simulation, an event-driven notification is triggered by the simulator, a new lot object is created in the data layer, and its status is updated in relation with its progress. Finally, the performance assessment module is executed.

Algorithm 5.2: Scheme for the simulation-based performance assessment of MP approach.

```
1  Repeat until the end of the simulation horizon is reached.
2      Run the simulation.
3      Break the simulation when the next planning occurrence is reached.
4          Calculate flow statistics and sold quantities of the last simulated periods.
5          Provide current demand, WIP, inventory, and backlog to the planning module.
6          Run the MP algorithm.
7          Provide planned completion dates and due quantities to the control module.
8          Run the production control algorithm.
9          Provide the lot release dates to the simulator.
10     End break.
11 End repeat.
12 Calculate and return the performance measure values.
```

5.2.4 Implementation of the Framework

Given the level of detail that is required in the MP algorithm, the planning module deals with an aggregated representation of the base level. The simulation is carried out on a detailed level. Therefore, a data structure is used for aggregation and disaggregation purposes within the framework. In Figure 5.5, a Unified Modeling Language (UML) class diagram shows the data model that is implemented in the data layer as adopted and modified from Mönch (2008).

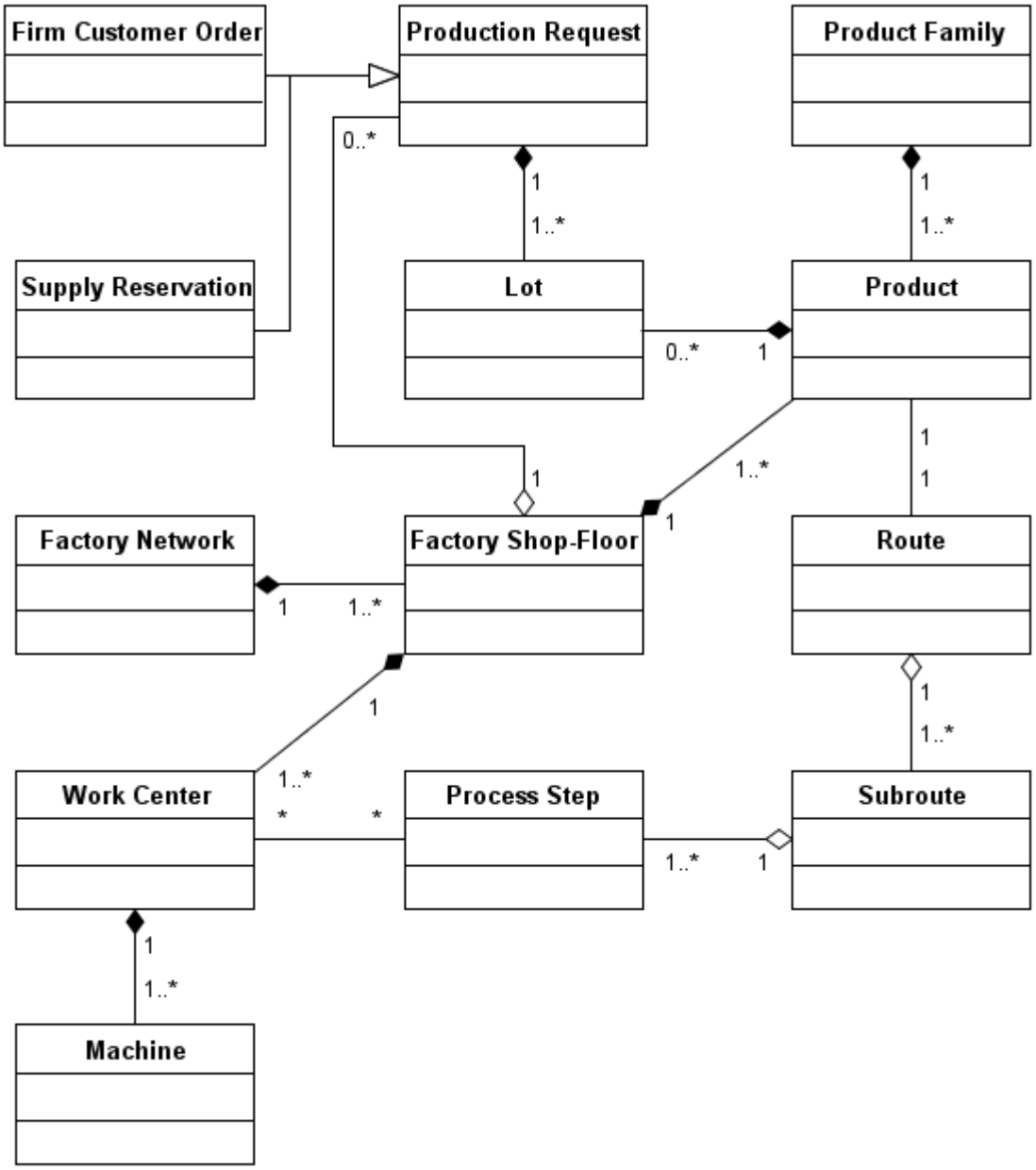

Figure 5.5: UML class diagram of the data model implemented in the framework.

Both demand classes “Firm Customer Order” and “Supply Reservation” are derived from the class “Production Request” that is converted into lots in the production control module. The product aggregation is made possible by the classes “Product” and “Product Family”. A product family encompasses products with similar features that pass the same sequence of process steps. The process flows of the products are given by the classes “Route”, “Subroute”, and “Process Step”. Within a given factory, a product is assigned to at most one route. The base system is represented by the classes “Factory Network”, “Factory Shop-Floor”, “Work Center”, and “Machine”. The class “Factory Network” models a one-layer manufacturing network with parallel wafer fabs. The sites are single entities with no flow of material across them (cf. Subsection 3.1.1). The factory shop-floor is composed of work centers. A single work center consists of identical parallel machines. In the case of re-entrant process flows, a work center is used several times at different process steps of a given route. A work center can be

labeled as a bottleneck resource. The attributes of the classes related to the equipment allow for representing the capacity of the base system.

The simulation model is implemented using the discrete-event simulator AutoSched AP, a class library that offers customization functionalities using the C++ programming language. The blackboard-like data layer and the other modules of the framework are also coded in C++.

5.3 Application of the Framework to RA-MPSC and GA-MPSC

In this section, the proposed framework is applied to evaluate the performance of RA-MPSC and GA-MPSC, i.e., $\gamma_{\max} = 2$. The PD-MPSC scheme is not used in the framework for practical reasons, mainly due to the large computing time that is required to solve a single problem instance of MPSC (cf. Table 4.6), that means, the application of PD-MPSC in a rolling horizon setting would be associated with an even larger computational burden.

First, details of the parameter settings of the planning and base level are presented. Then, the representation of the base level in the simulation model is described. The scheme used to generate the simulation scenarios is introduced. Research hypotheses are formulated, and the results of computational experiments are discussed.

5.3.1 Parameter Settings of the Planning and Base Level

The number of products is set to $p_{\max} = 32$. The base system consists of $m_{\max} = 4$ non-identical wafer fabs in parallel. The wafer fabs are supposed to be in-house locations since the base level is modeled with information that is usually not known for silicon foundries, e.g., number of available machines. Because of the expensive machines and the resulting re-entrant process flows, the photolithography work center is assumed to be the leading bottleneck resource in every facility. Thus, $b_{m,\max} = 1, \forall m = 1,..., m_{\max}$ holds. To achieve a heterogeneous base system, the capacity settings are varied in the wafer fabs of the network. The capacity with respect to facility m depends on the number of parallel machines at the leading bottleneck work center denoted by N_m, the machine breakdowns, and the PM jobs that are executed in the third and fourth wafer fabs. Table 5.1 summarizes the capacity settings of the base system.

Table 5.1: Capacity settings of the base system.

Wafer Fabs m	N_m	Breakdowns		PM	
		MTTF	MTTR	TTPM	PMT
Wafer Fab 1	4	2686	157	0	0
Wafer Fab 2	3	2686	157	0	0
Wafer Fab 3	3	2686	157	103	9
Wafer Fab 4	3	2686	157	824	72

Remark: MTTF, MTTR, TTPM, and PMT are given in minutes.

MTTR and MTTF are the mean time to repair and the mean time to failure in minutes, respectively, and PMT and TTPM are the time to complete the PM and the time to the next PM in minutes, respectively. The breakdowns follow an exponential distribution. The breakdown and PM settings are expected to decrease the operations time of the machines

(cf. SEMI, 2004) on average by 5.85% and 8.74%, respectively. It is ensured that photolithography remains the leading bottleneck work center despite the different capacity settings. The planned capacity, which corresponds to the estimated manufacturing time in machine-hours available in a planning period (cf. SEMI, 2004), is set as follows:

$$C_{mt}^{\max} = N_m T\left(1-\frac{MTTR_m}{MTTF_m}-\frac{PMT_m}{TTPM_m}\right), \forall m = 1,..,m_{\max}, \forall t = 1,...,t_{\max}, \qquad (5.14)$$

where T is the operations time in hours of the wafer fab in one planning period. Since wafer fabs typically operate twenty-four hours a day, it results in $T = 24\tau_T$ hours with τ_T being the length of a planning period. The engineering time is assumed equal to zero in all periods. The capacity that results from expression (5.14) is constant over the planning horizon.

The typical length of a time bucket in a master plan is one week. Here, the length of a planning period is reduced to two days, i.e., $\tau_T = 2$ days, to mimic that the average cycle time is around six periods and to decrease the computational effort. This setting corresponds to situations where weekly time buckets are considered in mid-term production planning approaches and the cycle times in wafer fabrication span from one to two months. A similar technique is used by Irdem *et al.* (2010). The time unit in the base level is one day. The following settings are used: $t_{\max} = 26$ periods, i.e., 52 days, $t_{s,\max} = 200$ days, and $\Delta t = 1$ period, i.e., 2 days. Consequently, $n_{\max} = 100$ planning occurrences are considered.

The value of the product lead time q is derived with respect to the demand level under consideration from the average cycle time across all products as obtained from preliminary simulation experiments. It is rounded up to the next integer number of periods. The lead time values are detailed in Subsection 5.3.3. The products have unequal raw processing times. However, it is ensured that the differences between the average cycle times of the different products are so small that they correspond to the same number of planning periods. Hence, the assumption of having a unique lead time for all products across all wafer fabs is reasonable. The capacity consumption factors, i.e., the entries of the matrices $\tilde{C}$ that are used in the planning system for limiting the loading (cf. Subsection 3.1.2), can be computed based on the given lead time and the processing times of the process steps on bottleneck machines. The settings for inventory holding costs, costs for unmet firm orders, variable manufacturing costs, fixed location costs, and revenues are identical to the settings used in Chapter 4. They are scaled-down by a factor of 3.5 to take into account the reduced length of a single planning period. The parameter settings of GA-MPSC and RA-MPSC are the same as in Chapter 4.

The standard deviation of the normally distributed random variate that is used in formula (5.1') to generate the final demands is defined as $3\sigma_1 = 0.1$. Given the distribution function of a standard normal distribution, most of the final demands do not deviate by more than 10% from the given demand level. It is also assumed that $3\sigma_2 = 3\sigma_3 = 0.1$. The quantity λ_S in expression (5.7) is set to $\lambda_S = 0.5$. Thus, the weight λ_S^{t-q} is twice as high as with a longer time lag of $t-q+1$ periods. Time lags greater than six periods lead to a weight less than 0.01.

5.3.2 Detailed and Reduced Simulation Models

A reduced simulation model that focuses on the major characteristics of the base system is used in the framework. It allows decreasing the computational burden that is typical for discrete-event simulation. The reduced simulation model is derived from a detailed model. First, the initial detailed model is described. Then, the reduction approach is explained.

The detailed simulation model is a scaled-down variant of the MIMAC-I data set (cf. MASMLab, 1997). It contains two products and two respective manufacturing routes from the MIMAC-I data set. Each route is divided into five subroutes where each subroute corresponds to the production of a certain set of layers. In the detailed model, an individual process flow is assigned to each of the $p_{\max} = 32$ products as one of the 2^5 combinations of the subroutes of both products. The processing of the products requires up to 66 process steps. The process flow definition is identical in all facilities. Consequently, each product has the same raw process time across all wafer fabs. The standard lot size is set to $ls = 48$ wafers in all facilities. Setups and operators are not included in the model.

The approach proposed in Hung and Leachman (1999) is applied to downsize the level of detail while achieving lot cycle time distributions that are comparable to those obtained with the detailed simulation model. A similar reduction method is discussed in Völker and Gmilkowsky (2003). Extensive computational experiments were performed to ensure the accuracy of the reduction. The reduction method leads to Algorithm 5.3 that is showed on the next page.

Given the different capacity settings of the wafer fabs, the parts of the simulation model related to the different facilities are reduced separately, and they are reassembled in the last step of Algorithm 5.3. The waiting times obtained at Step 7 of Algorithm 5.3 depend on the loading of the wafer fab and primarily on the demand level that leads to the master plan. Hence, the reduction of the simulation model is carried out with respect to the demand levels defined in Subsection 5.3.3 divided by the number of facilities $m_{\max}$. Let Ω_m be the set of all process steps in facility m. The subset $\Omega_m^r \subseteq \Omega_m$ refers to operations on machines of non-bottleneck work centers. The reduction is based on the observation that the average sum of waiting and processing times on a machine with low utilization provides an accurate estimate of the time required to complete the process step. Ten independent replications are performed, i.e., $\theta_{red} = 10$. Algorithm 5.3 assumes that the bottleneck resources are identified prior to the reduction, i.e., in the problem setting photolithography is considered as the only bottleneck work center in the facilities. The computational experiments carried out with the detailed simulation model Ψ_m^d confirm that the photolithography work center is characterized by the highest resource utilization as well as long and variable waiting times. Using time delays instead of explicitly modeling the processing on the non-bottleneck machines allows for a reduction of the average computing time by around 40% given the parameters described in Section 5.3.

Hung and Leachman (1999) discuss the usage of deterministic versus stochastic delays. Extensive computational experiments were performed to compare both approaches. As a result, a gamma distribution is chosen for modeling the delays with $\hat{\alpha} = \bar{\mu}^2 / \bar{\sigma}^2$ and $\hat{\beta} = \bar{\mu} / \bar{\sigma}^2$ being the estimators for the shape and rate parameters, respectively, where $\bar{\mu}$ is the sample mean and $\bar{\sigma}^2$ is the empirical variance of the θ_{red} sums of waiting and processing times for a given process step.

Algorithm 5.3: Scheme for the reduction of the simulation model.

1 Repeat until all facilities $m = 1,..., m_{\max}$ have been considered.

2 Generate a feasible master plan mp_m for a single facility m with respect to its capacity and a given demand level.

3 Convert mp_m into a lot release schedule rs_m with respect to a given lead time.

4 Build a set Ω_m^r of process steps on non-bottleneck work centers as implemented in a detailed simulation model Ψ_m^d of facility m.

5 For $i := 1$ to θ_{red}.

6 Simulate rs_m using Ψ_m^d.

7 Save the sum of waiting and processing times of each process step in Ω_m^r.

8 End For.

9 Compute the average sum over the θ_{red} iterations of the waiting and processing times, called delay, of each process step in Ω_m^r.

10 Replace in Ψ_m^d the raw process time of each process step in Ω_m^r by the respective delay.

11 Remove the non-bottleneck work centers from Ψ_m^d.

12 $\Psi_m^r \leftarrow \Psi_m^d$.

13 Return a reduced simulation model Ψ_m^r of facility m.

14 End Repeat.

15 Assemble the m reduced models into a single reduced simulation model Ψ^r.

5.3.3 Design of Experiments

A factorial design with eight varying factors is used. Seven factors are related to the planning system, i.e., planning algorithm (PA), variability and bias of supply reservations and firm customer orders (SRV, SRB, FOV, and FOB), and bias of planned capacity limits and lead times (CLB and LTB). Instances of the base system are given by the demand level (DL). It is assumed that the parameters in relation with the capacity of the base system (SC) are fixed, i.e., number of machines at the bottleneck work center, breakdown and PM settings. DL, SRV, SRB, FOV, and FOB are factors of the environmental uncertainty. CLB and LTB refer to an inappropriate

representation of the base system in the planning level. Given the number of levels for each factor, 1296 factor combinations are obtained. To limit the computational effort, only one scenario per factor combination is considered. Five independent simulation replications are performed for each combination. This allows for mitigating the effect of disruptions of the base system on the results. Thus, a total number of 6480 simulation runs is carried out. The design of experiments is summarized in Table 5.2.

Table 5.2: Design of experiments (II).

Factor	Notation	Level	Number
Planning System			
Planning Algorithm	PA	GA-MPSC, RA-MPSC	2
Supply Reservation Variability	SRV	low, high	2
Supply Reservation Bias	SRB	overestimated, underestimated, accurate	3
Firm Cust. Order Variability	FOV	low, high	2
Firm Cust. Order Bias	FOB	overestimated, underestimated, accurate	3
Capacity Limit Bias	CLB	overestimated, underestimated, accurate	3
Lead Time Bias	LTB	overestimated, underestimated, accurate	3
Base System			
Demand Level	DL	low, high	2
System Capacity	SC	fixed N_m, breakdown and PM settings	1
Total factor combinations			1296
Number of problem scenarios per combination			1
Number of simulation replications per combination			5
Total number of simulation runs			6480

The variability factors are either low, i.e., $\eta^{sr} = 0.01$, $\eta^{fo} = 0.01$, or high, i.e., $\eta^{sr} = 0.05$, $\eta^{fo} = 0.05$. The bias factors related to the demand are either overestimated, i.e., $\varepsilon^{sr} = 0.10$, $\varepsilon^{fo} = 0.10$, underestimated, i.e., $\varepsilon^{sr} = -0.10$, $\varepsilon^{fo} = -0.10$, or accurate, i.e., $\varepsilon^{sr} = 0.00$, $\varepsilon^{fo} = 0.00$. The high and low demand levels correspond to an average flow factor across all facilities of 2.95 and 2.45, respectively. The resulting utilization of the machines of the bottleneck work center is 92% and 84%, respectively. It leads to an average product cycle time of 11.5 days and 7.7 days, respectively. The length of a planning period being two days, the accurate setting of LTB is set to six periods and four periods with respect to the demand level under consideration. The accurate setting is increased and decreased by one period to obtain the overestimated and underestimated lead time, respectively. The accurate setting of CLB results from expression (5.14). The overestimated and underestimated values of CLB are computed by increasing and decreasing the accurate setting by 15%, respectively.

The computational experiments are carried out on a computer equipped with a 2.5 GHz dual processor and 2.0 GB memory. The average computing time for one simulation run with RA-MPSC and GA-MPSC as planning algorithm is around four minutes and seventeen minutes, respectively. The computational effort of GA-MPSC is smaller than stated in Chapter 4, mainly due to the smaller number of products. As the base system does not contain any lots in processing at the beginning of a simulation run, the simulation outputs of the first forty simulated days are not considered for the result analysis to exclude the warm-up phase of the simulation. The plan, simulation time, and lot indices in expressions (7)-(13) are initialized accordingly.

The following five hypotheses are investigated when applying the simulation-based framework to RA-MPSC and GA-MPSC.

- **Hypothesis 1.** An inaccurate representation of the base system in the planning system results in unmet production commitment, lower delivery reliability, lower delivery performance, and reduced earnings.
- **Hypothesis 2**. A high demand variability results in reduced planning stability, lower delivery reliability, lower delivery performance, and reduced earnings.
- **Hypothesis 3.** A demand bias results in lower delivery reliability, lower delivery performance, and reduced earnings.
- **Hypothesis 4.** The planning algorithm has an impact on the planning stability, the delivery reliability, the delivery performance, and the earnings.
- **Hypothesis 5.** A high demand level is an aggravating factor for Hypotheses 1-4.

5.3.4 Results of Computational Experiments

The results of computational experiments were analyzed using the ANOVA procedure. Normal probability plots of residuals and plots of residuals versus fitted values were used to check the model adequacy as suggested in Montgomery (2008). Departures from the normal distribution assumption and the assumption of constant variance are moderate.

Table 5.3 shows the main and two-way interaction effects of the independent variables PA, SRV, SRB, FOV, FOB, CLB, LTB, and DL for each of the dependent variables S, M_1, M_2, M_3, f', and L at a 5% significance level. The column DF gives the number of degrees of freedom. For each dependent variable, the F value is indicated as well as the probability that the factor has no effect, i.e., Pr>F.

A closer examination of the main interaction effects shows that PA significantly influences S, M_2, M_3, and f'. In addition, SRV and FOV influence S, M_2, M_3, and f'. SRB and FOB influence M_2, M_3, and f'. Moreover, CLB and LTB influence M_1, M_2, M_3, f', and L. Finally, DL influences each of the dependent variables. By comparing the F values of SRV versus FOV and SRB versus FOB for S, M_2, M_3, f' and M_2, M_3, f', respectively, it is notable that the alteration of the firm customer orders has a higher impact on the performance measure values than those of the supply reservations. This is explained by the higher priority given to orders in the planning heuristic and the respective cost settings.

An instance factor is included as a nested effect within the problem factors to account for blocking on problem instances. This nested factor cannot interact with the problem characteristics, because it is not comparable across different problem levels. It is assumed that it does not interact with both heuristics in order to have an estimate of the random error (cf. Rardin and Uzsoy, 2001). The instance factor Inst(SRV*SRB*FOV*FOB*CLB*LTB*DL) is abbreviated as Inst(.) in Table 5.3.

Table 5.3: Main and two-way interaction effects as provided by the ANOVA procedure (II).

		S		M_1		M_2		M_3		f'		L	
Source	**DF**	**F**	**Pr>F**	**F**	**Pr>F**	**F**	**Pr>F**	**F**	**Pr>F**	**F**	**Pr>F**	**F**	**Pr>F**
PA	1	935.60	0.0001	0.01	0.9359	1439.20	0.0001	1021.21	0.0001	1667.39	0.0001	0.02	0.8877
SRV	1	1263.25	0.0001	17.07	0.0001	292.47	0.0001	356.91	0.0001	923.44	0.0001	183.61	0.0001
SRB	2	11.95	0.0001	0.80	0.4499	178.39	0.0001	156.37	0.0001	95.34	0.0001	10.50	0.0001
FOV	1	2243.23	0.0001	16.82	0.0001	534.41	0.0001	545.20	0.0001	1496.75	0.0001	185.50	0.0001
FOB	2	67.55	0.0001	4.74	0.0088	364.24	0.0001	348.32	0.0001	324.97	0.0001	53.83	0.0001
CLB	2	7.06	0.0009	1203.58	0.0001	322.57	0.0001	583.12	0.0001	297.77	0.0001	840.35	0.0001
LTB	2	18.09	0.0001	1630.85	0.0001	444.62	0.0001	602.81	0.0001	235.46	0.0001	1196.80	0.0001
DL	1	4029.20	0.0001	3700.93	0.0001	3029.39	0.0001	2368.14	0.0001	1215.85	0.0001	4588.32	0.0001
Inst(.)	64	1.62	0.1670	0.13	0.9734	0.43	0.7895	0.34	0.8540	0.30	0.8787	0.74	0.5614
PA*SRV	1	4.38	0.0364	0.03	0.8549	0.71	0.4002	0.45	0.5030	2.97	0.0849	0.01	0.9831
PA*SRB	2	0.04	0.9576	0.01	0.9876	0.01	0.9891	0.05	0.9556	0.41	0.6643	0.01	0.9999
PA*FOV	1	5.41	0.0201	0.01	0.9138	0.82	0.3641	0.85	0.3570	4.68	0.0305	0.01	0.9520
PA*FOB	2	0.17	0.8451	0.02	0.9830	0.31	0.7350	0.33	0.7205	1.04	0.3544	0.01	0.9966
PA*CLB	2	0.01	0.9914	0.02	0.9795	0.05	0.9466	0.02	0.9806	1.11	0.3290	0.01	0.9974
PA*LTB	2	0.03	0.9738	0.01	0.9961	0.39	0.6746	0.18	0.8318	0.91	0.4045	0.01	0.9956
PA*DL	1	11.79	0.0006	0.09	0.7613	1.50	0.2209	1.30	0.2550	4.34	0.0371	0.01	0.9512
SRV*SRB	2	0.04	0.9581	0.02	0.9796	0.05	0.9528	0.18	0.8368	0.19	0.8284	0.01	0.9906
SRV*FOV	1	5.79	0.0162	0.01	0.9931	0.86	0.3536	0.71	0.4011	1.08	0.2978	0.01	0.9664
SRV*FOB	2	0.15	0.8586	0.01	0.9967	0.23	0.7931	0.48	0.6180	0.47	0.6224	0.05	0.9532
SRV*CLB	2	0.10	0.9052	0.96	0.3827	0.36	0.6991	0.31	0.7317	0.37	0.6916	0.66	0.5187
SRV*LTB	2	0.12	0.8902	1.06	0.3463	0.49	0.6100	0.53	0.5893	0.29	0.7499	0.97	0.3787
SRV*DL	1	17.00	0.0001	2.38	0.1233	3.49	0.0617	2.63	0.1047	5.89	0.0153	2.81	0.0939
SRB*FOV	2	0.08	0.9263	0.01	0.9867	0.16	0.8495	0.32	0.7284	0.14	0.8674	0.01	0.9966
SRB*FOB	4	23.08	0.0001	0.94	0.4411	45.11	0.0001	4.52	0.0012	23.02	0.0001	9.55	0.0001
SRB*CLB	4	0.01	1.0000	0.02	0.9994	0.03	0.9980	0.01	0.9997	0.09	0.9851	0.02	0.9991
SRB*LTB	4	0.01	1.0000	0.04	0.9475	0.03	0.9980	0.04	0.9964	0.07	0.9899	0.10	0.9812
SRB*DL	2	0.09	0.9105	0.29	0.7449	0.15	0.8566	0.43	0.6481	0.30	0.7444	0.31	0.7303
FOV*FOB	2	0.43	0.6530	0.02	0.9847	0.40	0.6680	0.62	0.5394	0.68	0.5068	0.03	0.9719
FOV*CLB	2	0.08	0.9266	0.87	0.4199	0.37	0.6898	0.52	0.5941	1.19	0.3041	0.63	0.5342
FOV*LTB	2	0.10	0.9030	1.05	0.3497	0.82	0.4393	0.63	0.5314	0.51	0.6032	0.82	0.4387
FOV*DL	1	31.00	0.0001	1.96	0.1617	4.00	0.0455	3.17	0.0750	9.37	0.0022	3.36	0.0670
FOB*CLB	4	0.02	0.9991	0.20	0.9379	0.08	0.9898	0.18	0.9497	0.17	0.9517	0.15	0.9624
FOB*LTB	4	0.01	0.9997	0.36	0.8404	0.12	0.9743	0.29	0.8866	0.19	0.9463	0.24	0.9181
FOB*DL	2	0.77	0.4617	0.79	0.4526	0.68	0.5065	1.62	0.1982	1.59	0.2044	0.58	0.5578
CLB*LTB	4	19.71	0.0001	124.81	0.0001	135.09	0.0001	197.52	0.0001	203.81	0.0001	130.03	0.0001
CLB*DL	2	0.03	0.9666	1.19	0.3033	0.99	0.3712	0.67	0.5111	0.63	0.5318	15.74	0.0001
LTB*DL	2	0.01	0.9929	4.30	0.0136	1.08	0.3397	1.42	0.2410	1.22	0.2966	27.32	0.0001

The two-way interaction effects PA*DL, SRV*DL, and FOV*DL indicate that DL is an aggravating factor, as stated in Hypothesis 5, with respect to S and f'. The factors CLB*DL and LTB*DL also show significant interactions for L values. Nevertheless, the interaction effects of SRB*DL and FOB*DL do not appear to be significant. Significant interactions with respect to S, M_2, M_3, and f' are indicated by the factor SRB*FOB. They can be explained when planning decisions made in the first place based on inaccurate forecasted demands are aggravated later by erroneous firm customer orders. The factor CLB*LTB also indicates significant interactions between biased capacity limits and biased lead times for all dependent variables.

In order to further check the hypotheses stated in Subsection 5.3.3, the results of computational experiments are presented with respect to the performance measures used. The results are grouped according to selected factors from the design of experiments in Tables 5.4-5.7. The rank values in Tables 5.4-5.7 are obtained using Duncan's multiple range test at a 5% significance level. The rank values are always presented at the right of the performance measure under consideration. The objective function values f' are given in millions Euros in Tables 5.4-5.7.

Table 5.4 shows the impact of inaccurate capacity and product lead time settings on M_1, M_2, M_3, f', and L. The best performance, i.e., the lowest M_1, M_3, and L values and highest f' value, is obtained with respect to the demand level when both parameters CLB and LTB are set accurately. One observes that underestimated capacities and overestimated lead times work towards early deliveries, which may result in higher fulfillment of sales commitments, i.e., lower M_2 values, than with accurate settings. The worst performance is obtained when capacities are overestimated and lead times are underestimated since the production requests provided by the planning system are cut down. Because of the lower resource utilization, the lots arrive earlier than expected in this situation. This leads to higher M_1 and L values. The delivery performance degrades, i.e., higher M_3 values are obtained, since released quantities are not high enough to cope with final demands. The increased costs due to unmet firm customer orders lead to lower f' values. In addition, one can see that M_1 and L values are correlated. Hence, Hypothesis 1 is supported by the results presented in Table 5.4.

Table 5.4: Impact of inaccurate representation of the base system in the planning system.

DL	CLB	LTB	M_1	Rank	M_2	Rank	M_3	Rank	f'	Rank	L	Rank
Low	Accurate	Accurate	0.5758	1	0.7521	1~2	0.8813	1	5.29	1	2569	1
		Underest.	0.6806	3	0.8132	3	0.9832	2~3	5.05	2~3	3030	3
		Overest.	0.6665	2	0.7520	1~2	0.9885	2~3	5.00	2~3	2966	2
	Underest.	Accurate	0.6571	1	0.7521	1~2	0.9903	1	4.89	1~2	2921	1
		Underest	0.7646	3	0.8151	3	1.0983	3	4.58	3	3518	3
		Overest.	0.7376	2	0.7504	1~2	1.0846	2	4.85	1~2	3205	2
	Overest.	Accurate	0.6720	1	0.8074	1~2	0.9792	1	5.00	1	2992	1
		Underest.	0.7880	3	0.8767	3	1.0996	3	4.53	3	3708	3
		Overest.	0.7582	2	0.8042	1~2	1.0797	2	4.82	2	3389	2
High	Accurate	Accurate	0.6710	1	0.8555	1~2	1.0065	1	5.78	1	3113	1
		Underest.	0.7805	3	0.9188	3	1.1133	2~3	5.52	2~3	3675	3
		Overest.	0.7658	2	0.8563	1~2	1.1194	2~3	5.46	2~3	3599	2
	Underest.	Accurate	0.7553	1	0.8548	1~2	1.1219	1	5.34	1~2	3541	1
		Underest.	0.8714	3	0.9237	3	1.2374	3	4.99	3	4266	3
		Overest.	0.8363	2	0.8469	1~2	1.2144	2	5.33	1~2	3888	2
	Overest.	Accurate	0.7712	1	0.9136	1~2	1.1078	1	5.46	1	3627	1
		Underest.	0.8963	3	0.9915	3	1.2386	3	4.93	3	4495	3
		Overest.	0.8592	2	0.9090	1~2	1.2143	2	5.27	2	4106	2

Table 5.5 shows the impact of supply reservation and firm customer order variability on S, M_2, M_3, and f'. The best performance with respect to the demand level is achieved when both demand variability settings are set to low. One can observe that increasing demand variability reduces the planning stability, i.e., higher S values, decreases the delivery reliability, i.e., higher M_2 values, decreases the delivery performance, i.e., higher M_3 values, and reduces the objective function values, i.e., lower f' values. Thus, these results support Hypothesis 2.

Table 5.5: Impact of demand variability.

DL	SRV	FOV	S	Rank	M_2	Rank	M_3	Rank	f'	Rank
Low	Low	Low	2.0851	1	0.7509	1	0.9604	1	5.31	1
		High	2.3942	2	0.7968	2	1.0248	2	4.84	2
	High	Low	2.3276	1	0.7842	1	1.0122	1	4.93	1
		High	2.6629	2	0.8341	2	1.0815	2	4.48	2
High	Low	Low	2.4637	1	0.8491	1	1.0842	1	5.82	1
		High	2.8247	2	0.9021	2	1.1576	2	5.29	2
	High	Low	2.7485	1	0.8892	1	1.1444	1	5.38	1
		High	3.1464	2	0.9463	2	1.2229	2	4.87	2

Table 5.6 shows the impact of biased demand. The highest delivery reliability (M_2) and highest objective function value (f') are obtained when both demand parameters SRB and FOB are set accurately. The M_2 values are affected both by underestimated and overestimated demand, whereas the M_3 values are mainly impacted when demand is underestimated since the released quantities do not allow for fulfilling final demands. On the contrary, the overestimated demand causes additional inventory that strives for increased delivery performance which may lead to lower M_3 values compared to the situation when SRB and FOB are set accurately. The objective function values f' are affected both by inventory holding costs and costs due to unmet orders. It can be seen that biased firm customer orders have a stronger impact on the performance than biased supply reservations mainly due to priority and cost settings. The results presented in Table 5.6 confirm Hypothesis 3.

Table 5.6: Impact of demand bias.

DL	SRB	FOB	M_2	Rank	M_3	Rank	f'	Rank
Low	Accurate	Accurate	0.7218	1	0.9765	1~2	5.18	1
		Underest.	0.7767	2	1.0500	3	4.90	2~3
		Overest.	0.7889	3	0.9785	1~2	4.91	2~3
	Underest.	Accurate	0.7589	1	1.0299	1~2	5.00	1
		Underest.	0.8218	2~3	1.1150	3	4.49	3
		Overest.	0.8235	2~3	1.0259	1~2	4.83	2
	Overest.	Accurate	0.7712	1	0.9777	1~2	5.04	1~2
		Underest.	0.8288	2~3	1.0540	3	4.66	3
		Overest.	0.8317	2~3	0.9701	1~2	5.01	1~2
High	Accurate	Accurate	0.8230	1	1.1070	1~2	5.66	1
		Underest.	0.8813	2	1.1853	3	5.35	2~3
		Overest.	0.8947	3	1.1076	1~2	5.35	2~3
	Underest.	Accurate	0.8626	1	1.1625	1~2	5.46	1
		Underest.	0.9320	2~3	1.2595	3	4.87	3
		Overest.	0.9287	2~3	1.1565	1~2	5.29	2
	Overest.	Accurate	0.8760	1	1.1072	1~2	5.50	1~2
		Underest.	0.9382	2~3	1.1906	3	5.08	3
		Overest.	0.9338	2~3	1.0944	1~2	5.50	1~2

Table 5.7 shows the impact of the planning algorithm used. On the one hand, GA-MPSC achieves higher objective function values than RA-MPSC, which comes along with a higher delivery performance, i.e., lower M_3 values. The GA-MPSC scheme allows for better coping with the planned sales quantities. It enables opportunistic planning strategies such as pre-producing lots that are kept on stock for a later fulfillment of customer demands (cf. Section 4.3). This leads to a higher correlation between planned and realized sales quantities, i.e., lower M_2 values. On the other hand, the planning stability is smaller when applying GA-MPSC. The nervousness can be explained by the optimization opportunities in GA-MPSC, e.g., dealing with the sales quantities and the local search procedure. While RA-MPSC uses a given set of rules, GA-MPSC is a stochastic scheme, i.e., random variables are used in the initialization scheme, in the genetic operators, and in the local search scheme, that may lead to different production and sales quantities in the re-planned periods compared to the previous planning occurrence. It can be concluded from the results in Table 5.7 that Hypothesis 4 is confirmed.

Table 5.7: Impact of the planning algorithm used.

DL	PA	S	Rank	M_2	Rank	M_3	Rank	f'	Rank
Low	RA-MPSC	2.2549	1	0.8302	2	1.0656	2	4.65	2
	GA-MPSC	2.4800	2	0.7528	1	0.9739	1	5.14	1
High	RA-MPSC	2.6626	1	0.9374	2	1.2010	2	5.07	2
	GA-MPSC	2.9290	2	0.8560	1	1.1036	1	5.61	1

Finally, considering the performance measure values in Tables 5.4-5.7, it is notable that the tendencies are enforced when the demand level is high. Consequently, Hypothesis 5 is supported by this observation.

5.4 Conclusion

This chapter introduced a simulation-based framework to evaluate the performance of MP approaches. An application of the framework was discussed by investigating the performance of RA-MPSC and GA-MPSC in a rolling horizon setting. The results of computational experiments emphasized the conflict that exists when striving for large objective function values considering single problem instances and the impact on the performance measure values achieved over the simulation horizon. It confirms the necessity to evaluate planning algorithms in a rolling horizon setting while confronted with demand and system uncertainties. Parts of the results discussed in this chapter are presented in Ponsignon and Mönch (2012b).

6. Using Iterative Simulation to Deal with Load-Dependent Lead Times in MPSC

Most of the existing production planning models assume a fixed lead time as an exogenous, prescribed parameter of the planning approach (cf. Voß and Woodruff, 2006). It is known from queueing theory that the cycle time increases nonlinearly with the utilization of the resources of the base system (cf. Hopp and Spearman, 1996). However, the utilization is a result of the release schedule used. This leads to circularity in production planning. On the one hand, the planning approach determines the release schedule based on a prescribed lead time. On the other hand, the cycle time depends on the lot release schedule (cf. Pahl *et al*., 2007; Missbauer and Uzsoy, 2011).

Iterative simulation is one approach that tackles this circularity by iterating between a production planning model that determines production quantities based on a prescribed lead time and a discrete-event simulation model that uses these production quantities to calculate new cycle time estimates (cf., among others, Hung and Leachman, 1996; Almeder *et al*., 2009; Irdem *et al*., 2010). However, iterative simulation is not used so far for supply chain planning in the semiconductor industry.

In the previous chapters, the formulation of MPSC problems assume a fixed lead time for all products. This is a limitation. Hence, the intention of this chapter is a first attempt to mitigate this assumption. Therefore, an iterative simulation approach is applied to MPSC problems in order to incorporate load-dependent lead times in the planning approach. Moreover, the convergence of the iterative scheme is investigated by performing designed experiments.

This chapter is organized as follows. First, the related literature is discussed. Then, the iterative scheme is presented. Finally, the results of computational experiments are showed and analyzed.

6.1 Literature Review

An iterative linear programming-simulation scheme is proposed by Hung and Leachman (1996) for production planning in wafer fabs. However, only a limited set of experiments is carried out to assess the performance of the scheme. Kim and Kim (2001) discuss another iterative approach for production planning. Irdem *et al*. (2010) show that only the latter

approach unambiguously converges when it used for production planning in wafer fabs. A disadvantage of the iterative simulation approaches for wafer fabs is the huge computational burden that is caused by the repeated simulation runs. Almeder *et al*. (2009) use iterative simulation in a supply chain context. However, the considered setting is quite different from semiconductor manufacturing.

Another stream of production planning research with load-dependent lead times is related to clearing functions (cf. Selçuk *et al*., 2008; Irdem, 2009; Missbauer and Uzsoy, 2011). A clearing function provides the expected aggregated output of a tool group as a function of an appropriate measure of WIP, typically aggregated over all products. Linearizations of the tool group-specific clearing functions are incorporated into the linear programming formulations for production planning. The clearing functions are derived from simulation output. It is a nontrivial task to fit appropriate clearing functions for the tool groups of a wafer fab. Kacar *et al*. (2012) compare the performance of clearing function- and iterative simulation-based linear programming formulations for production planning in a scaled-down wafer fab. It turns out that the linear programming model based on clearing functions outperforms the iterative simulation-based one with respect to variable production plans and profit.

The decision to look at an iterative scheme was influenced by the fact that the simulation-based framework presented in Chapter 5 can be reused.

6.2 Iterative Simulation Approach

In this section, the iterative simulation approach is described. First, the necessary notation is introduced. The index of the iterations is denoted by $\nu = 1,\ldots,\nu_{\max}$. The maximal number of iterations is set to $\nu_{\max} = 30$. LT_{pm}^{ν} and CT_{pm}^{ν} refer to the lead time and the estimated cycle time for lots of product p in wafer fabs m in the ν-th iteration, respectively. The set of lead times and cycle times in the ν-th iteration considering all products and wafer fabs are defined as follows:

$$LT^{\nu} := \left\{ LT_{pm}^{\nu} \middle| p = 1,\ldots, p_{\max}, m = 1,\ldots, m_{\max} \right\}, \tag{6.1}$$

$$CT^{\nu} := \left\{ CT_{pm}^{\nu} \middle| p = 1,\ldots, p_{\max}, m = 1,\ldots, m_{\max} \right\}. \tag{6.2}$$

The matrix $\widetilde{C}^{\nu}$ that contains the elements cc_{bk}^{ν} describes the capacity consumption matrix used in the ν-th iteration as described in Subsection 3.1.2 (to simplify the notation, the indices related to products and facilities have been omitted). The release schedule considering all products and wafer fabs is denoted by rs^{ν}. The release schedule rs^{ν} is derived from the master plan mp^{ν}. The quantity χ is a smoothing parameter with $0 \le \chi \le 1$. The main steps of the iterative simulation scheme are described by Algorithm 6.1.

Algorithm 6.1: Scheme for the iterative simulation.

1 $\nu := 1$.

2 Initialize LT^1 by using historical data or results from simulation runs where a given demand is used to determine a release schedule.

3 Determine the capacity consumption matrix $\widetilde{C}^{\nu}$ based on LT^{ν}.

4 Solve a given MPSC problem using $\widetilde{C}^{\nu}$ and LT^{ν}, respectively, and derive the release schedule rs^{ν} from mp^{ν}.

5 Perform three independent simulation runs based on rs^{ν}.

6 Estimate the cycle times in CT^{ν} by considering the mean cycle time as obtained from the simulation runs.

7 Update the lead times as follows:

$$LT_{pm}^{\nu+1} := (1-\chi)LT_{pm}^{\nu} + \chi\, CT_{pm}^{\nu},\ \forall p = 1,\ldots,p_{\max},\ \forall m = 1,\ldots,m_{\max}. \quad (6.3)$$

8 Round up $LT_{pm}^{\nu+1}, \forall p = 1,\ldots,p_{\max}, \forall m = 1,\ldots,m_{\max}$ to the next integer.

9 $\nu := \nu + 1$.

10 If $\nu < \nu_{\max}$, then go to Step 3, else stop.

Algorithm 6.1 does not include a convergence condition as a termination criterion since the intention is to investigate the evolution of the lead times in the long run. However, the scheme is limited to thirty iterations to keep the computational burden reasonable. A similar assumption is made by Irdem *et al*. (2010). The selection of the smoothing parameter χ determines how much cycle time information from the simulation is taken into account. Hence, χ values close to 1 have the effect that almost the full cycle time information is considered. Different χ values are considered within the design of experiments.

6.3 Computational Experiments

6.3.1 Design of Experiments

The purpose of the designed experiments consists in examining the effect of three factors from the base and planning system on the performance of the iterative simulation scheme. The demand level (DL) is either low or high to influence the resource utilization. Three different settings for the initial lead times (ILT) of the products are considered. Accurate initial lead times result from Step 2 of the iterative scheme described in Section 6.2. Over-

estimated and under-estimated initial lead times are obtained by increasing and decreasing the accurate values by one period, respectively. Values for χ are 0.20, 0.50, and 1.00. Irdem *et al*. (2010) consider similar settings for DL and χ. A factorial design is used that leads to eighteen factor combinations. To limit the computational effort, one problem scenario is considered per factor combination. Three independent simulation replications are performed for each combination. It allows mitigating the effect of variability in the base system on the results. Totally, 1620 simulation runs are carried out in the experiments. The design of experiments in use is summarized in Table 6.1.

Table 6.1: Design of experiments (III).

Factor	Notation	Level	Number
Planning System			
Initial Lead Time	ILT	overestimated, underestimated, accurate	3
Smoothing parameter	χ	0.20, 0.50, 1.00	3
Base System			
Demand Level	DL	low, high	2
System Capacity	SC	Fixed N_m, breakdown and PM settings	1
Total factor combinations			18
Number of problem scenarios per combination			1
Number of simulation replications per combination for a single iteration			3
Total number of simulation runs			54

The settings related to the base system presented in Subsection 5.3.1 as well as the reduced simulation model described in Subsection 5.3.2 are also used in the present chapter. The experiments are carried by taking RA-MPSC as a planning approach. The main reason for choosing RA-MPSC relies in the fast execution times that allow decreasing the computational effort in the context of an iterative scheme. Similar to the assumption made in Subsection 5.3.3, the length of a planning period in RA-MPSC is set to two days. The revenue and cost settings used by RA-MPSC are identical to the settings used in Chapter 4 to which a scaling down factor of 3.5 is applied due to the modified time unit. The parameter settings of RA-MPSC are also taken from Chapter 4.

6.3.2 Implementation of the Iterative Simulation Approach

The iterative simulation approach is implemented by means of the simulation-based framework that is described in Chapter 5 with the difference that no rolling horizon setting is applied.

6.3.3 Results of Computational Experiments

The convergence of the iterative scheme is investigated under different DL, ILT, and χ factor levels. Therefore, the percentage mean absolute deviation (MAD) of the cycle time of product p in iteration ν is used. It is defined as follows:

$$MAD_p^{\nu} := \frac{100\%}{m_{\max}} \sum_{m=1}^{m_{\max}} \frac{\left| CT_{pm}^{\nu} - \overline{CT}_{pm} \right|}{\overline{CT}_{pm}} , \tag{6.4}$$

where CT_{pm}^{ν} is the cycle time of product p averaged across all lots released from the wafer fab m during iteration ν and

$$\overline{CT}_{pm} := \frac{1}{\nu_{\max}} \sum_{\nu=1}^{\nu_{\max}} CT_{pm}^{\nu} . \tag{6.5}$$

MAD values near zero suggest the convergence of the scheme. A similar measure is used by Hung and Leachman (1996) and Irdem *et al*. (2010). The recommendation stated in the latter to consider the MAD values of the individual products is taken into account. In the following, the MAD values are plotted as a function of the iterations.

Figure 6.1 shows the MAD values for the (DL=High, ILT=Accurate, χ=0.20) case. The demand level setting leads to an average resource utilization of 92%. The overall maximum MAD value of 41.3% is obtained in the first iteration. One can see the continuous decrease of the MAD values in the first five iterations followed by a period of erratic fluctuations for some products. A rather stable level is observed after the thirteenth iteration. The maximum deviation from this iteration onwards reaches 10.8% while for the majority of the products their MAD values do not cross the 5% threshold.

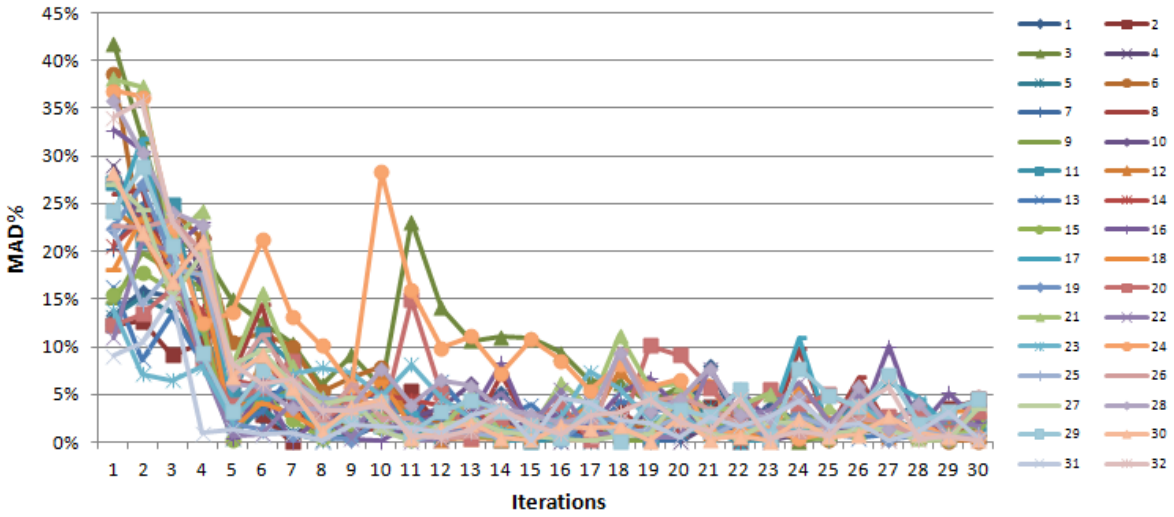

Figure 6.1: MAD in product cycle times for the (DL=High, ILT=Accurate, χ=0.20) case.

Figure 6.2 plots the rounded product lead times used in RA-MPSC as a function of the iterations. For the purpose of this figure, the lead times are averaged across all wafer fabs. The initial setting is four periods for all products. One sees the refinement of the lead times as a result of the iterative scheme, i.e., some product lead times converge to three periods while the others keep the four period setting.

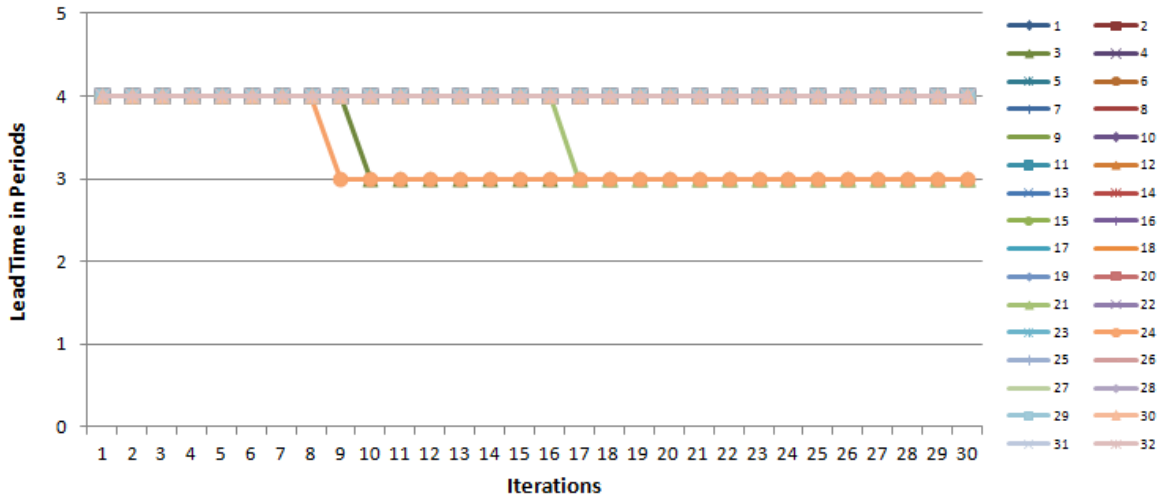

Figure 6.2: Product lead times for the (DL=High, ILT=Accurate, χ =0.20) case.

Then, the effect of ILT settings is of interest. The case of under-estimated ILT is showed in Figures 6.3 and 6.4. One observes higher MAD values in the first five iterations compared to the accurate case. The decrease of the MAD values follows a slower trend. All MAD values fall below the 10% threshold after 16 iterations and the average MAD value is slightly below 5%. The last lead time change occurs in the sixteenth occurrence. Hence, despite the initial bias of one period the iterative scheme seems to converge. A similar pattern can be observed for the cases with over-estimated ILT.

The cases with a low demand level are characterized by lower MAD values at the beginning of the iterative scheme and a steep decrease in the first five iterations. The worst case is observed for the (DL=Low, ILT=Under-estimated, χ =0.20) case where the maximum MAD value reaches 18% in the first iteration, and the last lead time change occurs at the twelfth iteration. The MAD values are below 4% from this iteration onwards. Hence, a convergence pattern is observed as well. This is not surprising since the average resource utilization of 54% leaves certain flexibility to the base system to deal with different initial parameters.

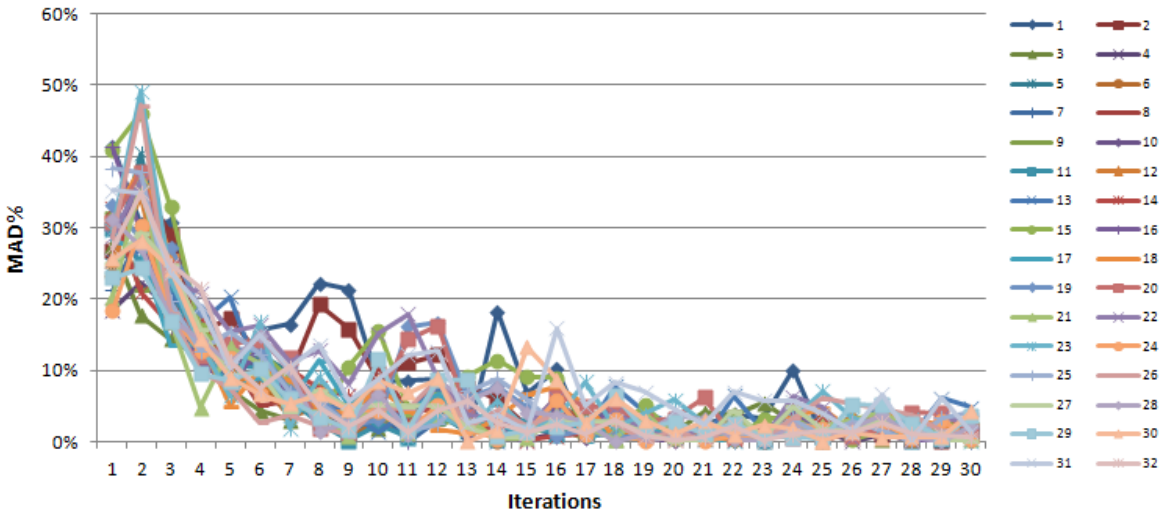

Figure 6.3: MAD in product cycle times for the (DL=High, ILT=Under-estimated, χ =0.20) case.

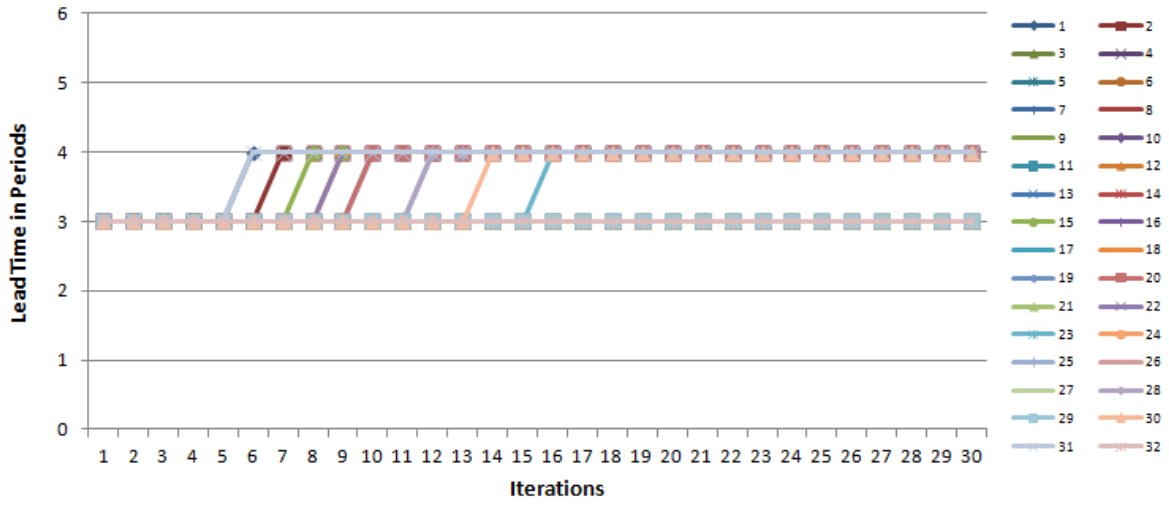

Figure 6.4: Product lead times for the (DL=High, ILT=Under-estimated, χ =0.20) case.

The effect of different χ values is analyzed. Figures 6.5 and 6.6 show the (DL=High, ILT=Accurate, χ =0.50) case. A higher χ value allows for taking more cycle time information from the simulation into account. One observes a much steeper decrease in the first three iterations than for the (DL=High, ILT=Accurate, χ =0.20) case. Also the last lead time change occurs much earlier, i.e., in the seventh iteration. It can be concluded that the χ =0.50 setting expedites the convergence of the iterative scheme.

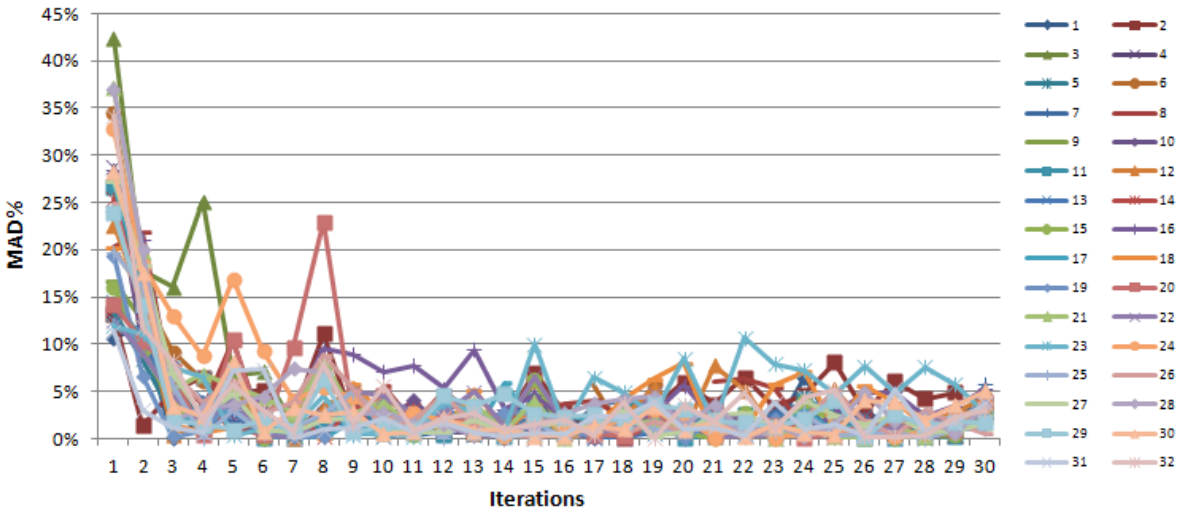

Figure 6.5: MAD in product cycle times for the (DL=High, ILT=Accurate, χ =0.50) case.

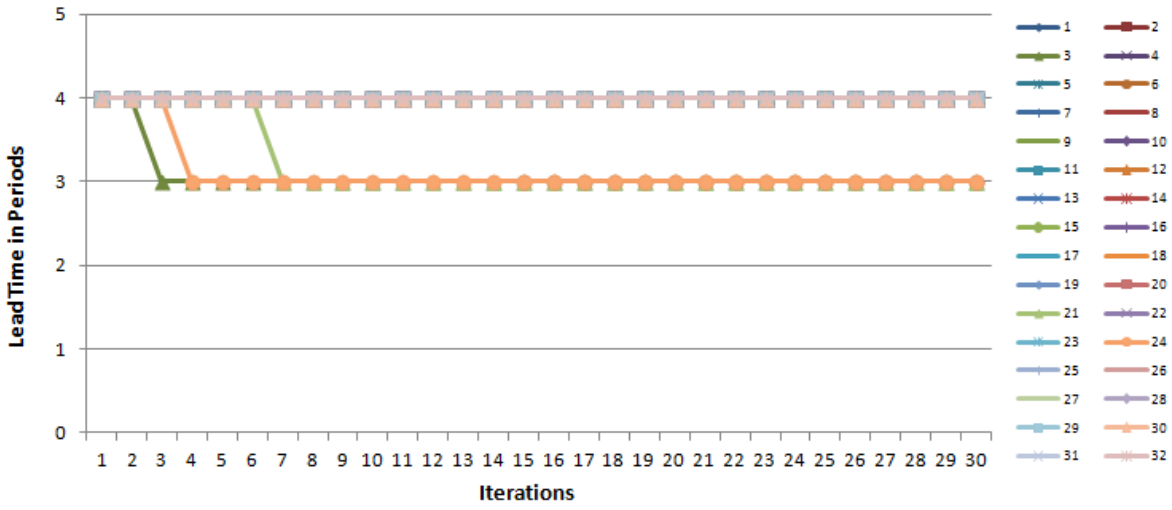

Figure 6.6: Product lead times for the (DL=High, ILT=Accurate, χ =0.50) case.

On the other hand, the MAD values obtained for the cases with χ =1.00 show high fluctuations and no convergence pattern. The worst case is obtained for the (DL=High, ILT=Under-estimated, χ =1.00) case where the overall maximum MAD value reaches 48% in the twentieth iteration. Since more information is taken from the simulation, the convergence of the scheme is subject to the variability of the base system. Hence, it seems to exist a tradeoff between the convergent trend and the convergence speed.

Besides the convergence of the cycle times, it seems important to investigate the impact of the iterative scheme on the objective function values of RA-MPSC. In this situation, the MAD measure is defined as follows:

$$MAD^{\nu} := 100\% \frac{\left| f^{\nu} - \bar{f} \right|}{\bar{f}}, \tag{6.6}$$

where f^{ν} is the objective function value in iteration ν and

$$\bar{f} := \frac{1}{\nu_{\max}} \sum_{\nu=1}^{\nu_{\max}} f^{\nu} \ . \tag{6.7}$$

Figure 6.7 plots the MAD values with respect to objective function values for the (DL=High, ILT=Accurate, χ =0.20) case. One sees a similar pattern as in Figure 6.1, i.e., the MAD values decrease in the first six iterations, followed by erratic fluctuations between the tenth and the eighteenth iterations. After the twentieth iteration, the MAD value stays at a rather low level. A similar convergence behavior is observed for all other cases.

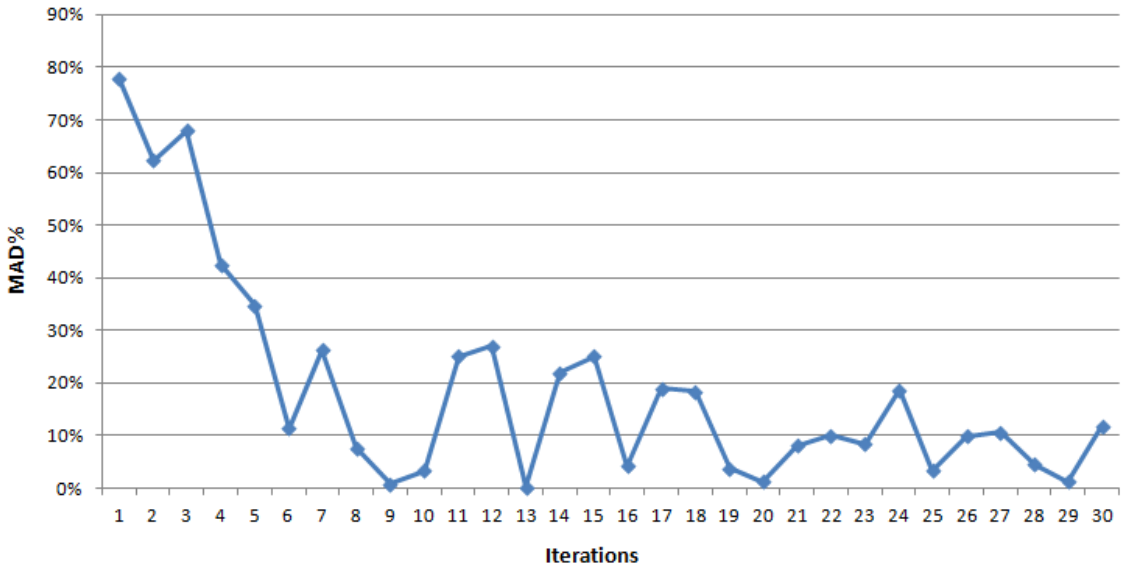

Figure 6.7: MAD in objective function values for the (DL=High, ILT=Accurate, χ =0.20) case.

To ensure the benefits of the iterative scheme, the throughput obtained from the simulation model is investigated as a function of the iterations. For each product and iteration the difference between the realized throughput in all wafer fabs and the average throughput across all iterations is measured without taking absolute values of the summands. It is called mean deviation (MD), and it is defined as follows:

$$MD_p^\nu := 100\% \frac{TP_p^\nu - \overline{TP}_p}{\overline{TP}_p}, \tag{6.8}$$

where TP_p^ν is the throughput of product p released from all wafer fabs of the network during iteration ν and

$$\overline{TP}_p := \frac{1}{\nu_{\max}} \sum_{\nu=1}^{\nu_{\max}} TP_p^\nu \,. \tag{6.9}$$

The mean deviations cumulated across the products are showed in Figure 6.8 for the (DL=High, ILT=Accurate, χ =0.20) case. One can see higher positive deviations towards the last iteration than at the beginning of the iterative scheme, i.e., a higher throughput is reached. On average up to two percent throughput improvement is reached across all products. The trend clearly increases from the nineteenth iteration onwards. A parallel can be drawn with Figure 6.7 that shows the lower MAD values starting from the same iteration. The higher throughput can be explained by the increasing number of products whose lead time is adjusted from four to three periods as showed in Figure 6.2. Lower lead times allow for more production requests planned by RA-MPSC.

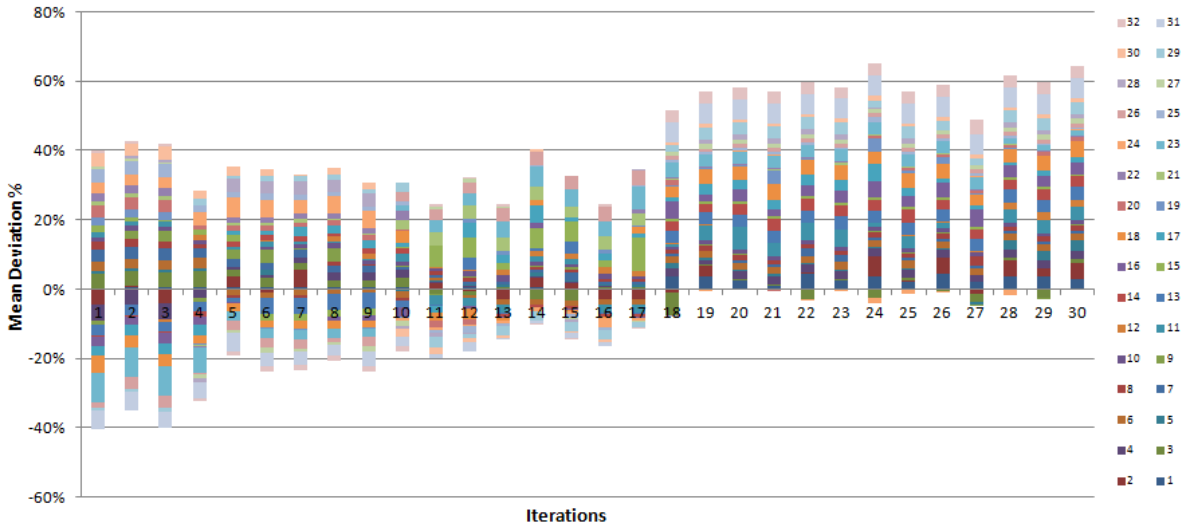

Figure 6.8: MD in realized throughput for the (DL=High, ILT=Accurate, χ =0.20) case.

6.4 Conclusion

This chapter discussed an iterative simulation scheme for MP approaches. The scheme alternates between a rather straightforward planning heuristic and a reduced discrete-event simulation model of a network of wafer fabs. Computational experiments confirmed that the scheme converges after a small number of iterations. It has been also demonstrated that the iterative approach leads to less variable, more profitable production plans compared to plans obtained by the fixed lead time approach. Parts of the results discussed in this chapter are presented in Ponsignon and Mönch (2012c).

7. Conclusion and Future Research

The present thesis tackled MP problems that arise in semiconductor manufacturing, especially in one-layer networks of parallel wafer fabs. An appropriate mathematical formulation of the considered problems has been proposed. Due to the computational complexity given by the problem setting, efficient solution approaches were proposed, i.e., a product-based decomposition scheme, a rule-based assignment scheme, and a GA. The performance of these approaches has been first evaluated by means of single problem instances. Computational experiments allows for differentiating the algorithms with respect to solution quality and computing time.

Because of the uncertainty that is typical for the semiconductor industry, an appropriate simulation-based framework has been suggested for the performance assessment of MP approaches in a rolling horizon setting while considering demand and execution disruptions. Extensive computational experiments confirmed the conflict that exists when striving for large objective function values considering single problem instances and the impact on the performance measure values achieved over the simulation horizon.

Finally, the simulation-based framework has been adapted to an iterative simulation scheme with the intention of incorporating load-dependent lead times in the MP approach. Some computational experiments successfully demonstrated that the scheme converges after a small number of iterations, and that the iterative approach leads to less variable, more profitable production plans compared to plans obtained with a fixed lead time approach.

In the following, some directions for future research are described.

- **Extended problem scope.** The considered manufacturing system could be extended to a multi-layer network. It would incorporate both frontend and backend operations. It implies the usage of a die bank as a decoupling point between both production segments. The anticipated challenge consists in finding an appropriate integrated algorithm that allows for a global optimization of the production. Furthermore, an enhanced network could also include more complex interactions between the fabrication facilities. An emerging trend in semiconductor manufacturing is the so-called borderless fab where the wafer fabrication of highly sophisticated semiconductor devices is split between several wafer fabs (cf. Gan *et al*., 2007). For MP decision, this implies allowing flows of material across the nodes of the considered network. Moreover, the assumption that is made for the computational experiments in Chapters 4, 5, and 6 on the number of bottleneck resources in each wafer fab, i.e., $b_{m,\max} = 1$, could be relaxed. It would be interesting to investigate the impact of a more complex representation of the base system on the MP approach. The formulation

proposed in this thesis is suitable for $b_{max} > 1$. Though, the simulation model would have to be modified. Depending on the researched problem, it also seems possible to extend the simulation-based framework by implementing a capacity planning module that matches aggregated demands and capacities, and that takes decisions on the long-term capacity provision, i.e., by activating or deactivating machines in the base system.

- **Alternative modeling approaches.** The benefit of planning approaches that incorporate uncertainty directly in the planning algorithm such as stochastic programming (cf. Barahona *et al*., 2005; Rastogi *et al*., 2011) can be further researched in the context of MPSC. In a complementary approach to the iterative simulation approach, clearing functions are discussed in Selçuk *et al*. (2008), Irdem (2009), Missbauer and Uzsoy (2011), and Kacar *et al*. (2012) for mid-term production planning problems. However, it seems challenging to try to use clearing functions in a setting different from linear programming.
- **Alternative simulation approach.** Ehm *et al*. (2011b) propose a coarse-grained simulation approach for supply chains that assigns stochastic cycle times to lots given the loading of the facility under consideration and based on empirical cycle time and throughput distributions. This method allows for fast execution times of the simulation. In future research, this approach may be a suitable alternative to the reduction method described in Subsection 5.3.2 since a lower modeling and computational effort is expected.
- **Integrated framework.** The benefits of an integrated simulation-based framework that allows both the performance assessment of planning approaches and the incorporation of load-dependent lead times in the planning algorithm could be analyzed. Figure 7.1 shows an example of an integrated simulation-based framework.

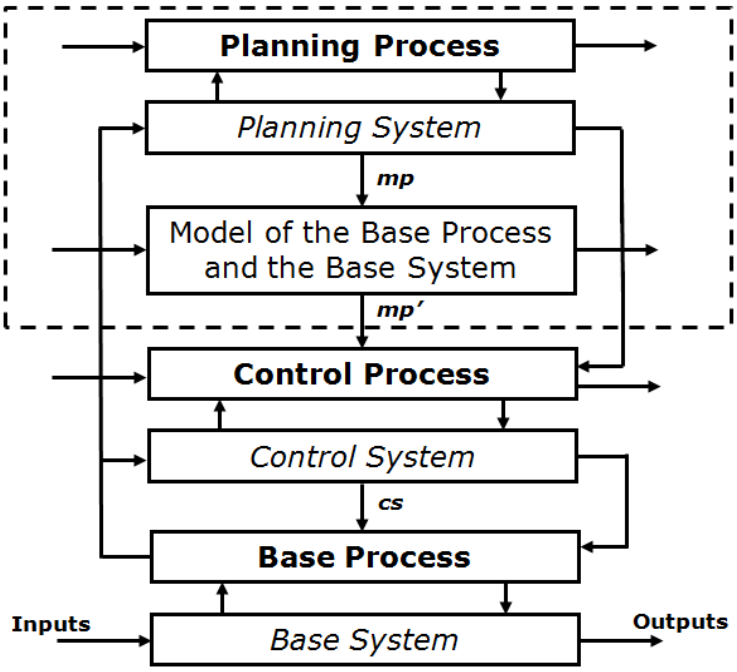

Figure 7.1: Integrated simulation-based framework.

The planning level and the model of the base process and system, which is a representation of the base level, build a subframework for the determination of

realistic lead times. The resulting master plan *mp'* is then provided to the base level via the control level for its evaluation in a rolling horizon setting while confronted with uncertainty. Because of the potentially high computational burden, the practicability of such an approach has to be investigated.

References

Almeder, C., Preusser, M., Hartl, R. F. (2009) 'Simulation and Optimization of Supply Chains: Alternative or Complementary Approaches?', *OR Spectrum*, Vol. 31, No. 1, pp.95-119.

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

Aytug, H., Khouja, M., Vergara, F. E. (2003) 'Use of Genetic Algorithms to Solve Production and Operations Management Problems: A Review', *International Journal of Production Research*, Vol. 41, No. 17, pp.3955-4009.

Banks, J., Carson, J. S., Nelson, B. L., Nicol, D. M. (2010) *Discrete-Event System Simulation*, 5th ed., Prentice-Hall.

Barahona, F., Bermon, S., Günlük, O., Hood, S. (2005) 'Robust Capacity Planning in Semiconductor Manufacturing', *Naval Research Logistics*, Vol. 52, No. 5, pp.459-468.

Bermon, S., Hood, S. J. (1999) 'Capacity Optimization Planning System (CAPS)', *Interfaces*, Vol. 29, No. 5, pp.31-50.

Bhagwat, R., Sharma, M. K. (2007) 'Performance Measurement of Supply Chain Management: A Balanced Scorecard Approach', *Computers & Industrial Engineering*, Vol. 53, No. 1, pp.43-62.

Brandimarte, P. (2006) 'Multi-Item Capacitated Lot-Sizing with Demand Uncertainty', *International Journal of Production Research*, Vol. 44, No. 15, pp.2997-3022.

Chern, C.-C., Hsieh, J.-S. (2007) 'A Heuristic Algorithm for Master Planning that Satisfies Multiple Objectives', *Computers & Operations Research*, Vol. 34. No. 11, pp.3491-3513.

Chien, C.-F. (2007) 'Made in Taiwan: Shifting Paradigms in High-tech Industries', *Industrial Engineer*, Vol. 39, No. 2, pp.47-49.

Chien, C.-F., Dauzère-Pérès, S., Ehm, H., Fowler, J. W., Jiang, Z., Krishnaswamy, S., Mönch, L., Uzsoy, R. (2011) 'Modeling and Analysis of Semiconductor Manufacturing in a Shrinking World: Challenges and Successes', *European Journal of Industrial Engineering*, Vol. 5, No. 3, pp.254-271.

Chien, C.-F., Wang, J. K., Chang, T.-C., Wu, W.-C. (2007) 'Economic Analysis of 450mm Wafer Migration', In *Proceedings of the International Symposium on Semiconductor Manufacturing*, pp.1-4.

Chong, C. S., Lendermann, P., Gan, B. P., Duarte, B., Fowler, J. W., Callarman, T. E. (2006) 'Development and Analysis of a Customer-Demand Driven Semiconductor Supply Chain Model Using the High Level Architecture', *International Journal of Simulation and Process Modelling*, Vol. 2, No. 3-4, pp.210-221.

Deb, K. (2000) 'An Efficient Constraint Handling Method for Genetic Algorithms', *Computer Methods in Applied Mechanics and Engineering*, Vol. 186, No. 2-4, pp.311-338.

Denton, B. T., Forrest, J., Milne, R. J. (2006) 'IBM Solves a Mixed-Integer Program to Optimize Its Semiconductor Supply Chain', *Interfaces*, Vol. 36, No. 5, pp.386-399.

Dueck, G., Scheuer, T. (1990) 'Threshold Accepting: A General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing', *Journal of Computational Physics*, Vol. 90, No. 1, pp.161-175.

Ehm, H., Ponsignon, T. (2010) 'Position Statement on Grand Challenges for Discrete Event Logistics Systems', In *Dagstuhl Seminar Proceedings*, pp.1-9.

Ehm, H., Ponsignon, T., Kaufmann, T. (2011a) 'The Global Supply Chain Is Our New Fab: Integration and Automation Challenges', In *Proceedings of the Advanced Semiconductor Manufacturing Conference*, pp.1-6.

Ehm, H., Wenke, H., Mönch, L., Ponsignon, T., Forstner, L. (2011b) 'Towards a Supply Chain Simulation Reference Model for the Semiconductor Industry', In *Proceedings of the Winter Simulation Conference*, pp.2124-2135.

Fleischmann, B., Meyr, H., Wagner, M. (2007) 'Advanced Planning', *Supply Chain Management and Advanced Planning: Concepts, Models, Software, and Case Studies*, 4th ed., Stadtler, H., Kilger, C. (eds.), Springer, pp.81-106.

Florian, M., Lenstra, J. K., Rinnooy Kan, A. H. G. (1980) 'Deterministic Production Planning: Algorithms and Complexity', *Management Science*, Vol. 26, No. 7, pp.669-679.

Gan, B. P., Liow., M., Gupta, A. K., Lendermann, P., Turner, S. J., Wang, W. G. (2007) 'Analysis of a Borderless Fab Using Interoperating AutoSched AP Models', *International Journal of Production Research*, Vol. 45, No. 3, pp.675-697.

Garey, M. R., Johnson, D. S. (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman.

Genin, P., Thomas, A., Lamouri, S. (2007) 'How to Manage Robust Tactical Planning with an APS (Advanced Planning Systems)', *Journal of Intelligent Manufacturing*, Vol. 18, No. 2, pp.209-221.

Goldberg, D. E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley.

Gunasekaran, A., Patel, C., McGaughey, R. E. (2004) 'A Framework for Supply Chain Performance Measurement', *International Journal of Production Economics*, Vol. 87, No. 3, pp.333-347.

Gupta, J. N. D., Ruiz, R., Fowler, J. W., Mason, S. J. (2006) 'Operational Planning and Control of Semiconductor Wafer Fabrication', *Production Planning and Control*, Vol. 17, No. 7, pp.639-647.

Hopp, W. J., Spearman, M. L. (1996) *Factory Physics: Foundations of Manufacturing Management*, Irwin.

Horiguchi, K., Raghavani, N., Uzsoy, R., Venkatheswaran, S. (2001) 'Finite-Capacity Production Planning Algorithms for a Semiconductor Wafer Fabrication Facility', *International Journal of Production Research*, Vol. 39, No. 5, pp.825-842.

Hornung, A., Mönch, L. (2008) 'Heuristic Approaches for Determining Minimum Cost Delivery Quantities in Supply Chains', *European Journal of Industrial Engineering*, Vol. 2, No. 4, pp.377-400.

Huang, M.-G., Chang, P.-L., Chous, Y.-C. (2007) 'Demand Forecasting and Smoothing Capacity Planning for Products with High Random Demand Volatility', *International Journal of Production Research*, Vol. 46, No. 12, pp.3223-3239.

Hung, Y.-F., Leachman, R. C. (1996) 'A Production Planning Methodology for Semiconductor Manufacturing based on Iterative Simulation and Linear Programming Calculations', *IEEE Transactions on Semiconductor Manufacturing*, Vol. 9, No. 2, pp.257-269.

Hung, Y.-F., Leachman, R. C. (1999) 'Reduced Simulation Models of Wafer Fabrication Facilities', *International Journal of Production Research*, Vol. 37, No. 12, pp.2685-2701.

Irdem, D. F. (2009) *Evaluation of Clearing Functions' Fitting Methodology and Performance for Production Planning Models*, Master Thesis, North Carolina State University.

Irdem, D. F., Kacar, N. B., Uzsoy, R. (2010) 'An Exploratory Analysis of Two Iterative Linear Programming-Simulation Approaches for Production Planning', *IEEE Transactions on Semiconductor Manufacturing*, Vol. 23, No. 3, pp.442-455.

ITRS (2010) 'More-than-Moore, White Paper', *http://www.itrs.net*.

ITRS (2011) 'International Technology Roadmap for Semiconductors, 2011 Edition, Executive Summary', *http://www.itrs.net*.

Kacar, N. B., Irdem, D. F., Uzsoy, R. (2012) 'An Experimental Comparison of Production Planning Using Clearing Functions and Iterative Linear Programming-Simulation Algorithms', *IEEE Transactions on Semiconductor Manufacturing*, Vol. 25, No. 1, pp.104-107.

Kallrath, J., Maindl, T. I. (2006) *Real Optimization with SAP® APO*, Springer.

Kim, B. and Kim, S. (2001) 'Extended Model for a Hybrid Production Planning Approach', *International Journal of Production Economics*, Vol. 73, No. 2, pp.165-173.

Kimms, A. (1998) 'Stability Measures for Rolling Schedules with Applications to Capacity Expansion Planning, Master Production Scheduling and Lot Sizing', *Omega – The International Journal of Management Science*, Vol. 26, No. 3, pp.355-366.

Kingman, J. F. C. (1961) 'The Single Server Queue in Heavy Traffic', *Proceedings of the Cambridge Philosophical Society*, Vol. 57, No. 4, pp.902-904.

Kleijnen, J. P. C. (2005) 'Supply Chain Simulation Tools and Techniques: A Survey', *International Journal of Simulation and Process Modelling*, Vol. 1, No. 1/2, pp.82-89.

Kumar, P. R. (1993) 'Re-Entrant Lines', *Queueing Systems*, Vol. 13, No. 1-3, pp.87-110.

Law, A. M., Kelton, W. D. (2000) *Simulation Modeling and Analysis*, 3rd ed., McGraw-Hill.

Lee, H. L. (2004) 'The Triple-A Supply Chain', *Harvard Business Review*, Vol. 82, No. 10, pp.102-112.

Lin, N.-P. (1989) *Master Production Scheduling in Uncertain Environments*, Ph.D. Dissertation, Ohio State University.

MASMLab (1997) 'Test Data Sets', *http://www.eas.asu.edu/~masmlab*.

Michalewicz, Z. (1996) *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd ed., Springer.

Missbauer, H., Uzsoy, R. (2011) 'Optimization Models of Production Planning Problems', *Planning Production and Inventories in the Extended Enterprise: A State of the Art Handbook*, Volume 1, Kempf, K. G., Keskinocak, P., Uzsoy, R. (eds.), Springer, pp.437-507.

Mönch, L. (2007) 'Simulation-based Benchmarking of Production Control Schemes for Complex Manufacturing Systems', *Control Engineering Practice*, Vol. 15, No. 11, pp.1381-1393.

Mönch, L. (2008) 'Simulationsbasierte Leistungsbewertung von Planungsverfahren für komplexe Produktionssysteme', *Intelligent Decision Support: Current Challenges and Approaches*, Bortfeld, A., Homberger, J., Kopfer, H., Pankratz, G., Strangmeier, R. (eds.), Gabler, pp.213-228.

Mönch, L., Fowler, J. W., Dauzère-Pérès, S., Mason, S. J., Rose, O. (2011) 'A Survey of Problems, Solution Techniques, and Future Challenges in Scheduling Semiconductor Manufacturing Operations', *Journal of Scheduling*, Vol. 14, No. 6, pp.583-599.

Mönch, L., Gmilkowsky, P. (2001) 'Steuerung des Waferfertigungsprozesses: ein agentenorientierter Ansatz', *Industrie Management*, Vol. 17, No. 6, pp.17-20.

Mönch, L., Rose, O., Sturm, R. (2003) 'Simulation Framework for Performance Assessment of Shop-Floor Control Systems', *SIMULATION: Transactions of the Society for Modelling and Simulation International*, Vol. 79, No. 3, pp.163-170.

Montgomery, D. C. (2008) *Design and Analysis of Experiments*, 7th ed., Wiley.

Moore, G. E. (1965) 'Cramming More Components Onto Integrated Circuits', *Electronics*, Vol. 38, No. 8, pp.114-117.

Moscato, P., Fontanari, J. (1990) 'Stochastic Versus Deterministic Update in Simulated Annealing', *Physics Letters A*, Vol. 146, No. 4, pp.204-208.

Mula, J., Poler, R., García-Sabater, J. P., Lario, F. C (2006) 'Models for Production Planning Under Uncertainty: A Review', *International Journal of Production Economics*, Vol. 103, No. 1, pp.271-285.

Pahl, J., Voß, S., Woodruff, D. L. (2007) 'Production Planning with Load Dependent Lead Times: An Update of Research', *Annals of Operations Research*, Vol. 153, No. 1, pp.297-345.

Pochet, Y., Wolsey, L. A. (2006) *Production Planning by Mixed-Integer Programming*, Springer.

Ponsignon, T., Mönch, L. (2012a) 'Heuristic Approaches for Master Planning in Semiconductor Manufacturing', *Computers & Operations Research*, Vol. 39, No. 3, pp.479-491.

Ponsignon, T., Mönch, L. (2012b) 'Simulation-based Performance Assessment of Master Planning Approaches in Semiconductor Manufacturing', submitted.

Ponsignon, T., Mönch, L. (2012c) 'Using Iterative Simulation to Incorporate Load-Dependent Lead Times in Master Planning Heuristics', In *Proceedings of the Winter Simulation Conference*, in press.

Rardin, R. L., Uzsoy, R. (2001) 'Experimental Evaluation of Heuristic Optimization Algorithms: A Tutorial', *Journal of Heuristics*, Vol. 7, No. 3, pp.261-304.

Rastogi, A. P., Fowler, J. W., Carlyle, W. M., Araz, O. M., Maltz, A., Büke, B. (2011) 'Supply Network Capacity Planning for Semiconductor Manufacturing with Uncertain Demand and Correlation in Demand Considerations', *International Journal of Production Economics*, Vol. 134, No. 2, pp.322-332.

Robinson, E. P., Sahin, F., Gao, L.-L. (2007) 'Master Production Schedule Time Interval Strategies in Make-to-Order Supply Chains', *International Journal of Production Research*, Vol. 46, No. 7, pp.1933-1954.

Russell, R. A., Urban, T. L. (1993) 'Horizon Extension for Rolling Production Schedules: Length and Accuracy Requirements', *International Journal of Production Economics*, Vol. 29, No. 1, pp.111-122.

Schömig, A., Fowler, J. W. (2000) 'Modelling Semiconductor Manufacturing Operations', In *Proceedings of* the *9th ASIM Dedicated Conference Simulation in Production and Logistics*, pp. 55-64.

Selçuk, B., Fransoo, J. C., De Kok, A. G. (2008) 'Work-In-Process in Supply Chain Operations Planning', *IIE Transactions*, Vol. 40, No. 3, pp.206-220.

SEMI (2004) 'SEMI E10-0304E – Specification for Definition and Measurement of Equipment Reliability, Availability, and Maintainability (RAM)', *http://www.semi.org*.

Spitter, J. M. (2005) *Rolling Schedule Approaches for Supply Chain Operations Planning*, Ph.D. Dissertation, University Press Eindhoven.

Sridharan, V., Berry, W. L., Udayabhanu, V. (1988) 'Measuring Master Production Schedule Stability under Rolling Planning Horizons', *Decision Sciences*, Vol. 19, No. 1, pp.147-166.

Stadtler, H. (2007) 'Supply Chain Management – An Overview', *Supply Chain Management and Advanced Planning: Concepts, Models, Software, and Case Studies*, 4th ed., Stadtler, H., Kilger, C. (eds.), Springer, pp.9-36.

Stadtler, H. (2012) 'Master Planning – Supply Network Planning', *Advanced Planning in Supply Chains: Illustrating the Concepts Using an SAP® APO Case Study*, 4th ed., Stadtler, H., Fleischmann, B., Grunow, M., Meyr, H., Sürie, C. (eds.), Springer, pp.109-148.

Stray, J., Fowler, J. W., Carlyle, M., Rastogi, A. P. (2006) 'Enterprise-wide Semiconductor Resource Planning', *IEEE Transactions on Semiconductor Manufacturing*, Vol. 19, No. 2, pp.259-268.

Supply Chain Council (2012) 'Supply Chain Operations Reference (SCOR) 10', *http://www.supply-chain.org*.

Tang, O., Grubbström, R. W. (2002) 'Planning and Replanning the Master Production Schedule Under Demand Uncertainty', *International Journal of Production Economics*, Vol. 78, No. 3, pp.323-334.

Tempelmeier, H. (2001) 'Supply Chain Planning with Advanced Planning Systems', In *Proceedings of the Aegean International Conference on Design and Analysis of Manufacturing Systems*.

Uzsoy, R., Lee, C.-Y., 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*, Vol. 24, No. 4, pp.47-60.

Uzsoy, R., Lee, C.-Y., Martin-Vega, L. A. (1994) 'A Review of Production Planning and Scheduling Models in the Semiconductor Industry Part II: Shop-Floor Control', *IIE Transactions*, Vol. 26, No. 5, pp.44-55.

Venkataraman, R., Nathan, J. (1999) 'Effect of Forecast Errors on Rolling Horizon Master Production Schedule Cost Performance for Various Replanning Intervals', *Production Planning and Control*, Vol. 10, No. 7, pp.682-689.

Venkateswaran, J., Son, Y.-J. (2005) 'Hybrid System Dynamic–Discrete Event Simulation-based Architecture for Hierarchical Production Planning', *International Journal of Production Research*, Vol. 43, No. 20, pp.4397-4429.

Vieira, G. E. (2006) 'Understanding Master Production Scheduling from a Practical Perspective: Fundamentals, Heuristics, and Implementations', *Handbook of Production Scheduling*, Hermann, J. W. (ed.), Springer, pp.149-176.

Vieira, G. E., Ribas, P. C. (2008) 'Fractional Factorial Analysis to the Configuration of Simulated Annealing Applied to the Multi-Objective Optimization of Master Production Scheduling Problems', *International Journal of Production Research*, Vol. 46, No. 11, pp. 3007-3026.

Völker, S., Gmilkowsky, P. (2003) 'Reduced Discrete-Event Simulation Models for Medium-term Production Scheduling', *Systems Analysis Modeling Simulation*, Vol. 43, No. 7, pp.867-883.

Voß, S., Woodruff, D. L. (2006) *Introduction to Computational Optimization Models for Production Planning in a Supply Chain*, 2nd ed., Springer.

Wall, M. (2012) 'GALib – A C++ Library of Genetic Algorithm Components', *http://lancet.mit.edu/ga*.

Wilcoxon, F. (1945) 'Individual Comparison by Ranking Methods', *Biometrics Bulletin*, Vol. 1, No. 6, pp.80-83.

WSTS (2012) 'Worldwide Semiconductor Trade Statistics, Market Statistic Reports', *http://www.wsts.org*.

Wu, J.-Z., Chien, C.-F., Gen, M. (2012) 'Coordinating Strategic Outsourcing Decisions for Semiconductor Assembly Using a Bi-Objective Genetic Algorithm', *International Journal of Production Research*, Vol. 50, No. 1, pp.235-260.

Xie, J., Lee, T. S., Zhao, X. (2004) 'Impact of Forecasting Errors on the Performance of Capacitated Multi-Item Production Systems', *Computers & Industrial Engineering*, Vol. 46, No. 2, pp.205-219.

Zant, P. v. (1996) *Microchip Fabrication: A Practical Guide to Semiconductor Processing*, 3rd. ed., McGraw-Hill.

Zhao, X., Xie, J., Jiang, Q. (2004) 'Lot-sizing Rule and Freezing the Master Production Schedule under Capacity Constraint and Deterministic Demand', *Production and Operations Management*, Vol. 10, No. 1, pp.45-67.

Zobolas, G. I., Tarantilis, C. D., Ioannou, G. (2008) 'Extending Capacity Planning by Positive Lead Time and Optional Overtime, Earliness and Tardiness for Effective Master Production Scheduling', *International Journal of Production Research*, Vol. 46, No. 12, pp.3359-3386.

Curriculum Vitae

Personal Details

Name	Thomas Ponsignon
Date of Birth	December 6th, 1983
Place of Birth	Châlons-Sur-Marne, France

Education

2006 – Present	**External PhD Student** University of Hagen, Germany
2001 – 2006	**Dual Master of Engineering in Production and Automation** University of Applied Sciences Munich, Germany EPF-Ecole d'Ingénieurs Sceaux, France
2001	**Dual Baccalauréat and Abitur** Lycée Pierre Bayen, Châlons-En-Champagne, France

Work Experience

2011 – Present	**Staff Engineer Supply Chain** Infineon Technologies AG Munich, Germany
2010 – 2011	**Supply Chain Specialist** Infineon Technologies Ltd. Dublin, Ireland
2006 – 2010	**PhD Student** Infineon Technologies AG Munich, Germany