universität wien

DIPLOMARBEIT

Titel der Diplomarbeit

A Product Mix and a Material Flow Problem Concerning the Semiconductor Manufacturing Industry

Verfasserin

Vera Hager

gemeinsam mit

Karin Spannraft

angestrebter akademischer Grad

Magistra/Magister der Sozial- und Wirtschaftswissenschaften
(Mag. rer. soc. oec.)

Wien, im Dezember 2009

Studienkennzahl lt. Studienblatt:	157
Studienrichtung lt. Studienblatt:	Internationale Betriebswirtschaft
Betreuer/Betreuerin:	o. Univ.-Prof. Dipl.-Ing. Dr. Richard F. Hartl

Ehrenwörtliche Erklärung

Wir versichern hiermit, dass wir die vorliegende Arbeit mit dem Thema

A Product Mix and a Material Flow Problem Concerning the Semiconductor Manufacturing Industry

selbstständig verfasst und alle uns zuteil gewordenen Hilfen sowie das benutze Schrifttum am Ende der Arbeit angegeben haben. Es wurden keine unerlaubten Hilfen eingesetzt und die vorliegende Arbeit wurde bisher in keiner Form als Prüfungsarbeit vorgelegt.

Wien, im Dezember 2009

Acknowledgement

We would like to dedicate this thesis to our parents for supporting and financing our studies.

Furthermore, we take this opportunity to thank all those who have supported us in preparing and finishing our work. Without the help of all those people, we would have never been able to complete it.

In particular, we wish to thank Dr. Martin Romauch for his active support in the preparation of the thesis. We are grateful for his helpful suggestions and patience.

We would also like to thank all those who have helped us in preparing our assignment. We are grateful for the help of o. Univ.-Prof. Dipl.-Ing. Dr. Richard F. Hartl, Sophie Parragh PhD and Mag. Dipl.-Ing. Dr. Verena Schmid, who supported us with useful information and helped us to overcome any kind of problem. Additionally, we want to thank Mag. Eva Ferrigato for proofreading our work.

Table of Contents

List of figures

List of abbreviations

ABS	anti-lock braking system
AG	Aktiengesellschaft / stock corporation
AGV	Automatic guided vehicles
AMHS	Automated material handling system
CEO	Chief executive officer
CMOS	Complementary metal–oxide–semiconductor
CPE	Customer premises equipment
LPT	Longest processing time
CR	Critical ratio
CVRP	Capacitated Vehicle Routing Problem
DRAM	Dynamic random access memory
EBIT	Earnings before interest and taxes
EDD	Earliest due date
EDGE	Enhanced Data rates for GSM Evolution
ESD	Earliest start date
ESP	Electronic Stability Programme
FA	Factory automation
FIFO	First in First out
GPRS	General packet radio service
GRASP	Greedy Randomized Adaptive Search Procedures
GSM	Global system for mobile communication
H_2	Hydrogen
H_2O_2	Hydrogen peroxide
HV	Highest value

IC	Integrated circuit
ITRS	International Technology Roadmap for Semiconductors
mm	Micrometer
MOSFET	metal–oxide–semiconductor field-effect transistor
NASA	National Aeronautics and Space Administration
O_2	Oxide
OVRP	Open Vehicle Routing Problem
PC	Personal computer
R&D	Research & Development
RF	Radio frequency
RGV	Rail guided vehicles
Rm	Random
SA	Simulated Annealing
S_i	Silicon
S_iO_2	Silicon oxide
SPT	Shortest processing time
SrPT	Shortest remaining processing time
TPM	Trusted platform module
TSP	Traveling Salesman Problem
VNS	Variable neighborhood search
VRP	Vehicle Routing Problem
VRPB	Vehicle Routing Problem with Backhauls
VRPPD	Vehicle Routing Problem with Pick up and Delivery

Introduction and Problem Description

The overall purpose of an enterprise consists of achieving the highest possible gain without extensive investments and disbursements. Exactly current economic instances enhance the pressure on companies to undertake extensive savings in order to ensure their continuity and to overcome challenges concerning their daily business. In most instances the workforce is the first position which is subject to abridgements. To prevent these retrenchments, the underlying work attempts to accomplish savings in their current expenditures and their utilized time units in order to pervade production orders while optimizing the overall production process.

The subjacent thesis is composed in collaboration with Infineon Technologies Austria AG which is a well acquainted business in the Austrian semiconductor manufacturing industry. As the whole production process in this branch is subject to a tremendous level of complexity, the work abstracts two different problems which are associated to Infineon Technologies Austria AG. More precisely the thesis deals with a product mix problem and a material flow problem

The product mix problem mainly emphasizes on maximizing the company's profit by taking capacity and demand restrictions into account. An important characteristic of this model insists on the decomposition of products to jobs and their further assignment to different tool groups.

However, the material flow problem aims at acquiring the optimal allocation of lots to machines while curtailing the overall time units for incoming production orders. The main property of the subjacent issue is the affinity to the well established open Vehicle Routing Problem, which targets to minimize its costs or time units while servicing customers by means of several vehicles. Furthermore, the material flow problem reveals the quick achievement of complexity limits in the implementation and thus attempts to ameliorate its boundaries of intricacy and lot assignment through heuristics and priority rules.

The work is organized as follows: In the first part, the organization of the company and as well the high-complex production process with its enormous requirements of high cleanness standards are outlined. The subsequent chapters deal with the presentation of the two underlying production issues, its mathematical formulation and capabilities for their optimization.

The second division of the work puts emphasis on the implementation of the theoretical models by means of two software packages namely XPRESS and C++. To evaluate the proposed models, generated results of the exact and heuristic approaches are assessed with respect to their computational times and deviations of optimality. Concluding remarks are quoted at the end of the thesis.

1. Infineon – The Company

1.1 Infineon Technologies AG

In 1990 Infineon Technologies AG was incorporated due to the fact that the semiconductor production was outsourced from the holding company, Siemens AG. The head office of the newly founded company is located in Neubiberg near Munich, Germany. Today Infineon Technologies obtains a global performance operating through its subsidiaries spread over all five continents, totaling in 58 countries, whereat the majority of their operations and investments is concentrated on Central Europe, Asia and North America[15] In the year 2006, Infineon Technologies realized a further step in its strategic by carving out its memory products division to a new company named Qimonda. Initially Qimonda resides as an entirely owned subsidiary of Infineon Technologies[56] Since the beginning of the calendar year 2007 Qimonda suffers great decreases in turnover through furthermore declining prices of its products. Due to this fact Infineon Technologies strives to divest its remaining 77, 5 % shareholding of Qimonda in the fiscal year 2008[15] Besides Qimonda, Infineon Technologies AG holds a 100 % share of the company Comneon Solutions, which is a leading provider of software for mobile communications[13]

In order to remain an innovative, foresighted and ambitious company, Infineon Technologies emphasizes its business on three values, which are significantly for the present economic status:

- Efficiency in Energy
- Communications
- Security

In addition to the semiconductor fabrication for automotive and industrial electronics Infineon Technologies provides chip cards as well as security

and communication devices. The outline below gives an impression of the comprehensive diversity of their existing product lines[15]

Automotive

components of the automotive infotainment (dashboard, navigation, multimedia), body and convenience elements (lightening, seat control), safety and vehicle devices (ABS / ESP, airbag)

Industry and Multimarket

renewable energy (solar, wind), automation, medical technology, power supplies (notebooks, servers), computer peripherals

Due to their creation of substantial know how and competence over the last 40 years in the sector of automotive, industry and Multimarket, Infineon Technologies holds a leading position in its global target markets[16]

Chip card and Security

payment, identification, SIM cards, pay-TV, computer and network security platforms, TPM[1]

Within in the market of Chip cards Infineon Technologies obtains the prime position with a market share of 27%.

Wireless Solutions and Communications

mobile phone platforms, mobile software, Radio Frequency (RF) technology, TV and satellite receiver, navigation, broadband CPE[2], wireless infrastructure and telephones, home network

In the field of RF technologies for mobile phones and in the wireline access market Infineon Technologies occupies the first place with a market share of 22 %.

The products of Infineon Technologies stand for unique quality, state-of-the-art technology and enormous persistency. Due to their profound knowledge and experiences over years in the semiconductor manufacturing, their steady improvement of existing products, their rapid adjustments as a result of quick changes in consumers' demands and their

[1] Trusted Platform Module: a specified chip for enhancing computer security
[2] Customer Premises Equipment: Devices which are interfaced with telephone or data network (e.g. fax machines, modes or telephones)

focus on environmental changes and requirements, Infineon Technologies obtains the leading position in the market[15]

The following utterances stated below should emphasize the leading position of Infineon Technologies AG.

“Semiconductors for power electronics are the key to efficient energy management. Infineon is the global market leader in power electronics.”[24]

“As the leader in this market and in power technologies, we can enable the industry to minimize power losses and maximize energy savings along the entire power cycle: generation, transmission and consumption.”[24]

“It is expected that 80 percent of all electrical energy will be controlled and regulated by power electronics by 2010.”[24]

Infineon Technologies AG recognized one of the first companies to develop modern technologies due to the awareness of finite energy resources and global warming. Several promising statements of Peter Bauer, CEO of Infineon Technologies AG, are given below out of the interview about energy efficiency used as competitive advantage.

“Our products make a fundamental contribution to energy efficiency. They operate in a phenomenally wide variety of everyday gadgets, such as PCs, notebooks, washing machines, cookers, lamps, air conditioning systems and so on, without the public at large being aware of the fact.”[36]

“[…] our chips are to be found in many industrial drives and consumer products […]. Using our products, these drives can be controlled so as to reduce the energy requirement by 30 to 40 percent.”[36]

“It pursues several goals. Firstly, to raise awareness of energy efficiency both internally and externally; secondly, to strengthen our market position and to address existing and new customers; thirdly, to initiate government

projects all over the world or get ourselves involved in existing projects devoted to energy efficiency."[36]

"We advise and throw light on existing possibilities for energy-efficient applications and on potential future developments. We know what is technically still feasible, […]. Infineon teams worldwide are working on raising the awareness of energy efficiency issues among politicians and organizations."[36]

"The market opportunities for our entire product portfolio are better than ever before. The energy-efficient product market will provide sustained growth. We started to position ourselves for this sphere in good time."[36]

Infineon Technologies AG employs 28.025 workers (status 12/2008 without Qimonda) around the world (status 09/2007 29.598 employees) and generated in the business year 2008 annual sales of 4.321 billion Euros (status fiscal year 2005 6.8 billion Euros). There leading position in the market arise form their profound understanding for technology (6.270 workers operate in the R&D sector) resulting in over 21.600 patents[14]

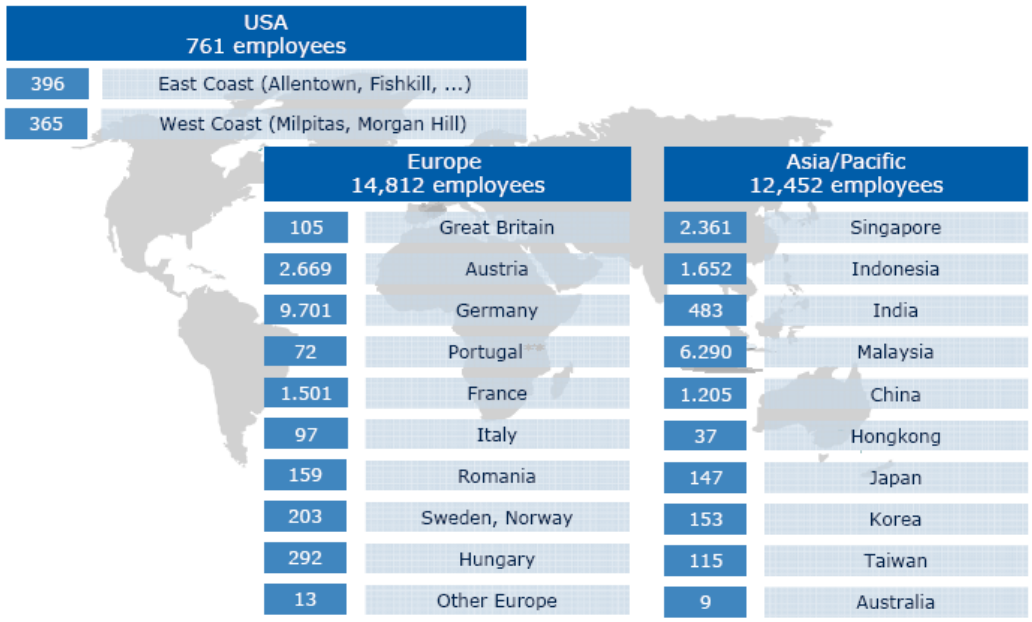

Figure 1: Number of employees per country[14]

1.2 Infineon Technologies Austria AG

In Austria the Infineon Technologies AG is represented through five subsidiaries, whereat the headquarter is located in Villach. Additional associated companies are established in Vienna, Linz, Graz and Klagenfurt. To obtain an overview of the track record of Infineon Technologies Austria AG over the last years the following chronological outline should help[15]

1970: Incorporation of the production entity of diodes in Villach
1979: Incorporation of the development entity in Villach
Production of the 4 inches (100 mm[3]) wafers initiated
1984: Wafer production of 5 inches (127 mm) started
1987: Expansion of the research and development department in Villach
1997: Villach receives the status of being a hub of competence for power electronics
Begin of the 6 inches (150 mm) wafer production
1998: Formation of the subsidiary in Graz
1999: Incorporation of the development centre in Linz
2000: Initial public offering of Infineon Technologies AG
Software Development centre COMNEON was incorporated
Start of the 8 inches (200 mm) wafer production
2001: Another incorporation of a development centre in Vienna until 2003
2003: Villach obtains the status and functions of an headquarter
2004: Formation of another subsidiary in Klagenfurt
2005: Foundation of the development centre in Bucharest
Start of construction of a front-end factory[4] in Kulim, Malaysia
Opening of a new research and development building in Villach
2006: Opening of the development centre in Bucharest
Disclosure of the front-end factory in Kulim, Malaysia
Expansion of the development centre in Graz
Integration of an expert support service centre in Klagenfurt
2008: Opening of the new building for research and innovation in Villach

[3] Micrometer, a human hair obtains a diameter of approximately 70 mm [14]
[4] Implies all production processes to fabricate a wafer.

In total, about 2.900 employees are operating in the established locations through Austria (until the end of the fiscal year 08/09 the workforce will be reduced to 2.600 workers due to the planned reduction of jobs) and approximately 1.000 workers of the entity are performing actions in the Research and Development (R&D) sector[14]

In the preceding financial year Infineon Technologies Austria AG generated 1, 2 billions of Euros by means of a production volume of 20 billions of chips. Due to their high requirement to innovate state-of-the-art technologies through Europe or at best through the world, it is not surprising that about one third of the Austrian employees are operating in the R&D segment and about 228 millions of Euros, this relates to 19% of the total revenues, are invested in the development of innovative approaches. Hence, Infineon Technologies Austria AG is one of the companies through Austria which undertakes that much vast capital expenditure in this department.

Solely Villach stands out for operating in the R&D as well as in the production segment. They focus their attention in the production segment towards the fabrication of automotive and industrial electronics. This sector will be operated with a production capacity of 97%. A variety of about 1,300 products will be fabricated in Austria. In the R&D department they put emphasis on power electronics for automotive and industrial devices as well as Integrated Circuits (ICs) for telecommunication[15]

All the remaining subsidiaries Graz, Linz and Klagenfurt are concentrating on their core competence in the R&D sector. In the fiscal year 2008 Infineon Technologies AG applied for about 200 new patents and at least one high potential from Austrians subsidiaries took part on this innovative technology. [14] However, Vienna is charged with the distribution of Austria and southeast European countries[15]

1.3 Infineon Technologies AG and the economic & financial crises

In the fiscal year 2008 the global economy slowed down substantially in comparison to the previous one. The economic recession was deepened through the crises in the financial markets and steadily increasing resource prices. Although there was high financial pressure, Infineon Technologies AG was able to increment their overall revenues about 6% compared to the previous year from 4,074 billion € in 2007 to 4,321 billion € in 2008. The Communication Solution sector denoted the greatest improvement in revenues due to their wireless communication applications.

	For the years ended September 30,			
	2007		2008	
	(€ in millions, except percentages)			
Automotive, Industrial & Multimarket . .	3,017	74%	2,963	69%
Communication Solutions	1,051	26	1,360	31
Other Operating Segments	219	5	100	2
Corporate and Eliminations	(213)	(5)	(102)	(2)
Total	4,074	100%	4,321	100%

Figure 2: Survey of revenue per segment[15]

Despite this positive improvement in the revenue performance, Infineon Technologies AG reorganized its cost structure in order to respond to the steady increasing risks in the market environment, the negative exchange rate and spread. Therefore, the cost reduction program “IFX10+” was invented in the third quarter of the fiscal year 2008. The reduction program contains the rejection of unprofitable product lines and the efficient utilization of the R&D sector. It also involves the retrenchment of manufacturing costs, a better coordination of the value chain, an optimization in the production processes but also a partly reduction in the workforce. Due to this program, Infineon Technologies AG realized restructuring charges in the amount of 188 million €[15]

The improvement in the overall revenues for the fiscal year 2008 is composed through the increase in the segment of Communication Solutions but also through an augmented sale of patents. However, increased negative affects of the foreign currency of 2% diminished the

overall performance of Infineon Technologies AG. Despite a positive result in the overall revenues, Infineon Technologies AG generates a negative performance of Earnings before interest and taxes (EBIT) resulting mainly through the restructuring charges of 188 million €.

However, which development experienced Infineon Technologies AG in the beginning of the current fiscal year? The revenues declined in the second quarter of the year about 10% that are 83 million € less compared with the previous quarter and a 29% per cent reduction of revenues according a quantity of 302 million € year-over-year. Through their cost reduction program “IFX10+” it was possible to generate about 60 million Euros savings, mainly acquired through reductions in the operating costs. Unfortunately also at the division workforce must diminish its quantity of employees. At the end of March solely 26.400 workers were employed compared to September where 29.100 workers operated at Infineon Technologies AG. Further measures like short time and uncompensated vacations should assist in terms of cost reduction.

For the upcoming quarter a forecasted improvement of 10% will be expected for Infineon Technologies AG in nearly all segments but a greater augmentation will be predicted in the segment of Wireless Solutions. Further, Infineon conducted an enormous reduction in inventory due to a couple of indications of demand stabilization, accordingly the level of inventory and production needs to be adjusted to the current customer deliveries and demands. Along with the benefits of increased sales and further reductions in costs, Infineon Technologies AG predicts a meaningful augmentation in the segment result[15]

In the face of the results of the past six months[5] and the forecasts of the forthcoming quarter, a reduction of the revenues of nearly 20% is expected for the fiscal year 2009.
CEO Peter Bauer valuates the current economic situation in conjunction with the actual positioning of Infineon Technologies AG as follows:

[5] Fiscal year October – September, Calendar year January - December

"In this challenging environment, we further need to have a tight grip on expenses. Strategically, Infineon is positioned well. We address the growth markets, which will gain even more in importance in the future with our three core topics energy efficiency, communications, and security. The recent customer endorsements from Bosch and Toyota in automotive power products and from Nokia in single-chip platforms for GSM/GPRS and EDGE illustrate the competitiveness of our product offering and our customers' faith in our operations."[15]

The announcement of the third quarterly report clarifies the upward tendency of Infineon Technologies AG in the current fiscal year. The expected recovery of Infineon´s revenues by 10 percent was achieved and could even be excelled by 3 percent. Compared to the previous quarter a significant improvement in its performance was accomplished, mainly driven through its intense production portfolio resulting in greater revenues, significant cost savings by the means of the cost retrenchment program IFX 10+ and through higher production utilization owing to an augmentation of demand. Hence, a positive performance in nearly all segments could be achieved.

The outlook for the remaining quarter is as well promising. Owing to the rise in demand, the production utilization will be accordingly adjusted. Together with a predicted continuing rise in revenues and a furthermore strict cost control, Infineon Technologies AG can assume further improvements in its segment result[15]

2. History and Development

2.1 History

The beginning of semiconductor manufacturing is dated back to 1948. This year AT&T Bell Lab built the first bipolar transistor. The inventors are called Walter Brattain, William Shockley and John Bardeen.[60] About ten years later the junction transistor technology (Planar) was invented. By then it was already possible to produce MOSFETs (metal oxide semiconductor field effect transistors) and transistors that contain integrated circuits, the so-called ICs. The mass production of devices started in the early 1960s because at that time it was possible to prepare silicon in its purest state. Further details concerning wafer fabrication and the basic raw materials are explained in the following chapters[35]

Some advantages of MOSFETs compared to bipolar transistors are:

- A MOSFET only uses 1/10 of the space of a bipolar transistor when their size of structure is even. Therefore, integrated circuits can be developed.
- The design of a device and the electrical features can stay the same even though the dimension of the component is proportionally diminished. The possibility to foresee the characteristics of the devices leads to the construction of roadmaps. The cost reduction by diminishing was enabled by the scalability.
- Low-level consumption circuits using CMOS-technology can be well adapted for ICs[60]

In 1971 the first microprocessor was launched, the mask of it was still handwork. Today's structures of processors cannot be tracked by human beings. Engineers only deal with structural considerations concerning the architecture and the procedures producing the chips. In 2003 more than 100 millions of transistors per head existed and one expected the number to increase up to one billion in 2007[59]

2.2. Moore's Law

The incredible fast growing of microprocessors is explained by Moore's Law. In 1965 Gordon Moore, who was one of the founders of INTEL, discovered the coherence of the capacity of microchips doubling every 18 months, while the production costs were cut into half. Between 1975 and 2001 the duplication really took place in the period of two years[35]

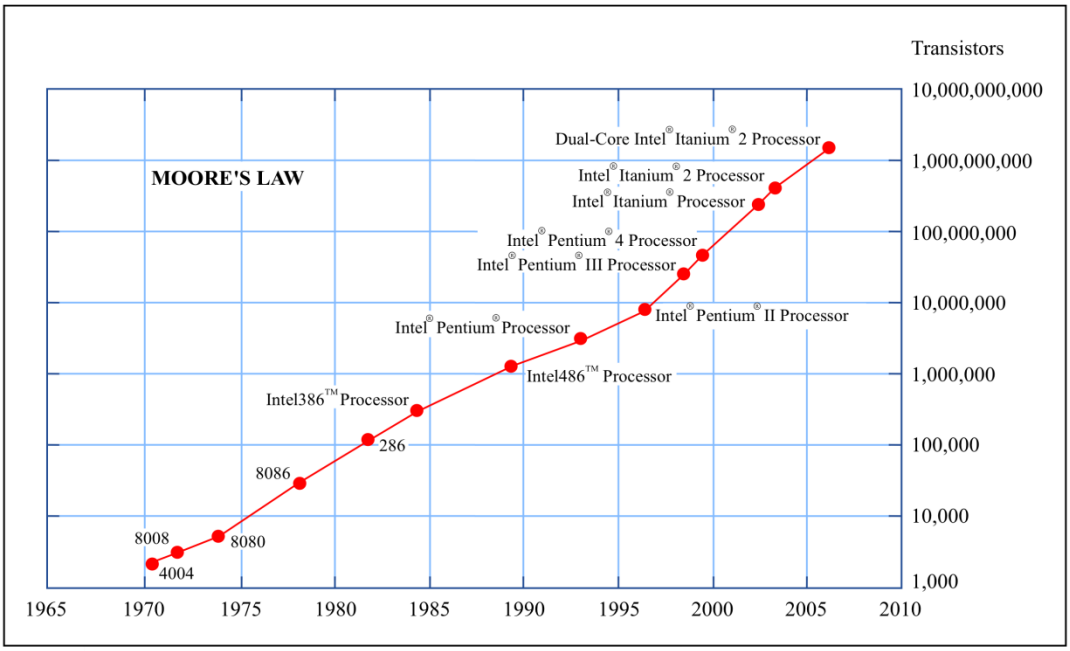

Image by MIT OpenCourseWare.

Figure 3: Moore's Law[43]

This law is not a physical law in the precise meaning; it is an observation which can be extrapolated to a real roadmap. Moore's law has mainly an economic relevance, which means: if the producers adhere by this law, they are able to maximize their profit because of knowing the exact market potential. Another advantage of Moore's finding is that planning becomes much easier for anybody in this business.

Since 1992 the so-called ITRS, the International Technology Roadmap for Semiconductors, exists. This Roadmap takes into account all the necessary technical characteristics for MOSFETs, which are necessary for the adherence of Moore's Law.

The cost reduction can be obtained by different methods. Diminishing represents one method where the size of the structures is minimized and more chips per wafer are produced. Another approach is to enlarge the wafers. The second procedure leads to high investments because new technical devices have to be installed.

Another important point concerning Moore's Law illustrates the Yield. This depicts the function of microchip sizes (the bigger the chip the more a defect is probable), structure sizes (smaller sizes are more susceptible) and shift number (every new shift causes new defect-possibilities).

2.3. Technological Development and Costs

The new technologies induce a cost reduction on one side, but also additional costs on the other side. New technologies require new equipment and new equipment is connected to high investments in semiconductor manufacturing. Expensive devices are needed for producing the constantly smaller structures of transistors. When a new semiconductor manufactory opens, approximately 75% of all costs are spent for new equipment. Therefore, the usage of the devices should be as efficient as possible in order to reduce costs for new investments[35]

An extraordinary expensive investment in this industry is the cleanroom, which will be explained in the next chapter.

3. Cleanroom

The most important point of cleanrooms is the fact that the level of environmental pollutants should be as low as possible. These pollutants consist of dust, airborne microbes, aerosol particles and chemical vapors.

“More accurately, a cleanroom has a controlled level of contamination that is specified by the number of particles per cubic meter at a specified particle size. To give perspective, the ambient air outside in a typical urban environment might contain as many as 35,000,000 particles per cubic meter, 0.5 µm and larger in diameter, corresponding to an ISO 9 cleanroom.”[11]

Class limits (maximum allowed particles):

ISO	FED STD 209	0.1 µm	0.2 µm	0.3µm	0.5µm	5.0µm
Class3	1	1,000/35			35/1	
Class4	10	10,000/345	75	30	352/10	0
Class5	100	100,000/3,450	750	300	3,520/100	0
Class6	1,000	1,000,000/ 34,500	N/A	N/A	35,200/ 1,000	7
Class7	10,000	345,000	N/A	N/A	352,000/ 10,000	70
Class8	100,000	3,450,000	N/A	N/A	3,520,000/ 100,000	700

ISO 14644-1 (per cubic meter)

Fed Std. 209 E USA (per cubic foot)

ISO standard requires results to be shown in cubic meters (1 cubic meter = 35.314 cubic feet)[12]

3.1. Two air flow principles in cleanrooms:

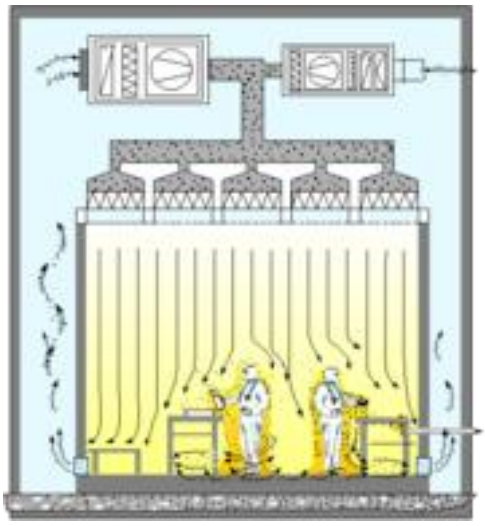

Figure 4: Airflow pattern for "Turbulent Cleanroom"

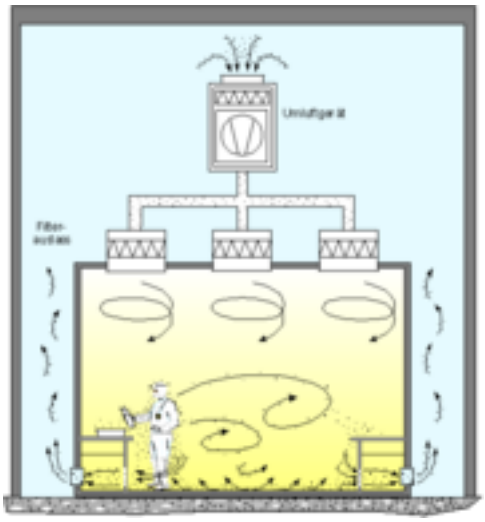

Figure 4- 1: "Laminar Flow Cleanroom"[11]

The disparity of these two cleanrooms lies in the structure, the airflow behavior and the overall specifications. In turbulent cleanrooms the air is changed 85 times per hour whereas in laminar flow cleanrooms this happens 420 times per hour. In laminar flow cleanrooms the range of overpressure, humidity and temperature is lower and so higher classes (standards) are achieved. It is clear that humidity as well as temperature, have to be kept constant all the time. Only very little deviations are allowed, otherwise higher failure rates would occur.

3.2. Cleanrooms in Semiconductor Manufacturing

All devices that are built on a chip are microscopically small. The components can be compared with tiny freeways where electricity either flows or does not flow. That is the reason why they are called semiconductors. Through these networks a microchip has the ability to accomplish the required functions and controls.

The small size of devices is the reason that some requirements have to be respected while working on wafers. The most important request is an artificial environment called cleanroom. Even the smallest pollution can cause a very hard damage of the device. For keeping the air as clean as

possible in these surroundings, it is sent through filtration plants. Both, the fresh and the circulating air are filtered to keep the contamination of the wafer as small as possible. One of the biggest problems in such cleanrooms is the worker. Each human being carries many particles. The first suggestion for avoiding the contamination by workers would be to automate the whole process, which is not possible because of the high complexity. Therefore, every employee must wear a so-called cleanroom-overall, a facemask and gloves.

The number of particles which are allowed in a production facility differs and the above-mentioned classifications are used as a guideline.

There exist two different types of contamination: chemical (fingerprints, sweat, and fat) and mechanical contamination (dust of machines, space components)[30]

After a short explanation of cleanrooms in semiconductor manufacturing the next chapter casts an eye at the basic materials used during the production process.

4. Basic Materials in Semiconductor Manufacturing

To produce an engine control out of a grain of sand is a very complex subject. Before starting the production the basic material, called monocrystalline silicon, is needed. This required chemical element does not exist in nature; but many chemical bonds with silicon are detected. The most common form is SiO_2 which is the basic for pure silicon[30]

4.1. Basic Elements used

4.1.1. Silicon

As mentioned before silicon Si is the most common basic building block of integrated circuits in semiconductor manufacturing. This element either conducts or insulates electricity; therefore silicon is a semiconductor at room temperature. For building p-n junctions on silicon dopant elements are needed. After adding these, electrical components and ICs can be built from these junctions.

“Silicon is obtained by heating silicon dioxide (SiO_2), or silica, with a reducing agent in a furnace. Silicon dioxide is the main component of ordinary sand.”[45]

4.1.2. Aluminum

Aluminum is used very often in semiconductor manufacturing. Metal lines are the conductors between the different components. It shows a good adherence to silicon, applied as a thin film. Wire bonding integrated circuits in ceramic packages is another usage of aluminum in semiconductor manufacturing.

4.1.3. Gold

“A good conductor of heat and electricity, it is also the most malleable and ductile of all metals.”

The main use of gold in semiconductor manufacturing is in the assembly, packaging process and wire bonding. Most commonly gold is used as wires for the connection of the IC to the leads of the package. The benefit of gold is that wires are resistant to wire breaking during the encapsulation process.

4.1.4. Silver

It is also a very good conductor of heat and electricity. Silver is used like gold in the assembly and packaging processes. It prevents plastic packages from chemical degradation.

4.1.5. Copper

Copper is a good conductor of heat, less efficient than silver but better than gold. It is used in assembly. Most of the lead frames for plastic packages are made out of copper. In semiconductor devices copper is applied as metal lines[45]

The following chapter deals with different procedures in semiconductor manufacturing, namely job shop, flow shop and the reentrant flow.

5. Scheduling Procedures in Semiconductor Manufacturing

In semiconductor manufacturing many different approaches are used to create a machine plan for the next periods. Three of the scheduling procedures will be explained, first theoretically and in a special case in one of the following chapters.

Scheduling deals with the chronology allocating of jobs to machines and vice versa with respect to different goals and restrictions. At every time, one machine can only work on a single job and each job can be processed by a single machine[51]

5.1. Flow Shop

Only once each job has to be processed precisely on each machine and the machine order is fixed. Every job has to be processed in the same order.

5.1.1. Permutation Flow Shop

A permutation flow shop includes another important rule. A restriction on overtaking must be considered. The job order on the machines remains the same for all jobs. A convenient permutation of the jobs must be determined.

5.2. Job Shop

Each job of such a problem has a given order. The machine orders of different jobs vary from each other but they are fixed in advance. Most of the time, each job must be processed exactly once on each machine[52] In semiconductor manufacturing job shops are more likely to be found because different products have different machine orders.

5.3. Re-entrant Flow

Another difficult point in semiconductor manufacturing relate to the fact that wafers have to be processed more than once on each machine.

“Most manufacturing systems do not have the same work piece revisiting the same equipment except for rework. In semiconductor manufacturing, recirculation is the essence of the system. Semiconductor devices are layered structures in which each layer is produced in essentially same manner, with some variations to deal with differing materials introduced, or accuracy required.”

Some of the lots will return to the same machine in a cyclic manner which causes unusual outcomes on one hand and provides an opportunity for production control on the other hand. The degree and impact of the re-entrant flow in semiconductor manufacturing is much higher than in any other industry[26]

The following chapter shows the general structure of a so-called wafer fab.

6. Structure of a Wafer Fab

In so-called semiconductor fabs wafers are produced. On each semiconductor wafer there are either logic chips (integrated circuits) or memory chips (DRAMs) located. For the production the difference is very important, because the variety of products affects the logistical requirements.

Because of the hundreds of different work plans, mass production cannot yet be realized in semiconductor manufacturing. It is very important to place as many types of equipment as possible in a small room in wafer fabs to obtain the work environment. Therefore, shop fabrication is the leading layout for wafer fabs. That means that the equipment is the influencing variable for the fab layout. Machines are pooled together in modules, with regard to the processes they are implementing. This production flow is also called farm-layout. In semiconductor manufacturing machines, that are grouped together, because they are exactly the same, are called work centers[35]

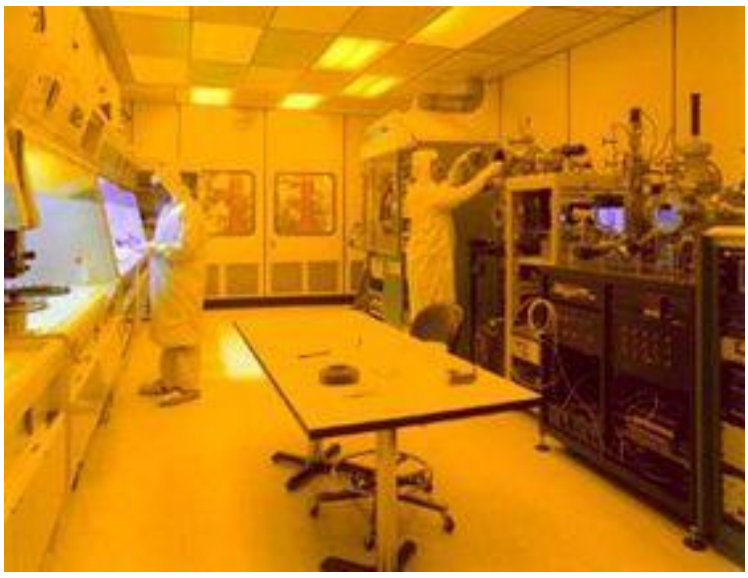

Figure 5: NASA's Glenn research centre cleanroom[62]

Since the workflow does not play any role in layout planning, lots might travel long distances when changing the equipment. As mentioned before, wafers are produced using the so-called re-entrant flow and so the way of transportation might be very long and this may cause a big loss of time.

Operators play a significant role in semiconductor manufacturing. The automation in the logic chip field stands on a low level. It is not possible to install equipment that accomplishes to produce all the different product variations. Therefore, the operator has to transport the lots, load in and unload the machinery, correct disruptions, adjust parameters and load in new process programs. In order to know when an operator has to perform which step of the production plan, a so-called lot traveler is added to each lot. On this protocol, each step is recorded and all parameters are listed. As the semiconductor manufacturing takes place in a paperless fab, communication between the operators is not that necessary. Paperless fab means that all the information about a lot has already been sent to the machines and is available in the computer system. Even the route of transportation is fixed. One reason for inefficient performance can be the fact that the process of manufacturing is preset. If anything unforeseen happens, it is very difficult to react and stay on schedule[30]

In order to recognize the complexity and diversity of the entire semiconductor manufacturing process in a wafer fab, the coming section should provide a deeper insight.

7. The Manufacturing Process – Overview

For the production of semiconductors, the raw materials need to undergo many hundreds of different process steps operated through machines with state-to-art technologies which are presupposing a high level of accuracy. The manufacturing process is furthermore characterized through the necessity of returning to the same machines for a number of times at different stages of their fabrication[2] Due to this complexity and the throughput of nearly thousands of production centers, the underlying progress is considered as one of the most difficult manufacturing processes in today's life[5]

In general, there is a coarse separation of the whole production progress: the front-end and the back-end part. The first fraction called front-end is responsible for the complete transformation of the raw material to the semiconductor substrate until the wafer probe where the selection of detective wafers occurs. Especially the wafer fabrication is the most complex and therefore also most cost-intensive part of the whole manufacturing process. Whereat the back-end production consists of assembling the completed semiconductors and additionally executes a final testing[27][35]

Since the manufacturing process of semiconductors is subjected to great complexity a successive production would make it rather impossible to generate a maximum throughput of finished microchips. Hence, depending on the diameter of a silicon wafer nearly thousands of identical circuit patterns or microchips, established of 25 different layers can be arranged on only one single semiconductor substrate[27]

Within a semiconductor plant, also known as a fab, the transportation of wafers is organized in lots. About 25 wafers (sometimes can also have a size of 50 wafers) are merged to one lot and several lots are integrated in one box while they are transported and traversing the route of production. For several production steps like processes in furnaces, the transportation and the processing can be abstracted in greater units like so called

batches. This supports the intra-plant logistic chain but also abbreviates the total throughput flow. As aforementioned, wafers are composed of 25 several layers with distinctive layer structures. This variety of layers is responsible for the fabrication of a circuit, which is the essential part of a finished semiconductor. Thus to the differences of the raw materials (silicon, gallium, gallium arsenide) layers may exhibit distinctions in their electrical conductivity and hence in their quality. Due to the production of the different layer surfaces a plurality of machines that possesses high modern technologies is essential. During the entire manufacturing process, a number of different products is fabricated and thus every product requires a certain production plan that dedicates a distinct route through the fab. According to the diversity of the product mix, it needs to be taken into account that set-uptimes and costs are occurring. The sequence of the production steps diversifies from product to product[35] The figure stated below should give a generic survey of the entire semiconductor manufacturing process.

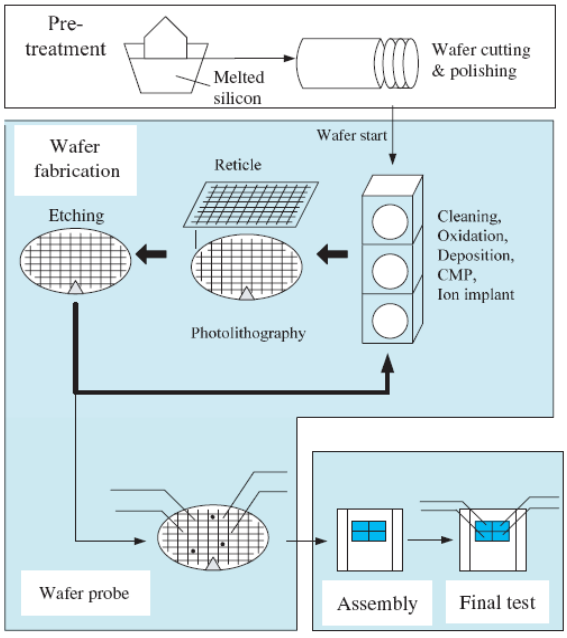

Figure 6: generic survey of semiconductor manufacturing process[27]

The whole production process consists of the following major procedures:

- Oxidation
- Photo Lithography
- Etching
- Doping/Ion implantation
- Anneal (heat treatment)[2]

In order to initiate the entire production process, raw silicon wafers are fabricated from silicon ingots. The ingot is first shaped and afterwards trenched into very thin wafers[45] In the first processing step, an oxide layer will be placed upon the raw silicon wafers surface through the exposure in a furnace. This first layer operates like a certain pattern thus forces the further treatment of the underlying material. Furthermore, a photosensitive film (photo-resist layer) will be deposited on the oxide layer and subsequent this layer is exposed with ultraviolet light through a resolution patterned mask. For these parts on the layer, which were exposed to the ultraviolet light, are now undergoing a transformation and these areas can be processed with a chemical solvent, a so called developer. After this procedure there exist only the unexposed parts of the photo resist. An etching process enables the translation of the generated pattern to the oxide surface beneath. Further processing steps like doping enable to diversify the remaining oxide layer in its structure and characteristics. Through the displacement of the pattern the application of the first layer is completed[35]

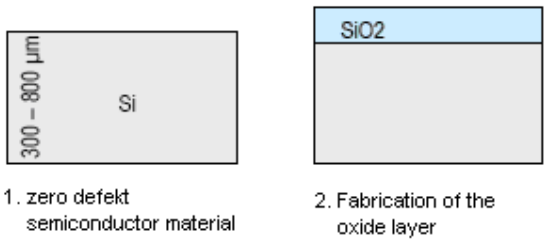

Figure 7: Fabrication of a single layer[35]

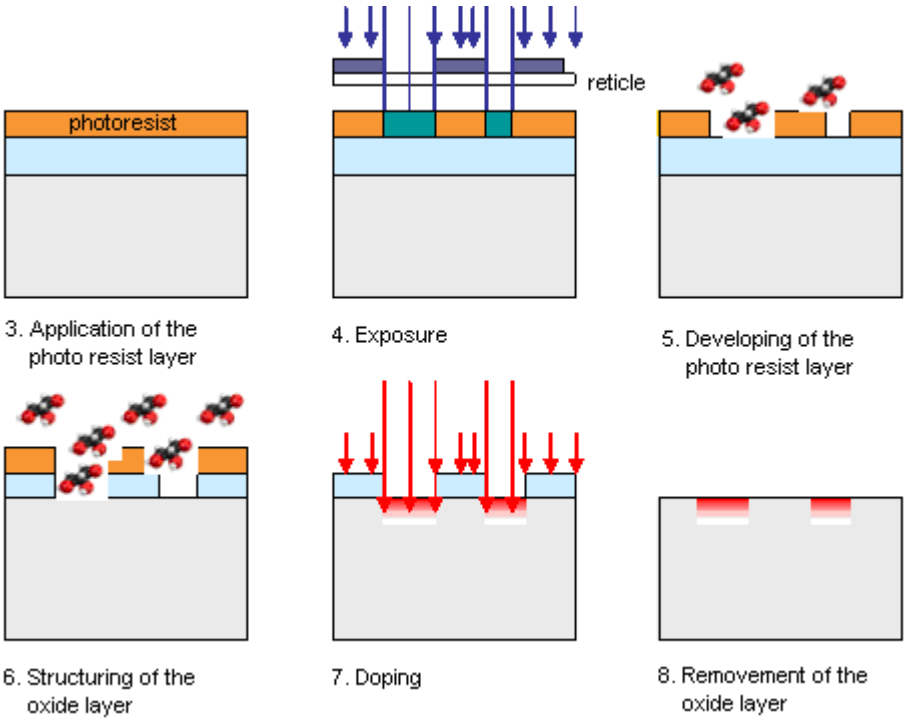

Figure 7- 1: Fabrication of a single layer[35]

7.1 Processing steps in detail

The previous chapter displayed the complexity of the entire manufacturing process in general as well as the application of solely one layer on a silicon raw wafer. As aforementioned, the fabrication of merely one layer is necessitating a number of distinct procedures. However, for the production of a complete integrated circuit a diversity of special layers is required. Thus the following section should give a deeper insight in the procedures a raw wafer traverses during its transformation to a semiconductor integrated circuit.

7.1.1 Oxidation

As mentioned above, a multitude of diverse layers is fundamental to generate an integrated circuit. The great number of different layers is the essential part for the further generation of distinct structures and patterns on a wafer through the photolithography. The different fabricated layers obtain certain functions, for instance for isolation, ion implant, planarization, diffusion, as an alignable mark or for protection.

Prior cleaning the raw wafer and thus the removal of eventual contamination needs to be achieved. In order to gain a disposition for the raw wafer, a procedure called oxidation is performed. This chemical reaction occurs through the compound of the raw material silicon Si and the exposure to oxide O_2. This generated silicon dioxide SiO_2 enables a very fine and evenly application of layers and is used furthermore to cover the wafer with the initial layer. Through this layer, the wafer substrate obtains certain resilience and can only be processed with hydrofluoric acid. Further substances like water or other acids cannot corrode the wafer's surface.

In course of this thermal oxidation the silicon raw wafer is transferred into a furnace which is heated with approximately 1000 °C. Subsequent gaseous oxide is admitted and owing to the reaction of the silicon raw wafer and the gaseous oxide silicon dioxide is generated upon the surface[54]

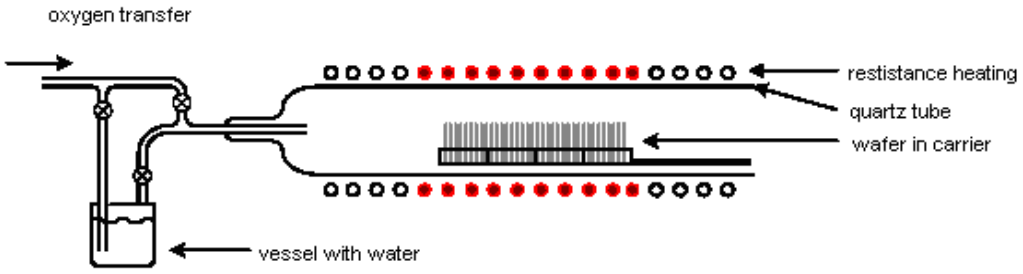

Figure 8: Demonstration of a furnace for oxidation[54]

The thermal oxidation differentiates between

- Dry oxidation
- Humid oxidation
    - Wet oxidation
    - H_2O_2 burning

Dry oxidation only takes place in a pure oxide atmosphere, whereas wet oxidation also disposes water in the element of water vapor. Dry oxidation is basically characterized through a rather steady growing oxide but with a great tightness. However, wet oxidation displays high growth of oxide already at low rate of temperature but with minor quality. H_2O_2 burning

utilizes besides pure oxide also pure hydrogen. Equal to the wet oxidation, H_2O_2 burning characterizes through its rapid growth of oxide layers with hardly any contamination. At a rather low level of temperature as well as fine and thick layers are fabricated.

Initially oxide reacts with the silicon surface of the wafer and thus constitutes a silicon dioxide layer. In order to obtain a reaction of the oxide with the silicon, diffusion occurs. The oxide grows with approximately 50% into the silicon dioxide wafer substrate[54]

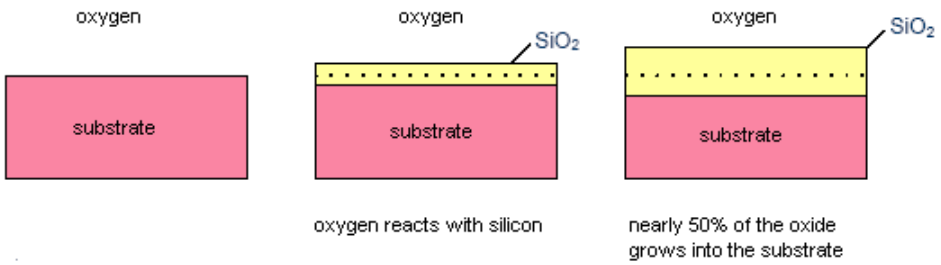

Figure 9: growth behavior of oxide on silicon[54]

7.1.2 Photolithography

The photolithography is seen as the operation that necessitates the greatest accuracy and is also subjected to an enormous complexity. Due to the photolithography, the wafer obtains its certain structure and pattern. Based on the prior generated oxide layer, a radiation-sensitive liquid is deposited onto the wafer, a so-called photoresist. Subsequently a very high resolution patterned mask is utilized in order to fabricate a certain pattern to the deposited photoresist. Furthermore, this wafer deposited now with the oxide layer and the photoresist, is exposed to ultraviolet light. Through a chemical solvent the exposed parts are developed. The remaining unexposed as well as the exposed fragments is forming now the fundamental pattern of the circuit achieved through the extern reticle. Due to the etching operation, the reticle is translated to the underlying silicon dioxide layer[53]

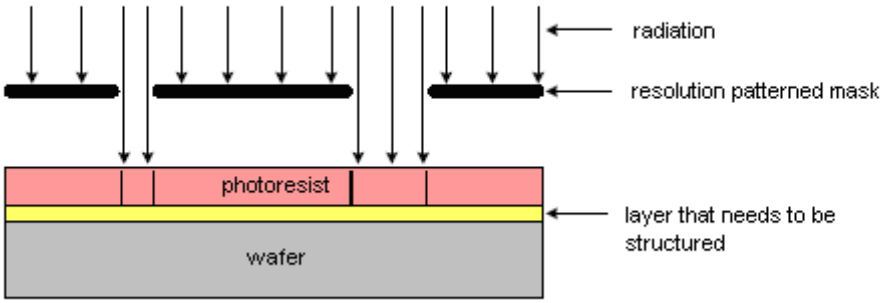

Figure 10: Transmission of the pattern on the oxide layer[54]

Owing to the requirement of a plurality of distinct layers also a magnitude of diverse reticles is necessary in order to obtain certain structures on the integrated circuit. As a guideline for the enormous amount of various masks it can be said that about 400 different types of integrated circuits require approximately 40.000 different reticles. Due to the fact that an integrated circuit contains of a magnitude of diverse layers, the photolithography as well as the oxidation operation need to be traversed approximately 30 times during the entire manufacturing process[35]

Owing to this essential fact, it is crucial that this operation is not starving a failure. Otherwise the total fabrication will be down and this will cause an enormous financial loss to the company.

7.1.3 Etching

Etching is essential either for the removal of an entire layer or for the transmission of a structured photoresist layer to the beneath surface through the operation of photolithography. The etching operation is differentiated in wet-chemical and dry etching. Furthermore, there is a distinction between isotropic and anisotropic etching procedures. An isotropic operation is mainly characterized through its freedom of etching. It may occur as well as horizontal or vertical but also can diversify the layer in its size, whereas an anisotropic procedure is restricted to a solely upright direction. Therefore, undercuts are unfeasible and etching the underlying surface occurs along the previous pattern[48]

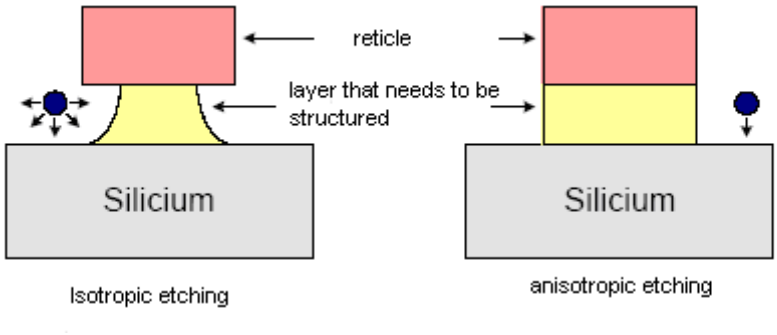

Figure 11: isotropic and anisotropic etching[48]

A further crucial factor in etching is the selectivity. Selectivity in this context is stated as the proportion by the removal of the layer that need to be structured and therefore etched and the removal of other layers. If the selectivity is quoted as 2:1 that implies that the oxide layer is etched twice as fast as the underlying layer. In order to spare the subjacent layer from further etching, the layer, which needs to be diversified, exhibits a great selectivity[48]

As aforementioned, there is the distinction among wet-chemical and dry etching. Wet-chemical etching is characterized through the conversion of the fixed layer into a liquid by the use of chemical solvents. This approach obtains a high selectivity thus to the fact that the chemical solvent can be deposited very precise onto the removable layer. For many solvents the selectivity accounts the proportion of 100:1. In this approach the abrasive occurs mainly isotropic while especially for extremely fine layers an anisotropic etching is recommendable. The method of dry etching is advantageous for exactly such situations. Besides a sufficient selectivity, this approach also exhibits the option of isotropic and anisotropic operations[65]

In order to fabricate integrated circuits it is furthermore essential to satisfy the conductivity requirements. The method of doping enables to enhance the conductivity of a semiconductor[30]

7.1.4 Doping

Doping is known as the implementation of external atoms into a chip in order to enhance its electrical properties. Through this insertion, it is possible to manipulate certain parts of the wafers conductivity. The selected impurities are introduced into the monocrystalline grid and through this procedure; the chips conductivity can be enhanced by a factor of 10^6. It will be distinguished among the n-doping and the p-doping. The amount of external electrons is essential for the type of doping. The two main elements in order to perform doping are Boron and Phosphor. Boron consists of 3 valence electrons and is therefore responsible for the p-type doping, whereas Phosphor inhabits 5 electrons and thus accountable for the n-type doping[22]

7.1.4.1 n-type Doping

As the silicon wafer possesses solely 4 valence electrons, the fifth electron provides as a charge carrier and can move freely in the lattice. This remaining valence electron needs significant less energy in order to be moved into the conduction band compared to the other electrons that are accountable for the intrinsic conductivity. The doping element Phosphor receives a positive charge through the emission of negative charge carriers and is permanent integrated into the grid, only the electrons are roaming freely. Doping is denoted as n-type doping owing to the generation of conductivity through negative free moving electrons. They also are named the majority charge carriers because of their surplus of free electrons[22]

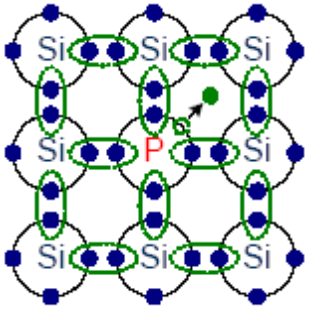

Figure 12: n-doping[22]

7.1.4.2 p-type doping

Exactly the counterpart occurs at the p-type doping. There are no valence electrons that can roam freely, contrariwise the doping element Boron consists solely of 3 valence electrons and therefore leaves a gap in the valence band of the lattice. Due to this now generated gap, the electrons can roam in the valence band. But the movement of the gaps occurs in the reverse direction to the electron flow. The doping element Boron is charged negatively through the insertion of an electron. As seen before, the doping element is bounded into the lattice and only the positive charge is roaming. This doping is named p-type doping due to the generation of conductivity through positively charged gaps. On the contrary to the n-type doping, freely moving gaps are now available in a greater amount and thus are called the majority charge carriers. The denomination of n-type or p-type doping conforms to the majority charge carriers[22]

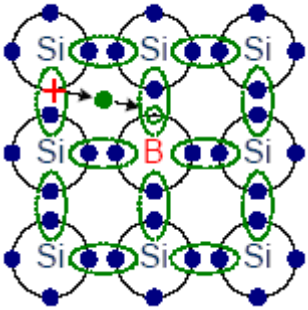

Figure 13: p-doping[22]

7.1.5 Ion implantation and annealing

The most important method as to doping is the ion implantation. This process implies the direct bombardment of the underlying wafer with high energy ions. Due to the advantage of high precision for depositing dopant atoms onto the wafer substrate, ion implantation replaced the thermal diffusion. Ion implantation as a method of doping is accountable for the diversion of the type and level of conductivity of semiconductor materials. During the process of ion implantation, atoms of impurity are vaporized and accelerated onto the silicon substrate. The atoms are now inserting the crystal lattice and are forfeiting their energy due to collidings with the

existing silicon atoms until the impurity atoms are located at some place in the crystal lattice. Through heat treatment, the lattice dysfunctions, resulting through the collidings of atoms, are mended and rebuilt. The collidings of atoms and thus the generated disturbances are accountable for the alteration of the electronic characteristics of the silicon wafer. Through the procedure of annealing, the lattice defects are repaired. This process, however, is also essential for the positioning of dopant atoms in order to guarantee the electrical activity[45]

7.1.6 Cleaning treatments and testing

The entire manufacturing process necessitates a high level of accuracy and purity. Already very small amounts of particle or other forms of contamination can destroy the whole wafer production. In order to maintain purity of the wafers during the throughput of the production chain, several cleaning procedures are integrated so as to retain a high qualitative final product. Before the completed circuits are assembled and delivered, the finished semiconductors traverse a number of testing instances owing to separate defective semiconductors and inspect once more the qualitative requirements[30]

All displayed procedures are traversed a number of times during the entire production process. As it is one of the most complex processes in today's life, it requires a high level of coordination and also awareness of potentially appearing issues in order to generate a trouble free production chain.

8. Potentially occurring issues

The semiconductor process itself is subjected to tremendous complexity but a few crucial factors need to be recognized as well in order to generate an optimal production and capacity plan for a semiconductor fab.

Initially, it is important to become aware of the fact that the semiconductor manufacturing underlies quickly altering technical and social requirements. Semiconductor fabs need to cope with the demands of their costumers towards high qualitative, always innovative products but also to enhance their delivery times. As customers are mainly demanding more and more customized products, causing a great diversity of the product mix, this also implies an enormous challenge to plan a feasible production flow in a wafer fab. Additionally, technologies obsolesce rapidly and in order to keep up with the fast changes it is essential to develop continuously state-to-the art technologies[53] Nowadays, technologies and machines are facing more and more shortened life cycles, as an innovative technology might be out of date within a few months.

Furthermore, the wafer fab faces also the issue of the capital-intensive equipment and its lead-time. In order to detect breakdowns or disruptions of machines early enough, a good working shop floor control system needs to be implemented in every fab. Otherwise the entire production might be subjected to a shutdown and that implies an enormous financial loss due to the fact that the new tools underlie a lead-time of several months ahead. If an existing fab wants to expand its capacity, this process will take about nine months, whereas arranging a cleanroom would take nearly a year. For instance, a new fab is worth at least a billion and its incorporation time is about one year. Due to this vast capital amounts inside a semiconductor shop floor, the fab planners need to decide whether to expand the capacity based on the fact that forecasted demands are rather imprecisely or to keep machines with older technology in the production flow[27]

Another important factor in association with the semiconductor fabrication is the fact that the demand underlies a stochastic behavior. So the

requested quantity for the coming periods cannot be predicted exactly and therefore fab planners are subjected to volatile demand behavior and need to make decisions to whether expand their capacity through the utilization of old machines or through an order of innovative machines which necessitating several month to be delivered[27]

As production plans are subjected to the volatility of demand and capacity but as well as disruptions or starvations of machines, the underlying production and capacity planning needs to be adjusted more often in order not to suffer a breakdown of the entire production flow. Therefore, a rolling production planning seems to be an improved method to overcome eventual breakdowns and changes in demands. A rolling production plan implies that after a certain interval of time the already occurred planning for the further months will be recalculated and updated due to the current challenges in the shop floor.

All these possible issues refer to the parameter called uncertainty which is contemplated in a stochastic model. Therefore, the following section of this work is dealing with the specification of a deterministic, whereat all factors are conversant and a stochastic model. Furthermore, the work attempts to display a generic deterministic mathematical formulation for our manufacturing issue and give adjustments for a stochastic model.

9. Deterministic and Stochastic Model Formulation

Due to the general description of the highly complex semiconductor manufacturing process, the objective is now to optimize the sequence of the processing steps by the means of two distinct subproblems: on the one hand the challenge of the material flow and on the other hand the product mix issue. Foremost, the thesis pays attention towards the generic formulation of deterministic and stochastic models. Further amplifications respectively the two subproblems will emphasize on the formulations of solely deterministic models.

The generic description of the deterministic model will be the first to be shown. Further, the stochastic model description is explained.

9.1 Deterministic Basic Model

In general, a multi-period deterministic model describes a problem in which all of the different parameters are known in advance. For each period the profit, demand, supply, capacity, processing times, set-up times, wafers and the number of equal machines is given. In this case no uncertainty exists and therefore this kind of model is called deterministic.

Another characteristic of this model shows the fact that in different periods the amount of each parameter is different. That emphasizes the circumstance that the basic model can be defined as a dynamic deterministic multi-period model.

In other words:

In general, as to the deterministic model formulation, it can further distinguish between a static and a dynamic model. In a static deterministic model all the parameters are assumed as conversant and are beyond that constant for all following periods. For instance, the demand forecasts are equal for the forthcoming periods, whereas a dynamic model indicates that the demand varies within defined periods. However, the demand is noted

but varies from period to period. The dynamic deterministic model is far easier to solve and to implement as the dynamic stochastic model, whereas the demand data is established solely through a probability distribution[50]

9.2 Stochastic Basic Model

In return to the deterministic model, where all necessary information about the demand forecasts are known in advance, the stochastic production model is mainly distinguished through its integration of current challenges that might occur in the operative level and its undetermined information level[63] All the unpredictable events along the production line make it quite hard to formulate a model which displays all uncertainties and dynamics of today's corporate developments[37] Despite these requirements of the current technological century, the model although needs to be resolvable in an economically justifiable computation time but which additionally needs to provide a feasible performance.

The influencing value, uncertainty, which is subjected to the stochastic models, possesses miscellaneous causes:

- Uncertainty of the demand (Output)
- Variable points of time for the inventory reappointment (Input)
- Delivered amount diverges from the initial order
- Potential failures in inventory maintenance[52]

Furthermore, stochastic models which are displaying the complexity of the real world, are underlying disruptions which can be distinguished in primary and secondary disruptions, whereat primary dysfunctions come along with modifications of information and impact the system externally. In return, secondary disruptions are resulting through primary dysfunctions and can be divided respectively to their expansion in vertical and horizontal secondary disruptions[37]

Particular in the semiconductor manufacturing, uncertainty may cause severe issues in the production level and as well as in the corporate profit. Therefore, this great number of unforeseen challenges need to be observed in order to formulate a validate model and additionally gain an optimization in the operative output. Besides optimal material handling and transfer, also potential down times of machine shortfalls need to be taken

into account. Especially in this manufacturing process, the flexibility towards the tremendous technological changes and thus the new requirements but also the factor cost respectively to the enormous capital intensive equipment need to be considered in order to formulate a model. So as to implement a solvable model, it is essential to pay attention towards solely one severe issue otherwise no acceptable solution can be found due to too comprehensive objectives and constraints. Therefore, the work concentrates solely on the generic formulation of the stochastic model. Further amplifications towards this basic model are not discussed in depth.

9.2.1 Methods for considering uncertainty

One approach for multi period production models in order to handle uncertainty considers a schedule which is restricted to a defined time horizon, whereas consecutive adjustments are undertaken so as to update future planning data with current information. Through these continuous rolling forecasts a retrenchment in the production planning dynamics is potential. Particular in the semiconductor manufacturing, as aforementioned, the technological alteration occurs in always abbreviated periods and results at all times in shortened product cycles, thus it is unavoidable to successively adjust the production plan in order to keep up with the new technological and economical requirements.

Further alternatives to delimit the probability of uncertainty or protection measures to counteract the expansion of dysfunctions in the planning system are essential to prevent ostensible primary disruptions which furthermore might expand to secondary ones. In order to constraint these dysfunctions an advanced procedure for predictions need to be implemented[37]

Another opportunity to prevent uncertainty is the utilization of safety stocks. The amount of these stocks will be determined through the usage of a probability distribution of the demand respectively the allocation of the prediction failure. In the course of the determination of the quantity for the

safety stock should also be contemplated the holding costs arising by reason of the storage of the additional stock. Therefore, it is recommendable to compare the supplementary costs for the heightening of the stock to potentially accruing excess holding charges.

Additionally to the safety stocks which are preventing quantitative issues, safety times should also be recognized. A defined interval is selected such that the customer order is realized exactly this period prior to the final due date[37]

The further section deals with the presentation of the two subproblems which are established as deterministic models.

10. Statement of task

The entire manufacturing process of semiconductors consists, as aforementioned, of high complexity and due to the influencing factor of uncertainty through the dynamic economy as well the high probability of disruptions in the great usage of machines in the shop floor, it rather complicate to formulate a production model which captures all potential scenarios. Therefore, the work extracts two separate issues form the whole production process: the material flow issue and the product mix subproblem. Foremost the thesis continues with a generic description of the two problems, afterwards the mathematical formulation will follow and the main task concentrates on the implementation, the improvement of the existing model through exact methods and heuristics and ultimately the evaluation of the received results.

11. Product Mix Subproblem

Since the end of the 50s, product-mix planning models have been investigated. The aim of such problems is to maximize profit subject to constraints on demand and production resources. In general, an aggregate planning module might be used to adjust the mix in accordance with available capacity in a semiconductor manufacturing company. Some information can also be obtained by such a model, for example:

- Demand feasibility estimations: These estimations are made in advance in order to determine whether a set of demands for product-mix for a given period is capacity-feasible.
- Bottleneck recognition: Some constraints have to limit capacity in a given period to avoid bottlenecks.
- Product-mix adjustment: Upper bounds should be met in general. If this is not possible for capacity reasons, the authorized manager reduces the volume of certain product types in the product-mix and by that tries to maximize the profit by manufacturing only highly profitable products.

The production quantities of all kinds of items in the product-mix could be determined by the appropriate aggregate plan. This applies from final products down to lowest-level components. Demand for final products is composed by the demands of the intermediate and primary products, which must be used in order to generate the certain type of final wafer. The Bill of Material (BOM) describes the direct relationship between end items and lower level items. The BOM lists of all the subassemblies, parts, raw materials and purchased components that reveal the final product[38]

As semiconductor manufacturing companies produce wafer make-to-order and assembly lines are used to produce the whole variety of products, the parts required and the operations that are performed differ from unit to unit and from product type to product type. This variability causes unit-to-unit differences in work-station task times, like processing and set-up times.

This circumstance has to be taken into consideration when trying to solve a product-mix problem with respect to wafer fabs[8]

As mentioned before, especially in semiconductor fabs, several dozens or even hundreds of different products with even more derivates are produced at the same time. In product-mix models one has to expect constant changes due to incoming orders that differ from each other. If the company policy is producing make-to-stock, the product-mix strongly depends on the current amount of received orders. Some other problems represent the continuously advancing process technologies, the reducing start rates of old technologies and the increase in start rates of new technologies. All of them lead to changes in the product-mix.

Many different kinds of products use the same machines for their production processes; a strong interaction exists among them.

“Therefore product mix has considerable impact on throughput, cycle times and hence on the capability of meeting due dates, which is considered to be one of the most important metrics to measure fab performance.”[23]

In production planning some really essential questions arise when considering product-mix problems:

- What short term effects an increase in the number of wafer starts of a specific product will produce?
- Is it possible to tolerate the resulting cycle times?
- Is it possible to handle the increase in work in process (WIP)?
- “Is the fab able to recover after a production surge, i.e., do the cycle times return to a “normal” level?”[23]

Product-mix problems do not only involve strategic planning problems but also contain two issues at the operational level. Cost accounting of capacity at the process step level and the optimization of product-mix levels affect

the operational level. The cost accounting side tries to accurately estimate the manufacturing costs of each product type, whereas the optimization aim is to maximize the efficiency of capacity allocation across products. To manufacture each type of product a certain amount of resources is required. It is a fact that resources are limited and profit varies from product type to product type, the optimal product-mix model can be seen as a combinatorial optimization problem. Also, the theory of constraints can be used to explain a product-mix model.

“However, it has been shown by numerical examples that both methods, although differing in their implementation procedure and rigorousness, are conceptually equivalent and could lead to the same solutions.”

A semiconductor fab, as mentioned above, comprises hundreds of machines and automated material handling systems and exhibits complex queuing network behaviors. Absolutely interrelated are the flow times, machine utilizations and throughput. Some unique characteristics of the semiconductor industry are the following:

- Long process routing
- More than one month present the average wafer lot flow time
- Encountering of significant and uncertain queuing delays
- Bottleneck shifting from one group of resource to another
- Complex performance trade-off in wafer plants[10]

“Engineers and managers on the shop floor have access to real-time information that could be used in order to dynamically enhance operation efficiency and productivity. That is to say, product mix planning as a decision task should not be separated from shop floor management. This perspective will affect how product mix planning should be done and will be elaborated on in later sections.”[10]

In this specific case processing and set-up times, as well as the total profit and costs are considered to form a product-mix model. The coherence of

products that are split to jobs and their allocation to machines is shown in the following graph:

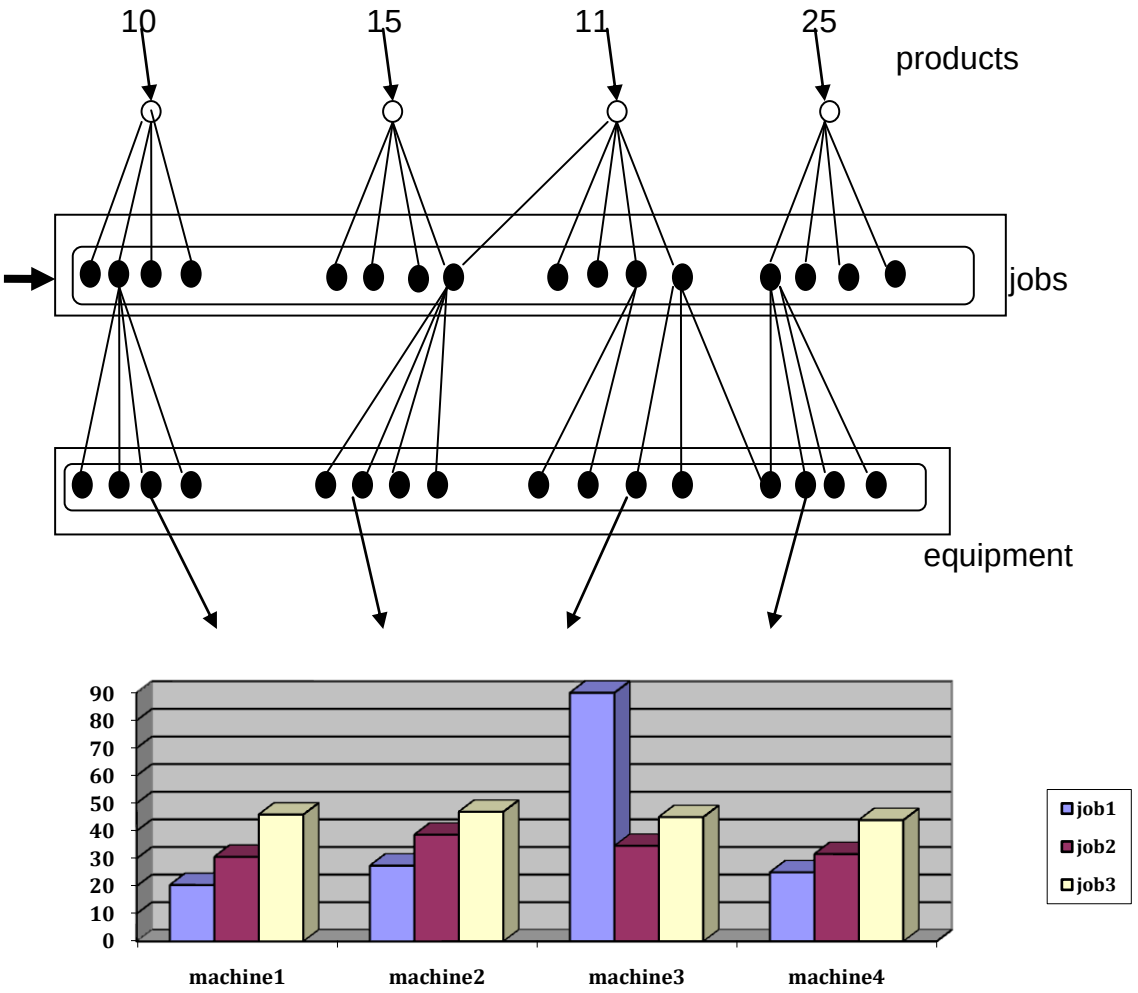

Figure 14: Product Mix Issue - outline

This graph shows different types of products which are assigned to different jobs. They are then split up in order to be processed on different kinds of machines to get the various product types that are ordered for the planning period. The last table shows the utilization of a certain number of machines. Each job and so each product type posses a specific production time and it varies from product to product.

The special product-mix model is explained in the following subchapter considering profit, costs, demand and capacity. Additionally, the product-mix model takes processing and set-up times into account for completing a full basic model.

11.1. Mathematical Formulation

As described beforehand the product mix problem is now shown in the mathematical way. The parameters almost remain the same. Profit is needed again as the impact of the model is to maximize profit.

Parameters:

U_m…maximum workload of machine m

S_i…selling price of product i

d_i…demand of product i

d_i^+…upper bound for produced demand of product i

$a_{j,i}$ …number of jobs j needed to produce product i

$t_{j,m}$…processing time of job j on machine m in days

N_m..number of tools m in a tool group

Indices:

iЄI…products

jЄJ…jobs

mЄM…machines

Variable:

x_i…production quantity of product i

$y_{j,m}$…amount of jobs j on machine m

Auxiliary Variable:

Z_j…sum of all steps needed of every job j

The aim of the objective function is to maximize the profit. The target is shown by multiplying the selling price times the products in each period.

Maximize $\sum_{i \in I} (x_i * S_i)$ (1)

Formula (2) expresses the needed steps of each machine m to manufacture job j. This amount of processing steps must not exceed the maximum capacity of available steps.

s.t. $$\sum_{m \in M} y_{j,m} \geq Z j \qquad \forall j \in J \qquad (2)$$

Formula (3) exhibits on the left hand side the used amount of steps times the quantity of product i that should not exceed the right hand side. This side displays a predetermined number of steps needed to perform job j.

$$\sum_{i \in I} (a_{j,i} * x_i) \leq Z_j \qquad \forall j \in J \qquad (3)$$

Formulas (2) and (3) can be combined to facilitate the implementation in Xpress.

$$\sum_{m \in M} y_{j,m} \geq \sum_{i \in I} (a_{j,i} * x_i) \qquad \forall j \in J \qquad (2) \cup (3)$$

The next formula represents the capacity constraint. The amount of jobs on each machine multiplied by their processing times must be less or equal the maximum workload of each machine available. If this constraint is injured bottlenecks might arise.

$$\sum_{j \in J(m)} (y_{j,m} * t_{j,m}) \leq U_m * N_m \qquad \forall m \in M \qquad (4)$$

Once again demand has to be satisfied. Constraint (5) shows that the produced amount each day is limited by the given customer demand and the upper bound of customer demand. The upper bound restricts the maximum amount that is allowed to be produced each day.

$$d_i \leq x_i \leq d_i^+ \qquad (5)$$

All of the parameters and variables have to be non-negative. As negative costs and times are not realistic, all of the parameters are required to be non-negative.

$$U_m, S_i, d_i, d_i^+, a_{j,i}, t_{j,m}, y_{j,m,} x_i \geq 0 \qquad \text{non-negativity}$$

12. Material Flow Subproblem

The fundamental function of a company is dedicated to the purchase of raw materials, their transformation and ultimately the disposal of the finished goods. In order to guarantee an optimal workload in the shop floor of a company, thus it is unavoidable to configure the material flow and its handling efficiently[4]

Without the consideration of obtaining sufficient profit, the company could not perpetuate its operative business. In addition to this fundamental objective which needs to be considered at all times in every company, the work now emphasizes on the purpose to diminish the total makespan through an efficient material flow in the entire manufacturing process. This recent objective target arises out of the generic farm layout in the shop floor. As aforementioned in the subsection “structure of the wafer fab”, the manufacturing sequence cannot be geared to the material flow since the production process of semiconductors is mainly characterized through its great variety of routes as well as its reentrant flows in the shop floor. Thus the influencing value as to the production process is the capital intensive equipment in the wafer fab. Therefore, identical machines are aggregated to so-called work centers and by the means of this generated machine groups the eradication of the miscellaneous production processes within a production line takes place. High priced machines are grouped together in order to create an efficient wafer fab layout but this circumstance causes apart from that long distances among defined work centers[35] The issue of bypassing the spatial distances within a shop floor will not be regarded further in this section. The chapter relating to the subject automated material handling system (section 12.4) will discuss potential approaches for the optimization of the material flow between diverse work centers.

However, our material flow issue is mainly aimed at optimizing the processing of arriving lots within a work centre. As the multitude of lot types is requesting different processing routes through the wafer fab, it is quite potential that several lots are waiting at certain work centers for their further handling. Since a semiconductor consists of several layers, a couple of

work centers need to be traversed a great number of times during the whole manufacturing procedure. This circumstance might lead to congestion in front of work centers and thus certain lots have to wait for their processing. This situation is exactly the main issue of our subproblem and our objective entails optimizing the assignment of lots to the machines in different work centers.

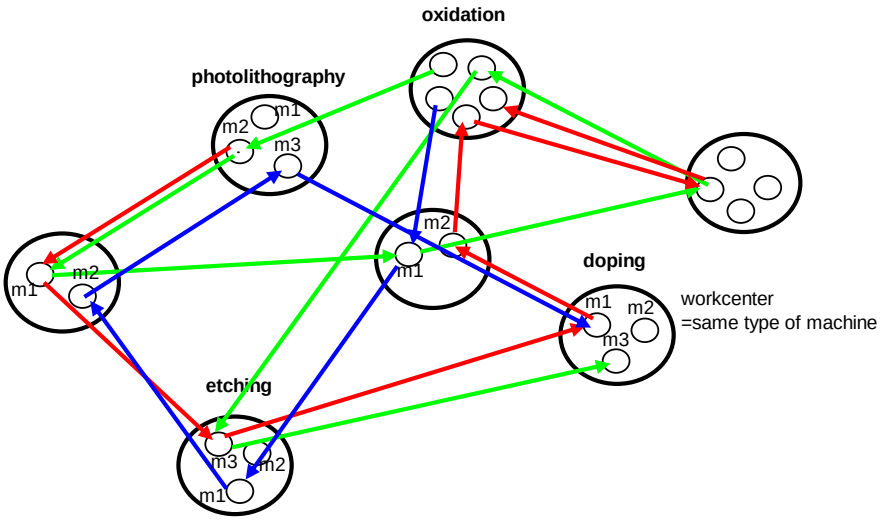

Figure 15: Lot scheduling within the wafer fab

The production of semiconductors encompasses a great number of different types and as well each lot type requires a different processing route through the shop floor. By way of illustration, this figure displays three different lot allocation sequences by the means of the red, blue and green lines, whereas these apparent lines need to be determined beforehand by the shop floor operator. However, the schedule of the arriving lots to one single work centre is the topic of our present subproblem. At some work centers lots are assigned to the same machine, in order to prevent waiting times or even congestions in front of the work centre the lot assignment to machines needs to be optimized. The sequence of the lot processing on one single machine is also subject of our statement of the problem.

As to display the whole issue in more detail, the next illustration abstracts solely one work centre with a number of arriving lots in front and their assignment towards the machines. These available lots in front need to be

scheduled optimally towards not operating or to be precise free machines. The optimal schedule of lots to machines can be enhanced by means of priority rules, whereas this subject matter will be regarded in detail subsequently. For instance, three arriving lots are assigned to machine 4 but this tool is not capable of processing all three lots simultaneously. That is why these lots need to be scheduled prior to there subsequent processing. However, it is not obvious in this figure which of these three arriving lots obtains priority.

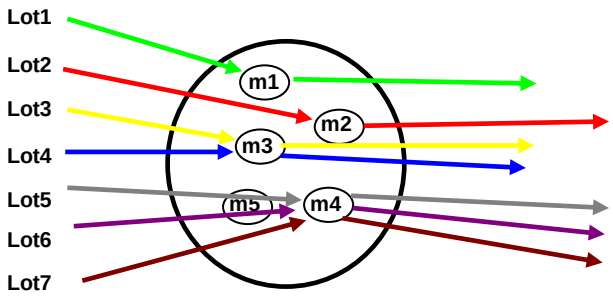

Figure 16: Work centre in detail with arriving lots

Hence, the two proximate figures attempt to illustrate the necessary schedule of lots more precisely. The initial figure outlines now the material flow problem by means of two facing subsets, whereat one consists of available lots and the other subset displays a single work centre with its quantity of machines. This illustration, however, expresses the opposite view of Figure 16, whilst lots where assigned to the machines. Hence, the machines here reveal which and how many lots will be processed at which particular machine. By way of illustration, lot number 2 and 3 are both determined to be handled on machine 3. As there are two lots but solely one machine, the sequence which of them will be processed prior is not evident in this figure. Owing to this double or even more occupancy, the subject of our material flow problem is the determination of an optimal schedule for arriving or even waiting lots in front.

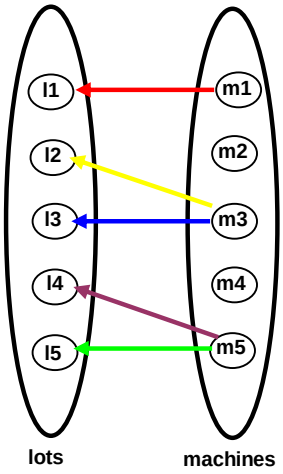

Figure 17: Lot processing on machine without sequence

On the basis of the following illustration the accurate processing of lots on one machine can be seen, whereat lot 4 is handled prior to lot 5 on machine 5. This generated execution might arises through incoming lots and just now processed lots be equipped with the same type and thus obtain priority, apart from that set-up times for different arriving lot types would be necessary.

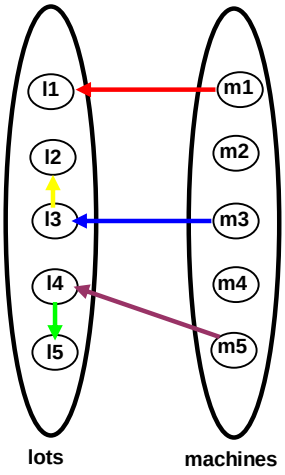

Figure 18: Lot processing on machine

In the course of this optimization, the underlying thesis also aims at minimizing the total makespan, whereat the makespan can be characterized as the utilized time for all machines in a work centre from the beginning till the end of an entire lot sequence. But before this work pays attention to the mathematical formulation of our material flow subproblem,

the thesis points out the conformability to an already existing and likewise well known problem: the Vehicle Routing Problem (VRP). Therefore, the next section provides prior a generic introduction of the general VRP and its potential modifications. Subsequently, the chapter outlines similarities of our basic material flow issue to a special variation of the VRP the so-called open VRP.

12.1 General Characterization of the Vehicle Routing Problem

In order to solve this underlying subject, the material flow problem can be formulated as a Vehicle Routing Problem (VRP) or as vehicle scheduling problem. A general VRP illustrates the delivery or the collection of finished goods between its depots and customers. Typical applications of the VRP, for instance, are the collection of waste, school bus routing and the generation of routes for the ambulance, however, it can also be seen as an extension of the generic traveling salesman problem (TSP), whereat the VRP consists of more than one entire route. A TSP displays solely one route of a salesman in which the traveler visits all destination points within this determined route[64]

The delivery of goods in a classical VRP includes the service of a determined set of costumers accomplished in a predefined time period by a set of vehicles. These vehicles are positioned at one or miscellaneous depots and are operated by the means of a set of drivers. The routes which are accomplished by the means of transportation are generated with the aid of a road network. The main object of a general VRP is the generation of a set of routes with least transportation as well as service costs and time consumption. Each route is executed by one vehicle which can deliver or pick up one or several customers. All these vehicles are forced to start at one particular depot and as well return to the same one. All gained routes are obliged to pervade all constraints which are established in a linear programming (LP) model[64]

The illustration stated beneath provides a potential solution for a Vehicle Routing Problem, for instance, consisting of three different routes. All

depicted routes start and end at one depot and contain a number of customers (destination points, baskets, etc.).

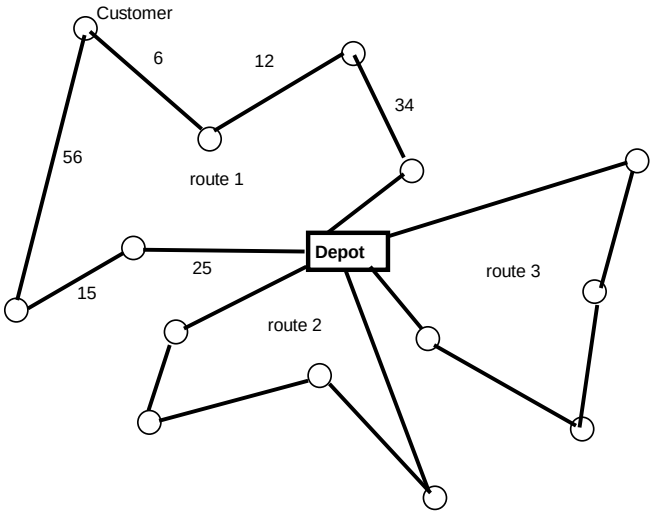

Figure 19: general Vehicle Routing Problem

This generated graph illustrates arcs and vertices, whereat arcs represent the distances between two customers which are projected through vertices. Each arc that connects two customers is validated with a certain amount of travel costs or distances. These arcs can be either directed or undirected. Directed arcs can only be traverse in one direction, whereat undirected ones can be utilized several times in either direction. Similar to the arcs, the vertices also possess specific attributes in the graph:

- Each vertex requests a certain quantity of demand, which needs to be satisfied
- The node in the graph implies the location of the customer
- Every vertices (=customer) possesses a determined period of time a so-called time window in which the costumer needs to be served, whereas these time windows can be further divided into soft and hard; soft time windows are permitted to be violated by adding penalty costs. However, a hard time window constraint needs to be accomplished exactly with no violation, otherwise the vehicle has to wait to start its service[9]
- Also the duration for the delivery or collection of the goods can be restricted to a certain amount of service time

This figure implies solely one depot where all means of transportation are collected. For instance, several depots can be located in the roadmap and are responsible for performing the delivery or collection of items. These means of transport also occupy certain properties:

- Every vehicle starts its route from the home depot but obtains the possibility to return after the completion of the route to another depot; in the existing figure the vehicles are forced to start and end their operations in their home depot
- Each vehicle possesses a capacity limit. So the carried along load cannot exceed the appointed capacity
- The amount of utilized vehicles should be as low as possible according to the fact that every used vehicle causes costs

All these characterizations are valid for a general Vehicle Routing Problem. Additionally to this generic description of the VRP, a number of amplifications have been developed.

The figure stated beneath provides a survey over potential variants of the general Vehicle Routing Problem. A capacitated VRP displays that solely the capacity restriction is implied on the vehicle. Additionally, this kind of variant states that the requested demand of a purchaser cannot be split and thus every customer needs to be served by one single vehicle. The distance constrained CVRP is an amplification of the CVRP, which furthermore implies a maximum distance restriction. A VRP with time windows signifies that each customer needs to be served within a predetermined time interval. Otherwise waiting times or penalty costs might occur. A VRPB accepts the option of backhauls and a VRPPD implies as well as pick and delivery operations[64]

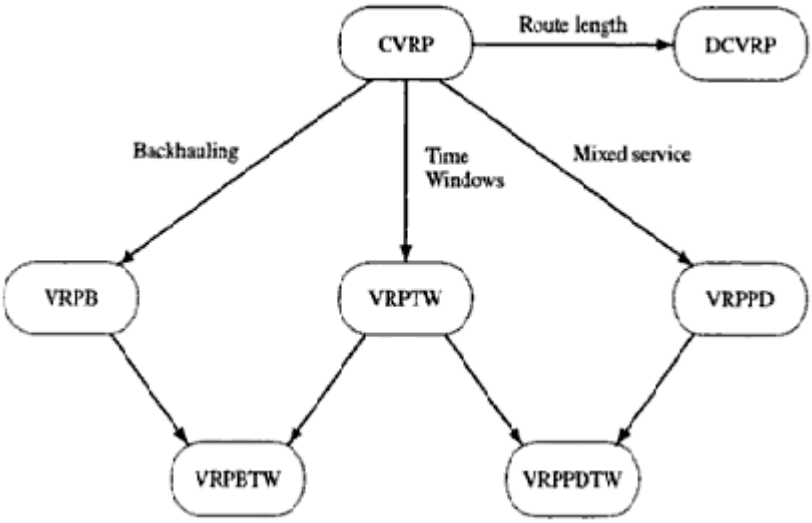

Figure 20: Survey of basic VRP´s [64]

12.1.1 Open Vehicle Routing Problem

As our underlying subproblem is related to the already acquainted VRP or more specifically to the open route variant, the following subsection will therefore deal with the general introduction of this specific version of the VRP.

In the generic variant of the Vehicle Routing Problem (VRP) a sequence of deliveries or collection of items for each vehicle in a homogenous fleet situated at one single depot is generated. All demanders need to be served and simultaneously the total distances traversed by the fleet should be minimized[40] A homogenous fleet features through vehicles of the same type, size and cost, whereas a heterogeneous fleet exhibits different vehicle kinds[3]

Each vehicle in this homogeneous fleet obtains a fixed capacity and occasionally a restriction concerning the route length limited to a maximum upper bound. Each demander on the tour requests a determined quantity and is served by exactly one vehicle which is forced to start from and return to its home depot.

In comparison to the standard VRP, the vehicle in an open Vehicle Routing Problem (OVRP) is not returning to its home depot after servicing the last customer on a route. These generated routes in an OVRP can be related to so-called Hamiltonian paths, whereas a Hamiltonian path features through servicing each vertex exactly once in an undirected graph without returning to the node in which the path began[18] The objective of the OVRP compared to the standard version lies in the fact that the number of accomplishing vehicles is minimized but nevertheless guarantees that all customers on the routes are satisfied[40]

Schrage was the pioneer in this scope who found the first appropriate description of the OVRP and tried to categorize the fundamental properties of VRPs discovered in practice.

“A vehicle can be characterized by at least the following three characteristics: its (multidimensional) capacity, cost rate, and whether it makes open or closed trips. In a closed trip, a vehicle returns to its starting location; in an open trip, it may not. For example, relative to private vehicles, common carrier vehicles tend to have a higher cost/kilometer; however, they make open rather than closed trips. … An air express courier which has planes depart from a single depot city early in the morning making deliveries and then has each plane retrace its route late in the evening making pickups effectively has open routes.”[58]

The appearance of an OVRP by FedEx, whilst generating “incomplete” delivery routes of airplanes, was described in the work of Bodin et al. For instance, an airplane is leaving a city in order to make deliveries to several other cities but after its route termination the airplane does not return to its initial city. This airplane completes its delivery route while servicing the last city and simultaneously initiates the route of collection from the final city of its preceding route. The authors Bodin et al. seized on the existing algorithm of Clarke and Wright which was applied to evolve an open route for each airplane and described a variant thereof[7]

The OVRP is also detected in the newspaper home delivery problem. The newspaper company signs on a carrier as to make distributions to homes.

Yet the company solely puts emphasis on the accomplished path with the last distribution site. Having served the last home site the assigned carrier will not receive further compensation[39]

The compensation model is the usual motive when a routing problem suits the OVRP scheme. While dealing with the formulation and the solution of real-world issues, Levy experienced that companies which are not compensating the contractor after servicing the final delivery site are interested in generating an efficient path which satisfies all demanders on the route without returning to the home depot. In practice this theme is encountered when the executor who undertakes the deliveries is not an employee of the company. At the most these external contractors obtain their own means of transportation and account for their own arising vehicle costs and therefore might be compensated on the basis of a model with mileage. This is meaningful because the contractee does not put emphasis on the necessity that the contractor and/or the vehicle to return to the home depot. The company would need to pay additional compensation to the contractor if his compensation is based on the mileage and includes the way back as well. The objective of the company entails in generating an efficient path without the heading back to the point of origin[39]

The OVRP approach gained only little attention in the early 80s and 90s. By contrast, since 2000, a couple of researchers have seized on this new version and have attempted to improve existing results by means of Tabu Search, deterministic annealing or neighborhood search[39]

12.1.2 Adjustments and similarities of the standard VRP and the open VRP to the material flow subproblem

In the following section the work tries to point out potential similarities of the general VRP and the open VRP to our existing material flow problem concerning the main purpose, the generation of routes, the depot, the demand and the network.

The main subject of a generic VRP implies the diminution of transportation costs or distances or as well as the minimization of the number of applied vehicles in an open VRP. In comparison to our underlying material flow problem the thesis is aiming at reducing the total makespan through the shop floor. As aforementioned in section 10.1, the makespan characterizes itself through the utilized time of handling a total lot sequence within a work centre. Consequently, there is no common ground regarding the main purpose of the present issue with the standard VRP as well with the open setting.

Equal to both VRP predecessors, the material flow problem is aimed at generating cost-efficient and likewise time-saving lot allocation routes through various work centers, whereas these gained routes are not terminated in the point of origin. Consequently, this property of our subproblem is similar to the open VRP approach.

As there are goods which are delivered or collected alongside the routes, the material flow problem displays the processing of different lots on distinct machines which are grouped together in so-called work centers. Owing to this there exists no “general depot” which is responsible for the release of goods. Thus in our optimization problem there are no finished commodities which are transported on routes or delivered to customer, the material flow issue rather displays the transformation process of a single wafer circuit into a finished semiconductor transported in so-called lots.

By comparison, each node in both previously described VRP versions requests a distinct quantity of demand that need to be satisfied within the

statement of the problem, whereat in our material flow problem the amount of lot starts at the beginning of the day is obligated to be the same at the end of one entire processing day. Thus the demand, which is obliged to be accomplished in either VRP variants, renders somehow a certain quantity of lots in the material flow problem that are processed in a determined period of consideration. In this particular case, the period encompasses one working day comprising 24 hours. This predefined time period does not signify that the machines are disconnected after 24 hours. This timeframe should solely appoint a certain period for our observation.

The network for our underlying material flow problem consists solely of directed arcs. This implies that the graph can only be passed through in one direction. The underlying graph mainly results out of the existence of predetermined sequences for each lot. So each lot has a certain route to traverse while it is dispatched through the shop floor, whereas the optimal predefinition of lot execution within one work centre is the main subject of our subproblem[47]

The figure headed beneath summarizes the main properties of VRPs compared to our present subproblem. Since the objective of either VRP variants is aimed at generating cost efficient routes initiating from the depot to its customer, the lot processing on one machine can also be displayed as a route of consecutive lots handled by one machine. After the machine processed one lot it “moves forward” owing to handle the preceding assigned lots.

In all versions of VRPs, three main characteristics can be found that excel this routing problem and are thus essential to generate either a delivery or a collection route: depot, customer and vehicle. However, in our subproblem, these properties cannot be translated in such a way, as the material flow problem does not consist of actual customers or vehicles. All distinctive criterions are comprised in our present subproblem but effectuate another notation. Hence, the preceding work centre with its machines displays the common depot of the VRPs. Usually the vehicle represents the mean of transportation which delivers a certain quantity,

whilst in the material flow model the machine acquires this kind of task. The processing machine pervades simultaneously the function of a vehicle as in a standard VRP but as well the good which is delivered at a certain customer site, in this particular case, the machine handles a particular lot. Thus the machine reflects the purpose of serving one customer within the routing issue. Recapitulating, the machine (=vehicle) is dispatched from a preceding work centre (=depot) and processes on lot (=customer) after another. By way of illustration, the following figure points out these resemblances to VRPs.

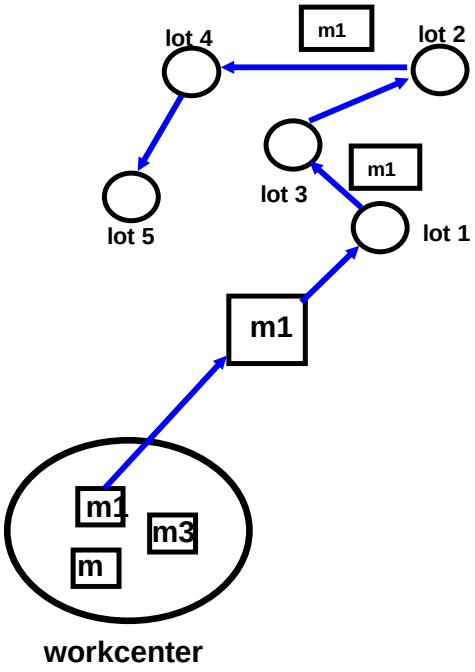

Figure 21: Material Flow issue

12.2 Mathematical Formulation

After the generic description and the comparison to the related VRPs, in this chapter the thesis pays attention to the mathematical formulation of the deterministic model.

Before being able to illustrate the underlying subproblem in its mathematical form, prior the presentation of several new parameters is necessary as to comprehend the afterwards succeeding formulation. Therefore, the parameters and indices headed below should enable a problem-free comprehension of the model.

Indices

$_i \epsilon I$ … lot

$_m \epsilon M$ … machine

Essential parameters

t_i^m … time for processing lot i on machine m

c_{ij}^m … set-up times for processing consecutively lot j and lot i on machine m

$INITIAL_i^m$ … essential set-up time for initial setting-up on machine m

$FINAL_i^m$ … required set-up time for final setting-up on machine m

a_i … release date of lot i

b_m … first potential starting time of machine m

N … very large number

Decision Variables

$start_i^m$ … starting time of lot i for its processing on machine m

x_i^m … lot i is assigned to one machine

$initial_i^m$ … $\begin{cases} 1 & \text{if lot i is the initial lot on machine m} \\ 0 & \text{otherwise} \end{cases}$

$final_i^m$ … $\begin{cases} 1 & \text{if lot i is the final lot on machine m} \\ 0 & \text{otherwise} \end{cases}$

y_{ij}^{m} … $\begin{cases} 1 & \text{lot i and j are processed subsequently on machine m} \\ 0 & \text{otherwise} \end{cases}$

Z_{max} … maximum makespan

Z … Total Makespan

The objective of the underlying sub problem, as aforementioned, focuses now on the minimization of the makespan, whereat makespan is defined as the utilized time for processing a lot sequence within one single work centre. Therefore, the work concentrates on the reduction of the processing times for each handled lot and the potential final set-up time for the ultimate lot. This objective is illustrated through the two formulas mentioned below

$$\text{Minimize } \sum Z_m + Z_{\max} * 5 \qquad \forall m \in M \qquad (6)$$

Formula (6) solely reveals that the total makespan within a work centre should be minimized for each machine m. This objective is further displayed through formula (7) which predicates that the starting time of the penultimate lot plus its processing time and its final machine set-up in the lot sequence are obliged to be as small as possible. If the decision variable x_i^m assumes value 1 the whole constraint is binding and thus the right hand side is forced to be smaller. Otherwise the left hand side assumes a very large value and therefore the formula becomes redundant.

$$Z_m + (1 - x_{i,m}) * N \geq start_{i,m} + t_{i,m} + final_{i,m} * FINAL_{i,m}$$
$$\forall i \in I \ \forall m \in M \qquad (7)$$

The previously displayed objective needs to be fulfilled under the constraints mentioned below.

The two subsequent inequalities ensure that within one lot sequence, processed on machine m, there exists at most one starting and one final lot.

$$\sum_{i \in I} initial_{i,m} \leq 1 \qquad \forall m \in M \qquad (8)$$

$$\sum_{i \in I} final_{i,m} \leq 1 \qquad \forall m \in M \qquad (9)$$

This constraint guarantees that a lot sequence obtains either an interface of lot i has an successor, so decision variable y_{ij}^m takes the value 1, or the assigned lot is the ultimate lot in the sequence and thus $final_i^m$ equals 1. The right hand side forces that at least one of those binary variables is greater than zero.

$$\sum_{m \in M} \sum_{j \in I, i \neq j} y_{i,j,m} + \sum_{m \in M} final_{i,m} = 1$$
$$\forall i \in I \qquad (10)$$

The auxiliary variable x_i^m indicates that lot i is assigned to machine m. This assignment can look like x_i^m either i.e. a lot is processed on the machine m if it has a successor on the machine ($y_{ij}^m = 1$)or it is its final lot..

$$x_{i,m} = \sum_{j \in I, i \neq j} y_{i,j,m} + final_{i,m}$$
$$\forall i \in I \ \forall m \in M \qquad (11)$$

The following equation illustrates the flow conservation which forces that if a lot is scheduled to be the final one of a sequence thus there has to be an interface to a previous lot i. Also, if the lot is designated to be the starting lot of a route there has to be a setup to a succeeding lot j.

$$\sum_{j \in I, i \neq j} y_{i,j,m} + final_{i,m} = initial_{i,m} + \sum_{j \in I, i \neq j} y_{j,i,m}$$

$$\forall i \in I \ \forall m \in M \qquad (12)$$

The utilized time of a preceding lot j needs to be greater or at least equal to the occupied starting time, processing time and set-up time of lot i. No set-up is essential if lot i and j obtain the same category. The parameter N assumes a very large number, for instance, 1000. If the binary variable y_{ijm} equals zero, then the right hand side of the equation obtains a very large value and thus the inequality gets ineffective. Otherwise further time is necessary to convert the machine for different lot types. However, the second part of the right hand side would takes the value zero and therefore the formula is restrictive.

$$start_{i,m} + t_{i,m} + y_{i,j,m} * c_{i,j,m} \leq start_{j,m} + (1 - y_{i,j,m}) * N$$

$$\forall i, j \in I, i \neq j \ \forall m \in M \qquad (13)$$

The earliest starting time of lot i on machine m arises from the release time of lot i which is assigned to machine m.

$$a_i * x_{i,m} \leq start_{i,m}$$

$$\forall i \in I \ \forall m \in M \qquad (14)$$

The formula mentioned below ensures the requirement, that each machine is occupied to its first potential time. The first batch is only allowed to start its further processing if machine m was already released and additionally accomplished a possible initial setup.

$$b_m * x_{i,m} + initial_{i,m} * INITIAL_{i,m} \leq start_{i,m}$$

$$\forall i \in I \ \forall m \in M \qquad (15)$$

In order to receive a convincing and comparable statement, nevertheless it is necessary to include any kind of occurring time units. Although not all machines are utilized in a work center they obtain certain release times which also need to be considered in the overall objective function. This statement is guarantee through the underlying formula which indicates that each tool group obtains its unique makespan which is at least greater than the release time. The makespan accounts solely the machine's release time once no lots are assigned.

$$Z_m \geq b_m \qquad \forall m \in M \qquad (16)$$

The following constraint defines the maximum makespan of any machine. This formula will further be incorporated in the objective function in order to generate a balanced assignment of lots to the machines within a work center.

$$Z_{\max} \geq Z_m \qquad \forall m \in M \qquad (17)$$

Since the formulation needs to be solvable, certain parameters are not allowed to obtain negativity.

$$x_{i,m}, start_{i,m}, Z_m, Z_{\max}, t_{i,m}, a_{i,m}, b_m, c_{i,j,m} \geq 0$$

$$\in R$$

The following decision variables are restricted to be binary and thus can only obtain the value 1 or 0.

$y_{i,j,m} = \{0,1\}$ binary variable

$\text{initial}_i^m = \{0,1\}$ binary variable

$\text{final}_i^m = \{0,1\}$ binary variable

12.3 Abbreviated Mathematical Formulation

By means of omitting one decision variable and simplifying the preceding model formulation, this thesis attempts to generate a less complex model which solves this lot assignment problem within more reasonable computing time. In order to contrast both formulations according to their running time, the work prior displays the amendments concerning the previous mathematical model.

As variable x_i^m is left out in the abbreviated mathematical formulation, decision variable $final_{im}$ will replace the left hand side. This change guarantees that the makespan of each sequence is completed after the ultimate lot accomplishes its processing and additionally its machine related final setup.

$$Z_m + (1 - final_{i,m}) * N \geq start_{i,m} + t_{i,m} + final_{i,m} * FINAL_{i,m}$$
$$\forall i \in I \ \forall m \in M \qquad (18)$$

In the second alteration, variable x_i^m is replaced by y_{ij}^m + $final_{im}$ relating to the earliest starting time of lot i. This new formula (17) ensures that lot i starts its processing on machine m if lot i is assigned to this machine and only when the lot is already released. This assigned lot i can be identified either as y_{ij}^m or $final_{im}$.

$$a_i * (y_{i,j,m} + final_{i,m}) \leq start_{i,m}$$
$$\forall i, j \in I \ \forall m \in M \qquad (19)$$

As previously illustrated, variable x_i^m was replaced by the term of y_{ij}^m + $final_i^m$. This modification also needs to be considered in terms of the earliest release time on any machine m.

$$b_m * (y_{i,j,m} + final_{i,m}) +$$
$$initial_{i,m} * INITIAL_{i,m} \leq start_{i,m} \qquad (20)$$

As decision variable x_i^m is omitted in this abbreviated mathematical formulation, the constraint concerning the auxiliary variable x_i^m is redundant.

12.4 Automated Material Handling System (AMHS)

In the case of Infineon Technologies AG no automation of the material handling and its material flow in the wafer fab exists. As there are miscellaneous methods in today's semiconductor industry to improve this efficiency, the following chapter should give an insight of potential ways to implement such methods and provide advantages towards a possible integration in the entire manufacturing process.

Several procedures and consistently more efficient approaches in the current semiconductor literature are discussed as to achieve the purpose to bridge these spatial distances and thus create an effective material flow with diminished lead times. One of the current approaches deals with the implementation of an automated material handling system (AMHS) in the wafer shop floor. This procedure supports the resolution of existing manufacturing issues as well provides more opportunities for adjustments in future towards material handling[47]

"[In addition, it [the work] will explore specific configurations] Intel is considering for future automation requirements, and evaluate the critical infrastructure and supporting elements necessary for making AMHS an enabling asset in the factories future."[55]

The AMHS is characterized through inter-bay and intra-bay transportation. Inter-bay is dedicated to the transport and storage of wafer lots and reticles between processing bays within a fab, whereas intra-bay implies the handling of material inside a bay or an area and furthermore consists of loading and unloading tasks as well metrology instrumentation[55]

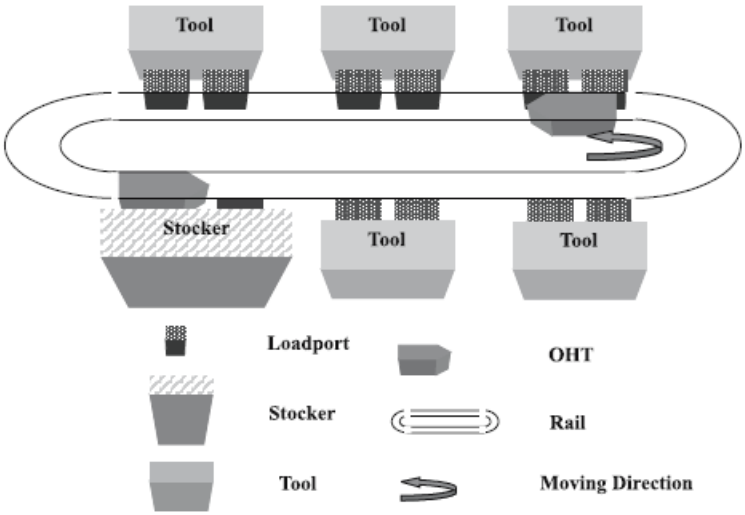

Figure 22: Intra-bay material handling [41]

The AMHS is a significant influencing value in the factory automation (FA) of today's semiconductor fabrication. The subsequent enumeration should give an insight for the efficient applicability of AMHS in the semiconductor fab.

- Improvement in the efficiency of lot storage
- Previsible transportation through wide distances in the fab
- No monotony of tasks
- More motivation within the workforce
- Deletion of ergonomic and safety disturbances
- Emendation in workload as well in lead times
- Increase in the cost efficiency
- Diminution of variability caused through human beings[55]

On the basis of consistently preceding requirements towards the economic and technological changes in the entire semiconductor industry, the following trends concerning the current challenges for AMHS in a wafer fab are listed beneath.

- Expansion of the factory layout requires improvements in logistics
- Inventive and quick enhancement of the shop floor
- Flexibility of the fab layout due to alteration in technique and instrumentation

- Requirements for qualified and trained operators
- Rise of the machine utilization[55]

The explosive growth of the semiconductor industry accompanies the requirements for a consistently increase in the wafer size and their improved capacity. Thus to these new challenges, the layout needs to cope with larger fabs, more wafer starts per day as well with the request for enlarged cleanrooms. So as to deal with these underlying issues, a manual handling and transportation through the fab would not yield an efficient utilization of the shop floor capacity. Therefore, inter-bay transportation systems have been adjusted to resolve these current issues and traverse now the threefold quantity of lots through the production line.

The enormous demand of semiconductor products as well results in the necessity to expand the factory layout. In order to cope with this challenge inter-bay transports are aligned with inter-floor elevators to procure a high-volume lot transport. Additionally to the greater volume transportation, this approach also declines building footprint and reduces logistical issues.

Due to the fact that the factory is subjected to continuous modifications in the fab equipment and in the layout, the approach towards automation achieves the flexibility requirement regarding rapid adjustments.

Yet new wafer fabs with the state-to-the-art technologies are requesting for qualified and high-skilled operators, whereat special trainings for the existing workforce cause additional costs. However, workers are subjected to more complex tasks and are handling highly expensive equipment which reduces the monotony of duties in the shop floor and advances the overall motivation in the staff. Through the implementation of AMHS the repetitive jobs such as loading and unloading, transport or storage are executed by intra-bay systems[55]

Especially for bottleneck machines, which need to be traversed a certain number of times during a whole production process and thus should not suffer of material constraints, the implementation of intra-bay AMHS

supports the factory output by the means of equipping these constraint areas with sufficient material, enabling the machine to run without disruptions and prevent down times during the manufacturing process.

Furthermore, the work gives an insight into the qualification options regarding at first inter-bay and afterwards intra-bay transportation systems. Inter-bay facilities are responsible for guaranteeing a secure and fail-safe high throughput of material transportation across great spatial distances. The current trend of overhead monorail transportation has excelled the existence of ground based automatic guided vehicles (AGV) due to their superior throughput capability, smaller factory footprint and owing to the ability to traverse repeated loops. Further advantages of overhead monorail are reflected in the competence of transporting different types of load such as wafer boxes, open cassettes or reticle boxes on solely one vehicle and as well in the minimization of the lot transportation time from one stocker to another through its optimal positioning. For instance, Intel reduced its stocker cycle time about 60% only as a result of the location at the end of one bay and not in its centre. On the one hand the ability to transport a variety of load types reduces on the one hand the factory footprint and on the other hand it augments the flexibility of the tool layout.

An intra-bay facility is accountable for the reliable transport of all load types within the bays. Additionally to this generic task, the intra-bay responsibility contains as well reliable unloading and loading functions. According to the transported load, there are multiple types such as boxes or opened cassettes that can be hauled through the shop floor. From the contamination standpoint, an explicit preference cannot be undertaken since both types are exposed to some kind of impurity. For the factory automation, fabs provided with open cassettes show a superior flexibility to the rearrangement to total intra-bay systems[55]

“[…] stockers supporting boxed intra-bay systems are subject to 50%-70% additional cycles of operation compared to their open cassette counterpart.”[55]

The transport between the bays either rail guided vehicle (RGV) or the free-moving AGVS can be suitable as conveyor, respectively. Both types are ground based transportation methods. Since AGVS obtain reduced throughput and as well need battery recharging stations which consume valuable factory footprint RGVS are preferred for undertaking the intra-bay functions. However, also RGVS have side-effects regarding to their inflexibility from a layout view and they might be subject to potential manufacturing disruptions during their implementation. Nevertheless, RGVS are preferred by multiple semiconductor manufacturers as a result of their very good effort in throughput and in unloading and loading activities[55]

Vehicle type	Throughput per vehicle	Time to position for load/unload	Routing flexibility
Rail guided vehicle	Very good	Very good	Poor
Tape guided AGVS	Good	Poor	Very good
Free-roving AGVS	Good	Poor	Very good

Figure 23: Vehicle type comparison[55]

For the future inter-bay transport will be in the need to unify load port across all machine types rather than advanced multiple axis robots for an unload/load conveyor. An ordinary 2-axis shuttle machine is predicted to displace the 6-axis intra-bay robot. Vehicles can be either free moving AGVS or based on rail. A qualification of rail-based mechanisms appears to be a hoist approach, whereas hoists are suspended from the ceiling and provide a great capability in diminishing bay width in the shop floor[55]

Carrier handling mechanism	Cycle time to load or unload	Placement versatility
Multi-axis robot	Poor	Very good
Simple 2-axis shuttle	Very good	Very poor

Figure 23- 1: Carrier handling[55]

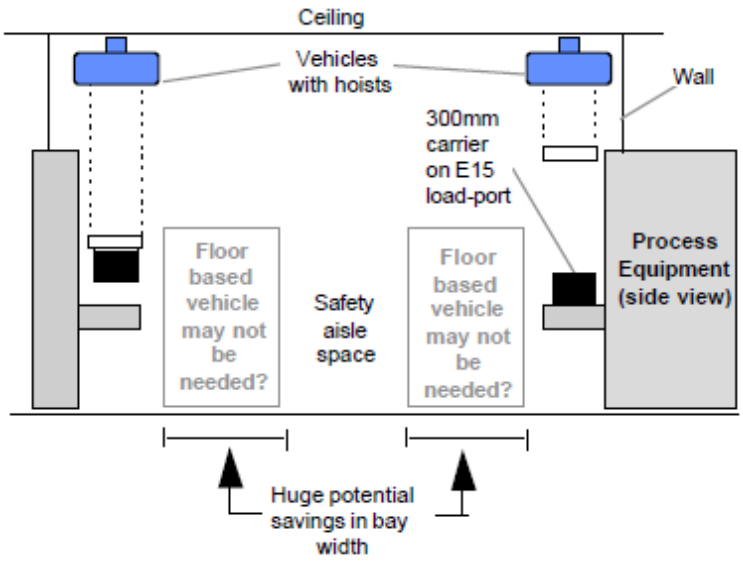

Figure 24: Over hoist intra-bay material handling[55]

As intra-bay mechanisms are becoming pervasive in future due to the fact of the consistently enlargement of wafer sizes, their optimization and adjustment flexibility are therefore challenges which need to be considered for future advisements[55]

This section should solely give an insight of potential ways to organize the internal material handling in the semiconductor manufacturing and as the wafer sizes are steadily increasing the AMHS pervades the necessity of flexible adjustments in the fab. As this issue of material handling and material flow in the shop floor is a current subject in today's semiconductor manufacturing there are steady new approaches towards the efficiency of the material flow in an automated fab. Since Infineon Technologies AG Austria has not yet implemented an automated material handling system, the upcoming updated approaches might excel the responsible operators to convert the existing system towards more automation.

13. Approaches to solving problems in general

For the solution of our underlying subproblems a great range of different solution methods are on hand. Thus, to get an overview of the variety the schedule below should outline the three main categories of existing approaches with its representatives. Furthermore, the following chapter disputes with some selected approaches which are reviewed closely and are afterwards applied in practice on the basis of our two subproblems.

Exact methods:

- Branch and Bound
- Branch and Cut

Heuristics:

- Constructive Methods:
    - Priority rules
    - Nearest Neighbor
    - Insertion
        - Sequential
        - Parallel
- Improvement Heuristics:
    - Intra-tour: r-opt
    - Inter-tour: exchange, move operator
    - Cluster first, route second Algorithms
        - Fisher and Jaikumar
        - The Petal Algorithm
        - The Sweep Algorithm
    - Route first, cluster second Algorithms
        - Giant Tour

Metaheuristics:

- Ant Colony Optimization
- Constraint Programming
- Deterministic Annealing
- Genetic Algorithms

- Simulated Annealing
- Tabu Search[66]

13.1 Exact Methods

This approach, as the notation suggests, contemplates to compute every potential solution until the best out of the generated results is reached. For rather small problems with few instances, this approach provides an optimum performance without a great computational effort. As data sets increase, for instance, the quantity of lots an exact algorithm cannot ensure an optimum route generation within reasonably computation time. Furthermore, two representatives of this solution method are looked at closely[66]

13.1.1 Branch and Bound

Evolved in the 1960s, this exact approach deals with the minimization with the maximization of a subjacent objective, respectively. While executing several steps of this approach some kind of “solution tree” with a number of ramifications occurs. Without considering the constraint of integrability, the model simplifies and is solved by the means of the simplex method. The Branch and Bound algorithm attempts to divide its solution space into smaller subsets and then focuses on the individual optimization of each subproblem. Based on this partition the solution therefore obtains a sort of tree structure. This process of examining the entire solution space and afterwards the decomposition in subsets is known as branching. While in the bounding phase the problem will be relaxed, whereat solely certain branches are evaluated with the aid of upper and lower bounds. Upper bounds are generated for every feasible solution. If a computation of a determined bound submits an inferior result than the previous one then this ramification of the “solution tree” requires no further consideration and thus can be pruned. Lower bounds are received while the problem is further simplified. Owing to these obtained bounds potential solutions can be valuated. The Branch and Bound algorithm stops although either all

potential solutions are considered until no better one can be generated or one bound is sufficient for the goodness of the subjacent problem[66][17]

13.1.2 Branch and Cut

This type of method was generated about 20 years later than its predecessor and thus constitutes an extension of the already existing Branch and Bound algorithm.
This current approach distinguishes mainly through the generation of sectional planes. These planes are located alongside already obtained integer elements within a solution set. By means of additional constraints in a linear programming model, these sectional planes are expressed. After adding this new constraint, the model is launched once more. If the solution again does not obtain integrability, new sectional planes need to be computed. If the creation of further sectional planes is impossible, the Branch and Bound algorithm needs to be accomplished. In comparison to the method consisting solely of the Branch and Bound algorithm, the Branch and Cut method enables a more efficient generation of solutions[49]

13.2 Heuristics – approximation procedure

Many scheduling problems use optimization algorithms which for sure always find the optimum solution. Yet as our subjacent problem exhibits a high level of complexity and is thus denoted as NP-hard[6], an exact optimization algorithm which solves the model optimally within polynomial time cannot be constructed. It might appear that the model's execution exceeds the capability of state-to-the-art computers. In such cases heuristics (suboptimum algorithms) are deployed which tend to but do not guarantee the finding of the optimum solution in reasonable time.[6] Thus, heuristics attempt to approach good solutions by means of problem-specific approximation procedures. The goodness of heuristics also accrues besides its solution quality from its utilized computation time. Thus, the time

[6] The classification of a problem to be NP-hard implies that there is no existing algorithm which solves the subjacent problem optimally with polynomial time.

which is spent owing to generate a good solution should not surmount a benchmark of approximately 3 to 5 % of the observed planning period.[57] The following displayed algorithms should give an insight of the functionality of such heuristics.

13.2.1 Constructive Heuristics

Constructive heuristics display methods to generate solutions step-by-step beginning at point “0”. These built solutions conduce as basis for improvement heuristics.

13.2.1.1 Nearest Neighbor Heuristic

The nearest neighbor heuristic is a deterministic constructive heuristic and class among the so-called “greedy”-heuristics.
In each iteration, only one alternative is taken into account without regarding the rest of the tour planning. This alternative is at that moment the most favorable one. This method starts at a freely chosen point and adds at each iteration the node that posses the shortest distance to its predecessor. The heuristic ends when all nodes are allocated to a route[20] The algorithm for this algorithm is the following:

1. “Start with any city, e.g. the first one
2. Find nearest neighbor (to the last city) not already visited
   Repeat 1 until all cities are visited. Then connect the last city with starting point”[34]

13.2.1.2 Best Insertion

This heuristic is also a deterministic constructive heuristic. At the beginning two random nodes have to be chosen but they should be as far-off as possible from each other. In the first iteration the node with the smallest cumulative distance to both start nodes is inserted. In every following step one node is added that has not been chosen yet. There exist 4 variations for choosing nodes:

1. The node with the smallest distance to the existing tour must be inserted.
2. Chose those nodes that raise the tour the least.
3. The nodes with the biggest distance to the existing tour have to be added.
4. Random choice.

The chosen node must be put at the best place so that the distance of the tour is as small as possible. When all nodes belong to one tour the algorithm stops.

“The following steps have to be done:

1. Select 2 cities A, B and start with short cycle A-B-A
2. Insert next city (not yet inserted) in the best position in the short cycle
3. Repeat 2 until all cities are visited”[20]

13.2.1.3 Priority Rules

A lot of different priority rules are used to build a start solution. The following figure shows the most popular:

Rule	Abbreviation	Description
First-In-First-Out	FIFO	The first job that arrives at a machine obtains the highest priority.
Shortest Process Time	SPT	The job with the smallest processing time has the highest priority.
Critical Ratio	CR	The job with the smallest quotient of time till the process end and the smallest processing time preserves highest priority.
Earliest Start Date	ESD	The jobs are queued due to their starting times. The earliest gets the highest priority.
Earliest Due Date	EDD	The job with the earliest delivery date is manufactured first.
Longest Processing Time	LPT	The job with the longest processing

		time is ranked first.
Shortest remaining Processing Time	SrPT	The job with the smallest remaining processing time is produced first.
Random	Rm	Jobs get random values and the job with the highest value starts.
Highest Value	HV	Jobs with the highest monetary value are manufactured in the first place.

Figure 25: Outline of Priority Rules[68]

These rules can be put together in different manners: additive, multiplicative or alternatively. Not every linkage brings out an improvement, deteriorations are possible[68]

Priority rules exhibit a big relevance in dynamic production processes and so in semiconductor manufacturing. One big problem of these rules display the circumstance that only one production step is regarded and therefore non-optimal solutions might occur in further steps especially when bottleneck machines are in use[67]

13.2.2 Improvement Heuristics

Improvement heuristics need a given solution. Then they try to optimize this solution. In this field many different methods are known and some of them are shown in this chapter.

13.2.2.1 R-Opt

This method changes r edges of a given solution. The old edges are replaced by r new edges. If the solution gets better, the change remains, if not, other edges and nodes are chosen to be substituted. The steps repeat as long as there is no combination left and no better solution can be found. In practice, 2-opt and 3-opt are the most known and used heuristics. The more edges are replaced the higher the computational effort is[1]

13.2.2.2 Intra-tour and Inter-tour Improvements

Intra-tour improvements take place within one single tour whereas inter-tour improvements change customers/nodes between two or even more tours. The most common improvements are:

- Relocate Operator (Move):
  One customer is moved into a new tour, while edges have to be replaced.
- Exchange (Swap):
  Two nodes are swapped. At the same time two customers from different tours are placed into the other tours.
- The λ-Exchange Generation Mechanism:
  A set of tours represent the given solution, a λ-exchange between a pair of routes displays a substitution of a subset of customers by another one to get two new tours and a new neighboring result.
- CROSS-exchange:
  Segments of two tours are simultaneously reinserted into the other route. The orientation, however, of both tours is preserved.
- Geni-exchange Operator:
  A customer/node of the upper route is inserted into the lower one while reordering the lower route.
- Cycle Transfer Operator:
  Simultaneously nodes are transferred between tours either clock-wise or anti-clockwise[34]

13.3 Metaheuristics

As heuristics can be tempted to be trapped in local optimum, metaheuristics attempt to generate a better solution in a subordinated (Meta) plane. Metaheuristics compared with heuristics are not problem-specified and are thus avertable to a greater range of optimization problems. As well as heuristics, metaheuristics cannot guarantee the detection of the optimum result but it enables to compute feasible and good solutions within a more rational computation time. In general,

metaheuristics encompass two paces, whereas prior the sub heuristic will be implemented which afterwards is guided by the subordinated heuristic in order to surmount local optimum. In the following the thesis outlines some acquainted representatives of this solution algorithm[30][69]

13.3.1 Ant Colony Optimization

Observed in natural environment, ant colonies are ramified systems that represent a highly structured social organization. Accordingly to this structuring, ant colonies execute highly complex functions that in some cases excel the individual capabilities of a single ant. This observed behavior served as a source of inspiration to derive a novel algorithm for the solution of hard combinatorial optimization problems.
“The main idea is that the self-organizing principles which allow the highly coordinated behavior of real ants can be exploited to coordinate populations of artificial agents that collaborate to solve combinatorial problems.”[21]

The inspiring source for the Ant Colony Optimization (ACO) is the marking of paths by means of pheromones. These scents are used as an indirect communication medium in order to persuade other ants to follow these distinctive trails. For instance, a foraging ant deposits an aromatic essence on the ground which enhances the probability that other ants will follow this “secure” trail in order to accomplish a food source. Ants tend to traverse paths which display a higher level of pheromones since this indicates shorter distances to the source and as well a good food quality. However, paths are discarded as the pheromone odor evaporates[21][29]

A set of n ants is deployed into the graph to generate n solutions per iteration. The ants which move along the graph apply a decision policy based on the two parameters trails and attractiveness. The ants generate a feasible solution for the subjacent problem while accomplishing their moves. As the ant completes its pace through the graph, it diversifies the trail quality through its release of additional scents. Thus the trail information is adjusted and will guide the search for further ants. Since all

solutions are generated, feasible and good solution paces of appointed ants will be accumulated into a matrix, whereas it is used further as basis for preceding iterations[44][21]

13.3.2 Tabu Search

The Tabu Search algorithm was proposed by Glover in 1986 for a number of combinatorial problems. In some cases, the Tabu Search method provides solutions near optimality and is therefore compared with other metaheuristics more effective. The basic concept of the Tabu Search is the permission of also non-improving solutions. However, moves to a prior generated solution are prevented through the use of so-called Tabu list which memorizes the recent history of the search results[29]

By means of a constructive heuristic an initial solution will be generated and through the Tabu Search further processed and improved. During the Tabu Search all solutions in a neighborhood are analyzed. Through modifications within the neighborhood, new solutions can be generated. If it is achievable to obtain a better solution then this new yield is defined as the current starting solution. In order to avoid cycles through a recurring selection of similar elements in the neighborhood, the previously mentioned Tabu list tries to prohibit (set Tabu) certain conditions for a predefined period of time (iteration) owing to generate new solutions without getting trapped in local optimum. Tabu lists quote forbidden elements which are not entitled to be part of a generated interim solution. As an improved solution was obtained, the restricted elements or conditions within the Tabu list are adjusted and can be released for the preceding iteration. [28]In order to maintain the efficiency of the subjacent algorithm, the parameter which defines the length and the duration of these Tabu lists need to be selected accurately. If the selected parameter exhibits a too small value it might happen that cycling movements occur, whereas a huge value creates a too restrictive solution space[69][19]

13.3.3 Variable Neighborhood Search

The Variable Neighborhood Search (short VNS) is another representative of a recent metaheuristic which tackles combinatorial and global optimization problems. The principle characteristic is the systematic alteration of neighborhood structures within a local search in order to escape the local optimum trap.[46] This iteratively alteration increases the problems size and will be proceeded until an improvement is detected. The achievement of a determined termination condition leads to a stop of the algorithm[42] The VNS approach is based on three main properties:

- A local minimum with respect to one neighborhood structure is not necessary so for another.
- A global minimum is a local minimum with respect to all possible neighborhood structures.
- For many problems local minima with respect to one or several neighborhoods are relatively close to each other[32]

In the last few years the VNS approach enjoyed great popularity since a multitude of authors seized the standard algorithm and evolved a wide range of novel variants. Thus the application of the VNS approach to a great range of small sized as well as large-scale optimization problems was enabled. As the explanation of several approaches would occupy too much time, the work therefore puts emphasis on the closer consideration on the basic implementation of the Variable Neighborhood Search.

For the implementation of the VNS, a set of selected neighborhood structures denoted with N_k $(k=1,\dots,k_{max})$ and a set of solutions $N_k(x)$ are determined. The detected solution x'' is identified as local optimum if there is no solution which satisfies the following constraint $f(x) < f(x'')$. This implies that no better solution than x'' was found in this neighborhood. Furthermore, an initial solution x must be found, the parameter k_{max} has to defined and also a termination criterion needs to be determined. After this initialization phase, the algorithm runs its first iteration, whereat the parameter k is equated with 1. The quantity of iterations is bound to the

previously determined factor k_{max}. In the following the thesis distinguishes three different phases: In the shaking phase a solution denoted as x' is generated randomly from the k^{th} neighborhood of x. For the generation of the local minimum *x''* a local search algorithm such as improvements heuristics (r-opt, exchange operators) is applied in each iteration. In the final phase these gained results are compared to each other. If the local minimum *x''* exhibits an improvement then *x''* will be placed as the new initial solution for the following iterations. The parameter k will be reset to the value 1 as this implies local minimum in the first neighborhood structure. Otherwise k is increased by 1 and the algorithm runs once more[33]

By way of illustration, the following figure should shortly instance the major properties of the Variable Neighborhood Search.

Input: maximum number of neighbourhoods $k_{\max}$, number of local searches in each neighbourhood L.
loop
  Set $k \leftarrow 1$, pick random point $\tilde{x}$, perform a local search to find a local minimum x^*.
  while $k \leq k_{\max}$ **do**
    Consider a neighbourhood $N_k(x^*)$ of x^* such that $\forall\ k > 1\ (N_k(x^*) \supset N_{k-1}(x^*))$.
    for $i = 1$ to L **do**
      Sample a random point $\tilde{x}$ from $N_k(x^*)$.
      Perform a local search from $\tilde{x}$ to find a local minimum x'.
      If x' is better than x^*, set $x^* \leftarrow x'$, $k \leftarrow 0$ and exit the FOR loop.
    end for
    Set $k \leftarrow k + 1$.
    Verify termination condition; if true, exit.
  end while
end loop

Figure 26: VNS algorithm[42]

13.3.4. Simulated Annealing

The SA is a local search algorithm and belongs to the group of metaheuristics because of its capability of escaping from local optima. Normally this technique is used for solving discrete but also, to a lesser extent, continuous optimization problems.

The name itself results from its analogy to the process of physical annealing with solids. In this process, a crystalline solid gets fast heated and cooled down very slowly until the most regular possible crystal lattice

configuration is achieved, whereas no crystal defects occur. “If the cooling schedule is sufficiently slow, the final configuration results in a solid with such superior structural integrity. Simulated annealing establishes the connection between this type of thermodynamic behavior and the search for global minima for a discrete optimization problem.”[29]

The SA contains several iterations. In each iteration, two solutions for the objective function are generated and compared:

- the current solution, and
- a newly selected solution.

If an improvement is found, the better solution is always accepted, while a fraction of non-improving results are only taken in the hope of escaping a local optimum via a global one. Here the temperature parameter is an issue. The temperature parameter is usually decreasing at each iteration and it shows the probability of accepting a non-improving solution.

In general, the simulated annealing allows hill-climbing moves in order to escape local optima. As mentioned before, the temperature parameter moves toward zero, hill-climbing moves appear less often and the solution distribution is approaching to a form in which all the probability is centralized on the set of globally optimal solutions.

The most important terms that are used for this algorithm are the following:

- Ω: the solution space
- f: Ω→: an objective function defined on the solution space, this function must be bounded in order to make sure that ω* exists.
- ω*: global minimum (ie ω*Є Ω such that $f(\omega) \geq f(\omega^*)$ for all ω Є Ω)
- N(ω): neighborhood function for ω Є Ω

The pseudo code for the simulated annealing is the following:

- “Select an initial solution ω Є Ω
- Select the temperature change counter k=0

- Select a temperature cooling schedule, t_k
- Select an initial temperature $T = t_0 \geq 0$
- Select a repetition schedule, M_k that defines the number of iterations executed at each temperature, t_k
- Repeat
- Set repetition counter m=0
    - Repeat
    - Generate a solution ω' Є N(ω)
    - Calculate $\Delta_{\omega,\omega'} = f(\omega') - f(\omega)$
    - If $\Delta_{\omega,\omega'} \leq 0$, then $\omega \leftarrow \omega'$
    - If $\Delta_{\omega,\omega'} > 0$, then $\omega \leftarrow \omega'$ with probability exp $(-\Delta_{\omega,\omega'}/t_k)$
    - $m \leftarrow m+1$
    - Until $m = M_k$
- $k \leftarrow k+1$
- Until stopping criterion is met"[29]

So all the iterations from $M_0 + M_1 + \ldots + M_k$ are executed and k corresponds to the value for t_k where the criterion is met. Additionally, if $M_k = 1$ for all k, the temperature decreases at each iteration via zero[29]

13.3.5. GRASP: Greedy randomized adaptive search procedures

The GRASP approach is a multi start or iterative algorithm for combinatorial problems. Each iteration persists of a construction heuristic and a local search. By the means of the construction a good and accepted solutions will be generated and is, in further consequence, optimized through the use of a local search algorithm as long as a local minimum is reached. The best improvement achieved within the local search phase will be accumulated as result.

As to implement the existing algorithm the problem which will be optimized needs to consist of a finite ground set E, a minimizing/maximizing objective function and a set of feasible solutions. In our subjacent issue the finite ground set is described as the available amount of lots that need to be assigned to an established number of machines. The objective function

displays the main target of minimizing the entire utilized time for processing all lots on machines. The construction algorithm beforehand generates a set of feasible solutions which are improved in the second phase of the GRASP metaheuristic[29]

```
procedure GRASP(Max_Iterations,Seed)
1    Read_Input();
2    for k = 1,...,Max_Iterations do
3        Solution ← Greedy_Randomized_Construction(Seed);
4        Solution ← Local_Search(Solution);
5        Update_Solution(Solution,Best_Solution);
6    end;
7    return Best_Solution;
end GRASP.
```

Figure 27: Pseudocode GRASP metaheuristic[29]

The results which are achieved by the greedy randomized construction are not necessarily optimal. Therefore the local search is applied in order to gain even better or optimal results. In each iteration, the local search algorithm performs a successive replacement of the constructive solution by a superior result in the neighborhood of the current solution. The GRASP algorithm is terminated as no better solutions are generated in the neighborhood.

```
procedure Local_Search(Solution)
1    while Solution is not locally optimal do
2        Find s' ∈ N(Solution) with f(s') < f(Solution);
3        Solution ← s';
4    end;
5    return Solution;
end Local_Search.
```

Figure 28: A pseudocode of a local search with the constructive solution (Solution)[29]

The quality of the GRASP algorithm depends on several aspects. First of all it is reliant on the neighborhood structure, as well its search procedure. The second important aspect is the prompt evaluation of the cost function and finally it depends on the starting constructive solution itself. As the constructive phase of the GRASP algorithm is highly important for its further improvement, it is advisable to choose a good algorithm for generating high quality starting solutions for subsequent local search. To

achieve fast and good results, the neighborhood search can be implemented utilizing either a first improvement or a best improvement approach. The first improvement approach implies that the current solution steps onward to the first neighbor whose cost function value is lower in comparison to the current solution value, whereas the best improvement strategy replaces the current solution through the best neighbor in the neighborhood. Alike the Tabu Search, the GRASP algorithm accepts also inferior solution qualities to achieve in further consequences superior results for the underlying optimization issue[29]

14. Implementation

In this part of the work, the focus is on the practical realization of the previously discussed theory. Primary, the work attempts to tackle our two subproblems by means of the exact solution algorithm coded in Xpress using Xpress-IVE version 1.19.01, Xpress Mosel Version 2.4.0, Xpress Optimizer Version 19.00.00 and a HP computer with an AMD Athlon (tm) 64 processor. The comprehensive codes of both issues are headed in appendix A. Further elucidations towards the results and the computing times of the product mix issue as well of the material flow problem are expressed in the chapter computational results. Appendix B registers all corresponding data files for either issue.

Furthermore, the material flow problem is attempted to be optimized with the aid of previously explained heuristics. Thus, several discussed approaches of heuristics and metaheuristics will be implemented through the software system Microsoft Visual C++ Express Edition Version 9.0.30729.1 SP and will be calculated by a HP computer with an AMD Turion 64^{x2} processor. The corresponding codes will also be headed in appendix A. In appendix B relevant data sets will be listed.

15. Computational results of the Product Mix Issue

This model underlies a huge rang of complexity, which implies that only Xpress is utilized to implement and optimize it. The instances for the product mix issue are either provided by MIMAC [70] or self-generated. The latter ones are stated in Appendix B.

For the evaluation of the model the thesis distinguishes three different scenarios, whereas the demand remains the same in all of them. In the first case, demand and upper bound demand equal each other. In the second scenario, upper bound demand is 1,2 times demand and in the last run, upper bound demand stays 1,2 times demand but now only integer values are generated.
Since the run time always amounts close to zero, only Xpress is used to solve this issue.

Demand=Upper bound demand:
Profit is maximized to **5540.97** Euro while the capacity limit will not be achieved. The produced quantity equals the required demand level.

Upper bound demand=1,2*demand:
In this scenario, profit gets higher, up to **6220.97** Euro. Some of the toolgroups now reach their capacity bound, as shown in the figure below. Taking products 4 and 9 only the demand can be manufactured. Products 1, 2, 5, 6, 7, 8 reach their maximum production border, whereas an amount between these two bounds of product 3 is processed.

Upper bound demand=1,2*demand: Integer
In the last case, the profit amounts at **5929** Euro and the entire products are fabricated between their minimum and their maximum.

The subsequent figure points out the toolgroups with no remaining capacity in the second scenario. As it is evident, in case one the capacity is never reached and in the last case the utilization of several toolgroups attains the border.

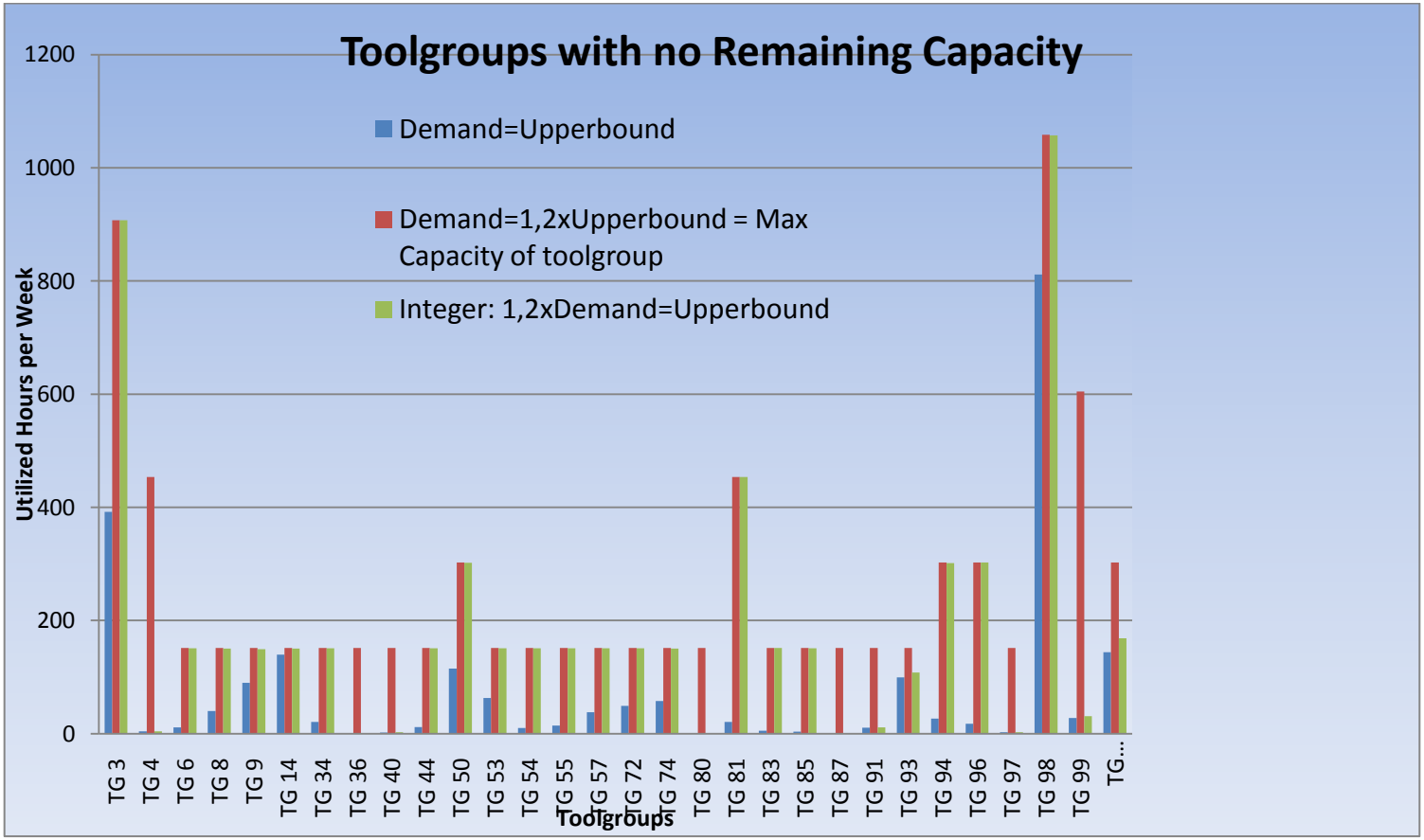

Figure 29: Toolgroups with no remaining capacity in scenario 2

The second illustration displays the occupancy in percent.

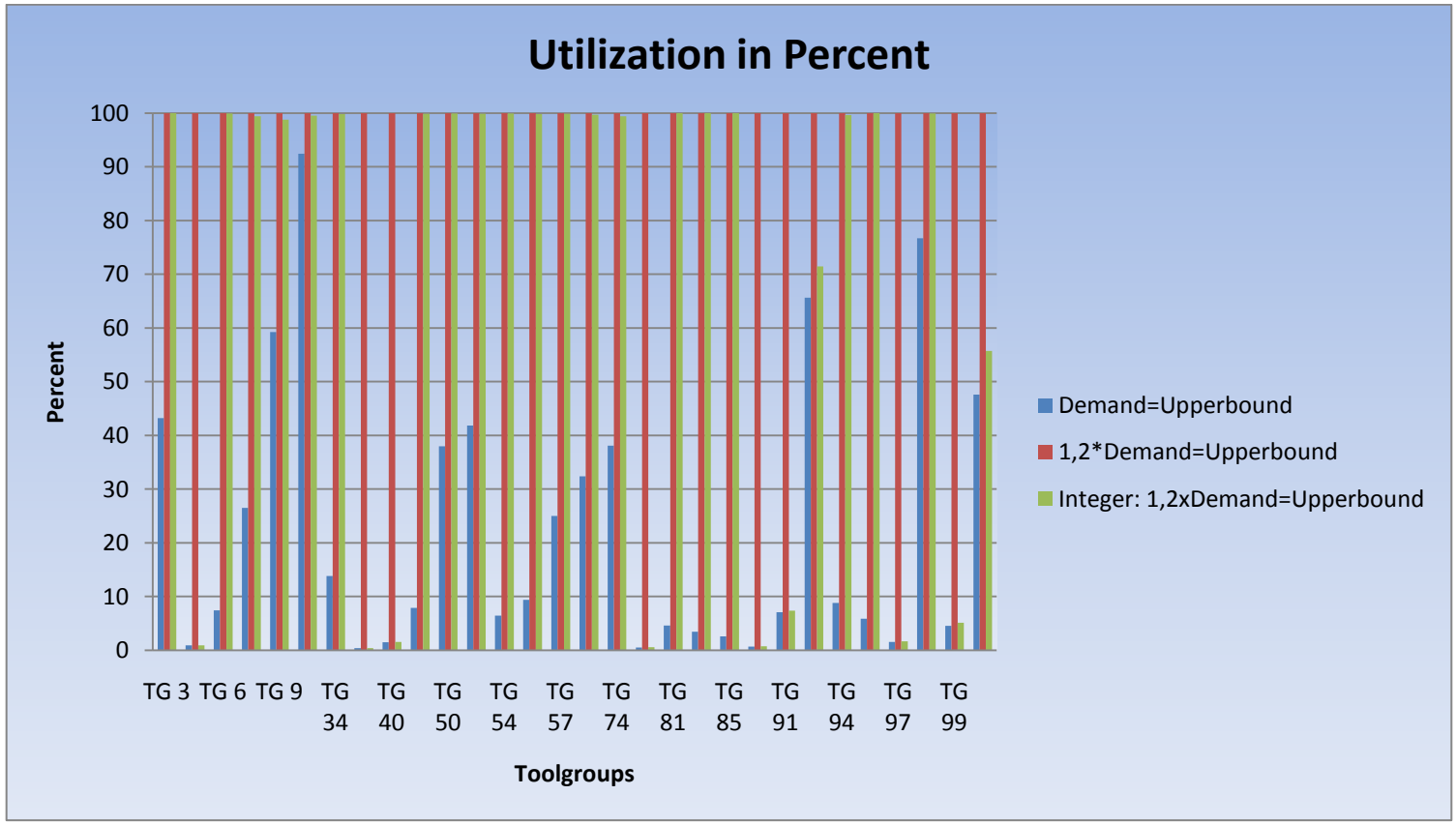

Figure 30: Utilization in Percent

16. Computational Results of the Material Flow Issue

In the first phase of evaluation, the material flow issue was implemented in Xpress by using data sets which are headed in Appendix B. This optimization program enables generating exact solutions up to 5 lots within reasonable computation time. Surprisingly, the software can only accomplish optimal results pending 15 lots. At this level of instances, the completion time rises sky-high.

In order to attempt to enhance the limit of complexity, the work established an abbreviated mathematical model. Also this minimization of decision variables cannot increase the level of solution quality. In contrast, this model exhibits at an even lower limit. The figure below outlines this fast growth in calculating time.

	Exact	
	Normal	Abbreviated
1	0,3 s	0,3 s
2	0,2 s	0,2 s
3	0,2 s	0,3 s
4	0,4 s	0,8 s
5	0,7 s	3,1 s
10	265,0 s	5317,2 s
15	3866,3 s	

Figure 31: Computation time of exact and abbreviated model

As the exact solution method is not able to solve the problem with all of the given data sets, the material flow issue is also coded within Visual Studio C++ by the means of heuristic and metaheuristic approaches. In the first step of the implementation, the work provides a First-In-First-Out priority rule procedure. These generated results obtain a worse quality level compared with the exact results but a better performance concerning the available instances. Generally, the solutions found by heuristics exhibit good results but not optimal ones. Furthermore, the work coded an improvement heuristic called 2-opt. By means of this approach, the outcome gets even better but still cannot reach the exact solution quality. In the last step of evaluation, the model is optimized by using a metaheuristic method named Greedy Randomized Adaptive Search Procedure. This approximation method obtains results that are nearly comparable to the exact solutions by Xpress. Solely at an amount of one lot, the metaheuristic exhibits the identical solution as Xpress. These performance of results stated below may even be optimized

by accomplishing further iterations and besides may attain the exact solutions originated in Xpress.

In general, heuristics can find good results for the subjacent material flow problem but solely metaheuristics enable generating solutions approximately to the optimum.

	Quality of results			
	Exact	FIFO	2opt	randomized
1	59,2072	60,6818	60,6818	59,2072
2	59,6885	64,1106	64,1106	60,3391
3	60,1920	65,2325	65,2325	63,6595
4	61,4967	68,4614	68,4614	63,8608
5	62,5981	69,7137	69,7137	65,7593
10	68,2558	90,7807	83,1182	81,2904
15	78,4026	101,5070	97,6072	94,4706
25		131,3950	123,7830	120,165
35		169,7840	163,3920	158,6800
45		203,7070	203,7070	201,516
55		239,7340	239,7340	236,6150

Figure 32: Comparison of total makespans

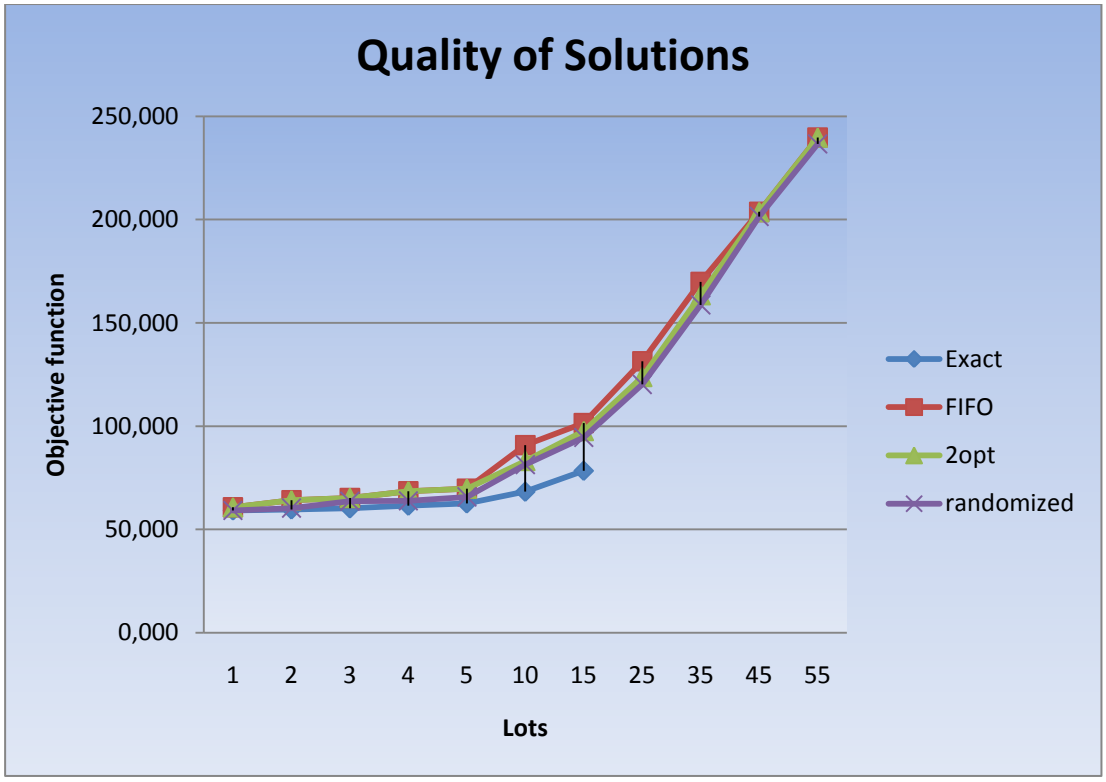

Figure 33: Quality of results

The figures below outline distinctions in the computation time among all used approaches. By channel in 10 lots, Xpress runs nearly 5 minutes, whereas the heuristics require a run time lower than one second, except the metaheuristic completes 100 iterations within two hours. All other solutions of

the remaining approaches are generated in only one iteration. Since the run time at the level of 10 lots is that high, the number of iterations is reduced by half. This abbreviation results in a lower completion time but nevertheless exhibits better solutions than other implemented heuristics.

	Computation Time				
	Exact	FIFO	2opt	randomized	Iterations
1	0,3	0,25	0,015	6,046	100
2	0,2	0,31	0,078	50,093	100
3	0,2	0,015	0,031	170,64	100
4	0,4	0,046	0,078	398,218	100
5	0,7	0,015	0,031	779,488	100
10	265	0,015	0,109	6943,160	100
15	3866,3	0,015	0,312	1308,520	50
25		0,015	1,203	6087,660	50
35		0,031	2,328	2163,860	30
45		0,031	3,796	3467,030	30
55		0,031	5,753	10163,200	30

Figure 34: Comparison of computation time

Conclusion

In times of the financial and economic crises, it is generally essential to reduce costs and production time, respectively to maximize the corporate profit. This work attempts to achieve this target while optimizing two different problems of a semiconductor manufacturing company.

On the one hand, the work attempted to maximize the overall profit value by illustrating three different scenarios. All of them exhibit rather good results while observing all constraints.

On the other hand, the thesis deals with a material flow problem which is aiming at minimizing the total makespan within one given work center. The exact solution method utilized in Xpress obtains indeed optimal results but solely for small data sets. Therefore, bigger instances are solved with heuristics and a metaheuristic in Visual Studio C++ which display suboptimal but feasible results.

This work provides a good basis for future research and optimization. First of all, the product mix problem offers the possibility to run the model with bigger instances. Secondly, the existing target may be replaced by other objectives functions, like maximizing the production quantity by using all of the available capacity, reducing costs while fabricating the products or adjusting the workload of the toolgroups. Furthermore, the material flow problem provides a good fundament for further implementation of diverse metaheuristics, like Tabu Search, Simulated Annealing and Variable Neighborhood Search. Finally, also the construction heuristic may be improved by using Cheapest Insertion, Nearest Neighbor or other Priority Rules.

Although the semiconductor industry underlies fast growing technological and economical challenges and changes, the subjacent models are adaptive to any kind of current and future requirements.

Bibliography

[1] Bell System Computer Journal 44 , S. 2245-2269.

[2] Barahona, F., Bermon, S., Günlük, O., & Hood, S. (2001). *Robust Capacity Planning in Semiconductor Manufacturing.* Report RC22196, IBM: T.J. Watson Research.

[3] Barnhart, C., & Laporte, G. (2007). Transportation, Handbooks in operation research and managment science. *14th Volume* , North-Holland.

[4] Baudin, M. (1990). *Manufacturing Systems Analysis with Application to Production Scheduling.* Yourdon Press Computing Series.

[5] Bermon, S., & Hood, S. (1999). Capacity Optimization Planning System. *Interfaces* , 31-50.

[6] Blazewicz, J., Ecker, K., Pesch, E., Schmidt, G., & Weglarz, J. (2007). *Handbook on Scheduling from Theory to Application, International Handbooks on Information Systems.* Berlin-Heidelberg-New York: Springer-Verlag.

[7] Bodin, L., Golden, B., Assad, A., & Ball, M. (1983). Routing and scheduling of vehicles and crews the state of the art. *Computers & Operations Research 10* , 63-211.

[8] Bradely, J. R., & Blossom, A. P. (2002). *Using Product-Mix Flexibility to Implement a Made-To-Order Assembly Line.* Ithaca, NY: The S.C. Johnson Graduate School of Managment.

[9] Chen, J., Chang, P., Chen, B., Chen, C., Chen, C., & Huang, S. (2003). *Application of Vehicle Routing Problem with Hard Time Window Constraints.* Hanoi: International conference on high performance scientific computing: Modelling, simulation and optimization of complex processes.

[10] Chou, Y.-C., & Hong, L.-H. (Vol. 13, Issue 3. August 2000). A Methodology for product mix planning in semiconductor foundry

manufacturing. *IEEE TRANSACTIONS ON SEMICONDUCTOR MANUFACTURING* , S. 278-285.

[11] Cleanroom. Abgerufen am 20. 06 2009 von Wikipedia: http://en.wikipedia.org/wiki/Cleanroom

[12] Cleanroom Classes. Abgerufen am 26. 06 2009 von Modularcleanrooms: http://www.modularcleanrooms.com/class.html

[13] Comneon Homepage. Abgerufen am 03. 08 2009 von About Us: http://www.comneon.com/CDA/about_comneon,11,0,,en.html (03.08.2009)

[14] Company Folder. (02 2009). Abgerufen am 16. 06 2009 von Infineon Technologies AG: http://www.infineon.com/dgdl?folderId=db3a3043132679fb01132ac168d701f5&fileId=db3a3043132679fb01133f37f72c08b1

[15] Company Homepage. Abgerufen am 16. 06 2009 von Infineon Technologies AG: http://www.infineon.com and www.infineon.at

[16] Company Presentation. Abgerufen am 16. 06 2009 von Infineon Technologies AG: http://www.infineon.com/cms/en/corporate/company/index.html

[17] Dakin, R. (1965). A tree search algorithm for mixed integer programming problems. *The Computer Journal, Volume 8* , 250-255.

[18] DeLeon, M. *A Study of Sufficient Conditions for Hamilton Cycles.* South Orange, New Jersey, New Jersey, USA: Department of Mathematics and Computer Science, Seton Hall University.

[19] Domschke, W. & Scholl, A. (2006). *Heuristische Verfahren.* Jenaer Schriften zur Wirtschaftswissenschaften.

[20] Domschke, W., & Drexl, A. (2005, 6. Auflage). *Einführung in Operations Research.* Berlin, Heidelberg, New York: Springer-Verlag.

[21] Dorigo, M., & Stützle, T. (2004). *Ant Colony Optimization, Bradford Books.* MIT Press.

[22] Dotieren. Abgerufen am 26. 06 2009 von Halbleitertechnologie von A bis Z: http://www.halbleiter.org/dotieren

[23] Dümmler, M., & Rose, O. (2000). *Analysis of the Short Term Impact of Changes in Product Mix.* Würzburg: Universität Würzburg, Institut für Informatik Research Report Series RR.

[24] Energy Efficency. Abgerufen am 17. 06 2009 von Infineon Technologies AG: http://www.infineon.com/cms/en/product/promopages/energy_efficiency/faq.html

[25] FAQs (Engergy Efficency). Abgerufen am 17. 06 2009 von Infineon Technologies AG: http://www.infineon.com/cms/en/product/promopages/energy_efficiency/faq.html

[26] Fowler, J. W., Hogg, G. L., & Mason, S. J. (No. 7. Vol. 13 2002). Workload Control in the Semiconductor Industry. *Production Planning & Control* , S. 568-578.

[27] Geng, N., & Jian, Z. (2008). A review on strategy capacity planning for the semiconductor manufacturing industry. *International Journal of Production Research* , 1-17.

[28] Glover, F. (No. 2. Volume 1 1987). Tabu Search Mehtods in Artificial Intelligence and Operations Research. *ORSA Artificial Intelligence* , S. 6.

[29] Glover, F., & Kochenberger, G. A. (2003). *Handbook of metaheuristics, Band 57 von Interantional series in operations research and management science.* Springer.

[30] Gmeilbauer, C. J. (2006). *Einsatz von Simulation zur Evaluierung von Prioritätsregeln im Scheduling der Halbleiterindustrie - Ein Fallbeispiel aus dem Bereich der Implanationstechnik am Produktionsstandort Infineon Villach.* Wien: Universität Wien, Betriebswirtschaftszentrum, Lehrstuhl für Produktion und Logistik.

[31] Halbleitertechnologie von A bis Z. Abgerufen am 26. 06 2009 von http://www.halbleiter.org/oxidation

[32] Hansen, P., & Mladenovic, N. (2003). Variable neighborhood search. In F. Glover, & G. A. Kochenberger, *Handbook of Metaheuristics* (S. 145-184). New York: Kluwer Academic Publishers.

[33] Hansen, P., & Mladenovic, N. (3. May 2001). Variable neighborhood search: Principles and applications. *European Journal of Operations Research 130* , S. 449-467.

[34] Hemmelmeyer, V. (WS 08/09). *Transportation Logistics.* Wien: Institut für Produktion und Logistik.

[35] Hilsenbeck, K. (2005). *Optimierungsmodelle in der Halbleiterproduktionstechnik.* München: Technische Universität München, Fakultät für Elektotechnik und Informationstechnik.

[36] Interview with Peter Bauer. Abgerufen am 16. 06 2009 von Infineon Technologies AG: http://www.infineon.com/cms/en/product/promopages/energy_efficiency/interview.html

[37] Jensen, T. (1996). *Planungsstabilität in der Material-Logistik.* Heidelberg: Physica-Verlag.

[38] Küttner, & Rein. (2004). INDUSTRIAL ENGINEERING - INNOVATION AS COMPETITIVE EDGE FOR SME. *4th International DAAAM-Conference*, (S. 276-278). Tallinn, Estonia.

[39] Levy, L. (2007). Private communication. RouteSmart Technologies, Inc. *Computers and Operations Research, Volume 34* , 2918-2930.

[40] Li, F., Golden, B., & Wasil, E. (2005). The open Vehicle Routing Prolbem: Algorithms, large-scale test problems, and computational results. *Computers & Operations Research 32 (5)* , 1165-1179.

[41] Liao, D., & Wang, C. (9-10. Volume 29 2006). Differenciated preemptive dispatching for automatic materials handling services in 300mm semiconductor foundry. *The International Journal of Advanced Manufacturing Technology* , S. 890-896.

[42] Liberti, L., & Drazic, M. (2005). Variable Neighbourhood Search for the Global Optimization of Constrained NLPs. *Proceeding of GO* , 1-5.

[43] Managing Innovation & Enterpreneurship. (Spring 2008). Abgerufen am 18. 06 2009 von Massachusetts Institute of Technology (MIT): http://ocw.mit.edu/NR/rdonlyres/Sloan-School-of-Management/15-351Spring-2008/BF95B0B6-91A4-4A66-924C-3929A8B2BC52/0/01_lecture.pdf

[44] Maniezzo, V., Gambardella, L. M., & de Luigi, F. (2004). Ant Colony Optimization. In A. Wesley, *Optimization Techniques in Engineering* (S. 101-117). Springer-Verlag.

[45] Material Used in Semiconductor Manufacturing. Abgerufen am 26. 06 2009 von Siliconfareast: http://www.siliconfareast.com/semicon_matls.htm

[46] Mladenovic, N. (2007). Variable Neighborhood Search, *Presentation.* City University, London, UK.

[47] Montoya-Torres, J. (2006). *Internal transport in automated semiconductor manufacturing systems Novel approaches for tactical and operational management.* PHD Thesis: Springer Verlag.

[48] Nasschemie. Abgerufen am 26. 06 2009 von Halbleitertechnologie von A bis Z: http://www.halbleiter.org/nasschemie

[49] Nemhauser, G., & Wolsey, L. (1999). *Integer and Combinatorial Optimization.* Canada: Willey-Interschience.

[50] O.Univ.-Prof.DI.Dr. Hartl, R. (2006). *ABWL Produktion & Logistik - Einführender Universitätskurs.* Wien: Lehrstuhl für Produktion und Logistik.

[51] O.Univ.-Prof.DI.Dr. Hartl, R., & Mag. Preusser, M. (2006). *Operations Management - Skriptum.* Wien: Universität Wien.

[52] O.Univ.-Prof.DI.Dr. Hartl, R., Univ.-Prof.Dr. Dawid, H., Dr. Dörner, K., Krippl, B., & Dr. Reimann, M. (WS 2005/06). *ABWL Produktion & Logistik - Vertiefender Kurs - Skriptum.* Universität Wien, Lehrstuhl für Produktion und Logistik.

[53] Ovacik, I., & Uzsoy, R. (1997). *Decomposition Methods for complex factory scheduling problems.* Bosten/Dordrecht/London: Kluwer Academic Publishers.

[54] Oxidation. Abgerufen am 26. 06 2009 von Halbleitertechnologie von A bis Z: http://www.halbleiter.org/oxidation

[55] Pillai, D. D., & Srinivasan, S. R. (1996). *Material Handling Automation - Trends, Version and Future Plans.* Tokyo, Japan: Preceedings from ISSM.

[56] Press Release. (31. March 2006). Abgerufen am 03. 08 2009 von Infineon Technologies AG: http://www.infineon.com/cms/en/corporate/press/news/releases/2006/175417.html

[57] Sagat, C. (WS 04/05). *Heuristiken für das Dial-a-Ride Problem.* Universität Wien.

[58] Schrage, L. (2006). Formulation and structure of more complex/realistic routing and scheduling problems. *Networks - An International Journal* , Vol.11, Issue 2, 229-232.

[59] Schulenberg, M. (2003). *Vom Transistor zum Maskenzentrum Dresden*. Abgerufen am 15. 06 2009 von http://www.bmbf.de/pub/vom_transistor_zum_maskenzentrum_dresden.pdf

[60] Semiconductor Era. Abgerufen am 18. 06 2009 von http://www.austincc.edu/HongXiao/overview/history-semi/sld005.htm

[61] Semiconductor Manufacturing. Abgerufen am 24. 06 2009 von Siliconfareast: http://www.siliconfareast.com/manufacturing.HTM

[62] Semiconductor Manufacturing. Abgerufen am 29. 06 2009 von Wikipedia: http://en.wikipedia.org/wiki/Semiconductor_manufacturing

[63] Tempelmeier, H. (1994, 3. Auflage). *Material-Logistik Grundlagen der Bedarfs- und Losgrößenplanung in PPS-Systemen.* Springer Verlag.

[64] Toth, P., & Vigo, D. (2002). *The Vehicle Routing Problem (Siam Monographs on Discrete Mathematics and Applications).* United States of America: Society for Industrial and Applied Mathematics.

[65] Trockenätzen. Abgerufen am 26. 06 2009 von Halbleitertechnologie von A bis Z: http://www.halbleiter.org/trockenätzen

[66] VRP Web. Abgerufen am 11. 09 2009 von http://neo.lcc.uma.es/radi-aeb/WebVRP/

[67] Zahn, E., & Schmid, U. (1996). *Produktionswirtschaft 1: Grundlagen und operatives Produktionsmanagement.* Stuttgart: UTB Stuttgart.

[68] Zäpfel, G. (2001). *Grundzüge des Produktions- und Logistikmanagement.* München: Oldenbourd Wissenschafts Verlag, 2. Auflage.

[69] Zäpfel, G., & Braune, R. (2005). *Moderne Heuristiken der Produktionsplanung, WiSo Kurzlehrbücher.* München: Vahlen, 1. Auflage.

Dataset Resource:

[70] MIMAC (*Measurement and Improvement of Manufacturing Capacities*) test bed; *Dataset 6*; the datasets are available via anonymous ftp from ftp.sematech.org, or by contacting John Fowler at SEMATECH or Ben Rodriguez at Nimble.

Appendix A – Models

1. Product Mix without Integer Variables

First of all, the product mix model was implemented without requiring any integer numbers. The code for Xpress is the following:

```
model "Product Mix"  uses "mmodbc", "mmxprs"
declarations

        PRODUCTS = 1..9
        JOBS = 1..323
        MACHINES = 1..104

        amount: array(PRODUCTS) of mpvar
        steps: dynamic array(JOBS, MACHINES) of mpvar
        producejob: array(JOBS, PRODUCTS) of integer
        processingtime: array(JOBS, MACHINES) of real
        demand: array(PRODUCTS) of real
        upperbounddemand: array (PRODUCTS) of real
        Sellingprice: array (PRODUCTS) of real
        capacity: array(MACHINES) of real
        numberoftools: array (MACHINES) of integer

end-declarations

!Modify Optimizer control parameter MIPTO
setparam("XPRS_MIPTOL",0)

! Read data from spreadsheet productmixdaten.xls
initializations from "mmodbc.odbc:ProductMixII.xls"
         producejob as "producejob"
        processingtime as "processingtime"
        demand as "demand"
         upperbounddemand as "upperbounddemand"
```

```
        Sellingprice as "sellingprice"
        capacity as "capacity"
        numberoftools as "numberoftools"
end-initializations

forall (i in PRODUCTS)
        create (amount(i))
finalize(PRODUCTS)

forall (j in JOBS, m in MACHINES| processingtime(j,m)< 1000000000)
        create (steps (j,m))
finalize (JOBS)
finalize (MACHINES)

Z:= sum(i in PRODUCTS) Sellingprice(i) *  amount(i) !objective function

forall (j in JOBS)
                sum(m in MACHINES) steps(j,m)>=sum(i in PRODUCTS)
producejob(j,i) * amount(i)

forall(m in MACHINES)               !capacity constraint
                testNB(m):=sum(j  in  JOBS|  processingtime(j,m)<1000000000)
steps(j,m) * processingtime(j,m)<=capacity(m) * numberoftools(m)

forall(i in PRODUCTS) do
        demand(i)<= amount(i)
        amount(i) <= upperbounddemand(i) !demand constraint
end-do

maximize(Z)

writeln("Produced Amount:", getobjval)
forall(i in PRODUCTS|getsol(amount (i))> 0) do
      write(i, ": ")
      writeln( getsol(amount(i)) )
```

```
    end-do

!print output to run pane
writeln("Steps:", getobjval)
forall(j in JOBS, m in MACHINES|getsol(steps(j,m))>0) do
        write(j, ": ")
        write(m, ": ")
        writeln( getsol(steps(j,m)) )
    end-do

 end-model
```

1.1 Product Mix with Integer Variables

The second step was to implement the same model by requiring integer numbers as output. The only difference exists while creating the two variables as the produced amount and the steps needed to finish a job must be integer in real world.

```
model "Product Mix"  uses "mmodbc", "mmxprs"
declarations

        PRODUCTS = 1..9
        JOBS = 1..323
        MACHINES = 1..104

        amount: array(PRODUCTS) of mpvar
        steps: dynamic array(JOBS, MACHINES) of mpvar
        producejob: array(JOBS, PRODUCTS) of integer
        processingtime: array(JOBS, MACHINES) of real
        demand: array(PRODUCTS) of real
        upperbounddemand: array (PRODUCTS) of real
        Sellingprice: array (PRODUCTS) of real
        capacity: array(MACHINES) of real
        numberoftools: array (MACHINES) of integer
```

```
end-declarations

!Modify Optimizer control parameter MIPTO
setparam("XPRS_MIPTOL",0)

! Read data from spreadsheet productmixdaten.xls
initializations from "mmodbc.odbc:ProductMixII.xls"
        producejob as "producejob"
        processingtime as "processingtime"
        demand as "demand"
        upperbounddemand as "upperbounddemand"
        Sellingprice as "sellingprice"
        capacity as "capacity"
        numberoftools as "numberoftools"
end-initializations

forall (i in PRODUCTS) do
        create (amount(i))
        amount(i) is_integer
end-do
finalize(PRODUCTS)

forall (j in JOBS, m in MACHINES| processingtime(j,m)< 1000000000) do
        create (steps (j,m))
        steps(j,m) is_integer
end-do
finalize (JOBS)
finalize (MACHINES)

Z:= sum(i in PRODUCTS) Sellingprice(i) *  amount(i) !objective function

forall (j in JOBS)
                sum(m in MACHINES) steps(j,m)>=sum(i in PRODUCTS)
producejob(j,i) * amount(i)
```

```
forall(m in MACHINES)                !capacity constraint
                  testNB(m):=sum(j  in  JOBS|  processingtime(j,m)<1000000000)
steps(j,m) * processingtime(j,m)<=capacity(m) * numberoftools(m)

forall(i in PRODUCTS) do
        demand(i)<= amount(i)
        amount(i) <= upperbounddemand(i) !demand constraint
end-do

maximize(Z)

writeln("Produced Amount:", getobjval)
forall(i in PRODUCTS|getsol(amount (i))> 0) do
      write(i, ": ")
      writeln( getsol(amount(i)) )
   end-do

!print output to run pane
writeln("Steps:", getobjval)
forall(j in JOBS, m in MACHINES|getsol(steps(j,m))>0) do
      write(j, ": ")
      write(m, ": ")
      writeln( getsol(steps(j,m)) )
   end-do

 end-model
```

2. Material Flow Model

The underlying Xpress code represents the basic model which was used to compute meaningful results within reasonable calculating time.

```
model "Material Flow" uses "mmodbc", "mmxprs"
```

```
!model encompasses 5 machines, 55 lots and 10 lottypes
declarations

        MACHINES= 1..5
        LOTS= 1..55
        LOTTYPE= 1..10

        lots: array(LOTS, MACHINES) of mpvar
        setup: array(LOTS,LOTS,MACHINES) of mpvar
        startlot: array(LOTS,MACHINES) of mpvar
        initial: array(LOTS,MACHINES) of mpvar
        final: array (LOTS, MACHINES) of mpvar
        Z_m: array (MACHINES) of mpvar
        makespan: mpvar


        PROCESSINGTIME: array(LOTTYPE,MACHINES) of real
        SETUPTIME: array(LOTTYPE,LOTTYPE)of real
        RELEASETIME: array(LOTS)of real
        MACHINERELEASE: array(MACHINES)of real
        TYPE: array(LOTS) of integer
        INITIAL: array(MACHINES,LOTTYPE)of real
        FINAL: array (LOTTYPE,MACHINES)of real

end-declarations

!Read data from spreadsheet MaterialFlowNeu.xls
initializations from "mmodbc.odbc:MaterialFlowNEU.xls"
        PROCESSINGTIME as "processingtime"
        SETUPTIME as "setuptime"
        RELEASETIME as "releasetime"
        MACHINERELEASE as "machinerelease"
        TYPE as "lottype"
        INITIAL as "initialtime"
        FINAL as "finaltime"
```

```
end-initializations

forall (i in LOTS, m in MACHINES)do
        create (lots(i,m))
        lots(i,m)is_integer
finalize (LOTS)
finalize(MACHINES)
        end-do

forall (i,j in LOTS, m in MACHINES) do
        create (setup(i,j,m))
        setup(i,j,m) is_binary
finalize (LOTS)
finalize (MACHINES)
        end-do
forall (i in LOTS, m in MACHINES)
        create (startlot(i,m))
finalize (LOTS)
finalize (MACHINES)

forall (i,j in LOTS, m in MACHINES) do
        create (initial(i,m))
        initial(i,m) is_binary
finalize (LOTS)
finalize (MACHINES)
        end-do

forall (i in LOTS, m in MACHINES) do
        create (final(i,m))
        final(i,m) is_binary
finalize (LOTS)
finalize (MACHINES)
        end-do

forall (m in MACHINES)
```

```
        create (cycletime(m))
finalize (MACHINES)

!declaration of objective function

Z:= sum(m in MACHINES) Z_m (m) + makespan*5

forall(i in LOTS,m in MACHINES)
        Z_m(m)+ (1-lots(i,m))*1000  >=  startlot(i,m) +
        PROCESSINGTIME(TYPE(i),m) + final (i,m)*FINAL(TYPE(i),m)

!each lot is assigned
forall(i in LOTS)
        sum(j in LOTS,m in MACHINES|i<>j) setup(i,j,m) + sum(m in
        MACHINES) final(i,m)  =  1

!either initial lot or not
forall(m in MACHINES)
        sum(i in LOTS) initial(i,m) <= 1

!either final lot or not
forall(m in MACHINES)
        sum(i in LOTS) final(i,m) <= 1

!flow conservation
forall(i in LOTS,m in MACHINES)
        final(i,m) + sum(j in LOTS|i<>j) setup(i,j,m)  = initial(i,m) + sum (j in
        LOTS|i<>j) setup(j,i,m)

!essential setup within lot i and j
forall(i,j in LOTS|i<>j,m in MACHINES)
        startlot(i,m)+ PROCESSINGTIME(TYPE(i),m)+
        setup(i,j,m)*SETUPTIME(TYPE(i),TYPE(j))<= startlot(j,m)+(1-
        setup(i,j,m))*1000
```

```
!auxilliary variable lots(i,m)
forall(i in LOTS,m in MACHINES)
        lots(i,m) = sum(j in LOTS|i<>j) setup(i,j,m) + final(i,m)

!earliest starting time for machine m
forall (i in LOTS,m in MACHINES)
        MACHINERELEASE(m)* lots(i,m) + INITIAL(m,TYPE(i))*initial(i,m) <=
        startlot (i,m)

!arrival of lot i
forall(i in LOTS,m in MACHINES)
        RELEASETIME(i)*lots(i,m) <= startlot(i,m)

!Z_m on each machine is at least greater than MACHINERELEASE
forall(m in MACHINES)
                Z_m(m) >=MACHINERELEASE(m)

!identifies the highest makespan on machine m
forall(m in MACHINES)
                makespan >= Z_m(m)

!objective function ought to be minimized
minimize (Z)

!print output to run pane
!objective function
writeln("Makespan: \t")
        forall (m in MACHINES| getsol (Z_m(m))>0) do
                write (m, ":")
                writeln("\t", getsol(Z_m(m)) )
        end-do

!print output to run pane
!decision variable
writeln("Assignment of lot i to machine m: \t")
```

```
        forall(i in LOTS, m in MACHINES| getsol(lots(i,m))>0) do
                write(i, ": ")
                write(m, ": ")
                writeln("\t", getsol(lots(i,m)) )
        end-do

!print output to run pane
!decision variable
writeln("Essential Setups: \t")
forall(i,j in LOTS, m in MACHINES| getsol(setup(i,j,m))>0) do
                write(i, ": ")
                write(j, ": ")
                write(m, ": ")
                writeln("\t", getsol(setup(i,j,m)) )
        end-do

!print output to run pane
!decision variable
writeln("Startingtime of lot i on machine m: \t")
        forall(i in LOTS, m in MACHINES| getsol(startlot(i,m))>0) do
                write(i, ": ")
                write(m, ": ")
                writeln("\t", getsol(startlot(i,m)) )
        end-do

!print output to run pane
!decision variable
writeln("Inital Setup of lot i on machine m: \t")
        forall(i in LOTS, m in MACHINES| getsol(initial(i,m))>0) do
                write(i, ": ")
                write(m, ": ")
                writeln("\t", getsol(initial(i,m)) )
        end-do

!print output to run pane
```

```
!decision variable
writeln("Final Setup of lot i on machine m: \t")
        forall(i in LOTS, m in MACHINES| getsol(final(i,m))>0) do
                write(i, ": ")
                write(m, ": ")
                writeln("\t", getsol(final(i,m)) )
        end-do

end-model
```

2.1 Abbreviated Material Flow Model

Without the utilization of the decision variable lots(i,m), the thesis attempts to obtain results with an even better computing time. Through this omission the model reduces its complexity as the size of loops and as well the number of variables abates. Therefore, the calculations of the objective function and even of the decision variables should be accomplished within less computing time. Furthermore, lots(i,m) will be replaced by decision variable final(i,m) which is already existent in the model.

```
model "Material Flow" uses "mmodbc", "mmxprs"

!model encompasses 5 machines, 55 lots and 10 lottypes
declarations

        MACHINES= 1..5
        LOTS= 1..55
        LOTTYPE= 1..10

        !variable lots(i,m) won´t be consulted in this model
        !lots: array(LOTS, MACHINES) of mpvar
        setup: array(LOTS,LOTS,MACHINES) of mpvar
        startlot: array(LOTS,MACHINES) of mpvar
        initial: array(LOTS,MACHINES) of mpvar
        final: array (LOTS, MACHINES) of mpvar
```

```
        Z_m: array (MACHINES) of mpvar
        makespan: mpvar

        PROCESSINGTIME: array(LOTTYPE,MACHINES) of real
        SETUPTIME: array(LOTTYPE,LOTTYPE)of real
        RELEASETIME: array(LOTS)of real
        MACHINERELEASE: array(MACHINES)of real
        TYPE: array(LOTS) of integer
        INITIAL: array(MACHINES,LOTTYPE)of real
        FINAL: array (LOTTYPE,MACHINES)of real
end-declarations

!Read data from spreadsheet MaterialFlowNeu.xls
initializations from "mmodbc.odbc:MaterialFlowNEU.xls"
        PROCESSINGTIME as "processingtime"
        SETUPTIME as "setuptime"
        RELEASETIME as "releasetime"
        MACHINERELEASE as "machinerelease"
        TYPE as "lottype"
        INITIAL as "initialtime"
        FINAL as "finaltime"
end-initializations

!since lots(i,m) is not declared, it does not have to be created
!forall (i in LOTS, m in MACHINES)do
        !create (lots(i,m))
        !lots(i,m)is_integer
!finalize (LOTS)
!finalize(MACHINES)
        !end-do

forall (i,j in LOTS, m in MACHINES) do
        create (setup(i,j,m))
        setup(i,j,m) is_binary
finalize (LOTS)
```

```
finalize (MACHINES)
        end-do

forall (i in LOTS, m in MACHINES)
        create (startlot(i,m))
finalize (LOTS)
finalize (MACHINES)

forall (i,j in LOTS, m in MACHINES) do
        create (initial(i,m))
        initial(i,m) is_binary
finalize (LOTS)
finalize (MACHINES)
        end-do

forall (i in LOTS, m in MACHINES) do
        create (final(i,m))
        final(i,m) is_binary
finalize (LOTS)
finalize (MACHINES)
        end-do

forall (m in MACHINES)
        create (Z_m(m))
finalize (MACHINES)

!declaration of makespan

Z:= sum (m in MACHINES) Z_m(m)+ makespan*5

forall(i in LOTS,m in MACHINES)
        Z_m(m) + (1-final(i,m))*1000  >=  startlot(i,m) +
        PROCESSINGTIME(TYPE(i),m) + final(i,m)*FINAL(TYPE(i),m)

!each lot is assigned
```

```
forall(i in LOTS)
        sum(j in LOTS,m in MACHINES|i<>j) setup(i,j,m) + sum(m in
        MACHINES) final(i,m) = 1

!either initial lot or not
forall(m in MACHINES)
        sum(i in LOTS) initial(i,m) <= 1

!either final lot or not
forall(m in MACHINES)
        sum(i in LOTS) final(i,m) <= 1

!flow conservation
forall(i in LOTS,m in MACHINES)
        final(i,m) + sum(j in LOTS|i<>j) setup(i,j,m)  = initial(i,m) + sum (j in
        LOTS|i<>j) setup(j,i,m)

!essential setup within lot i and j
forall(i,j in LOTS|i<>j,m in MACHINES)
        startlot(i,m)+ PROCESSINGTIME(TYPE(i),m)+
        setup(i,j,m)*SETUPTIME(TYPE(i),TYPE(j))<= startlot(j,m)+(1-
        setup(i,j,m))*1000

!unnecessary as lots(i,m) is not declared
!auxilliary variable lots(i,m)
!forall(i in LOTS,m in MACHINES)
        !lots(i,m) = sum(j in LOTS|i<>j) setup(i,j,m) + final(i,m)

!earliest starting time for machine m
forall (i in LOTS,m in MACHINES)
        MACHINERELEASE(m)*(setup(i,j,m)+ final(i,m) )+
        INITIAL(m,TYPE(i))*initial(i,m) <=  startlot (i,m)

!arrival of lot i
forall(i,j in LOTS,m in MACHINES)
```

```
        RELEASETIME(i)*(setup(i,j,m)+ final(i,m) ) <= startlot(i,m)

Z_m on each machine is at least greater than MACHINERELEASE
forall(m in MACHINES)
                Z_m(m) >=MACHINERELEASE(m)

!identifies the highest makespan on machine m
forall(m in MACHINES)
                makespan >= Z_m(m)

!objective function ought to be minimized
minimize (Z)

!print output to run pane
!objective function
writeln("Makespan: \t")
        forall (m in MACHINES|getsol (Z_m (m))>0) do
                write(m, ":")
                writeln("\t", getsol(Z_m(m)) )
        end-do

!unnecessary as lot(i,m) is not declared
!print output to run pane
!writeln("Assignment of lot i to machine m: \t")
        !forall(i in LOTS, m in MACHINES) do
                !write(i, ": ")
                !write(m, ": ")
                !writeln("\t", getsol(lots(i,m)) )
        !end-do

!print output to run pane
!decision variable
writeln("Essential Setups: \t")
forall(i,j in LOTS, m in MACHINES| getsol(setup(i,j,m))>0) do
                write(i, ": ")
```

```
                write(j, ": ")
                write(m, ": ")
                writeln("\t", getsol(setup(i,j,m)) )
        end-do

!print output to run pane
!decision variable
writeln("Startingtime of lot i on machine m: \t")
        forall(i in LOTS, m in MACHINES| getsol(startlot(i,m))>0) do
                write(i, ": ")
                write(m, ": ")
                writeln("\t", getsol(startlot(i,m)) )
        end-do

!print output to run pane
!decision variable
writeln("Inital Setup of lot i on machine m: \t")
        forall(i in LOTS, m in MACHINES| getsol(initial(i,m))>0) do
                write(i, ": ")
                write(m, ": ")
                writeln("\t", getsol(initial(i,m)) )
        end-do

!print output to run pane
!decision variable
writeln("Final Setup of lot i on machine m: \t")
        forall(i in LOTS, m in MACHINES| getsol(final(i,m)) )>0) do
                write(i, ": ")
                write(m, ": ")
                writeln("\t", getsol(final(i,m)) )
        end-do

end-mode
```

Appendix B – Data Sets

Product Mix Issue

Demand	upperbounddemand	selllingprice	capacity
1,80687516	2,168250192	101	151,2
3,824918891	4,589902669	103	151,2
4,612419489	5,534903387	105	151,2
15,37461907	18,44954288	102	151,2
9,749529933	11,69943592	108	151,2
3,525086868	4,230104242	100	151,2
4,08741223	5,108739	105	151,2
4,76233672	5,714804064	107	151,2
5,512498277	6,614997932	103	151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2
			151,2

151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2
151,2

Material Flow Issue

	releasetime	lottype
1	0	3
2	0,540652896	5
3	1,138852381	4
4	1,455394888	7
5	1,947216727	1
6	2,656644154	6
7	2,932350025	1
8	3,171263195	7
9	3,706863958	1
10	4,157158347	2
11	4,9460011	8
12	5,016279641	2
13	5,317665236	8
14	5,773233059	6
15	6,339706949	10
16	7,072234613	2
17	7,599234296	6
18	8,064793936	5
19	8,529512701	10
20	8,74097878	10
21	9,004156487	9
22	9,52456179	9
23	10,08093893	10
24	10,32556247	2
25	10,68883045	10
26	11,12827608	9
27	11,6356799	7
28	12,11044322	6
29	12,72309166	10
30	13,3678865	3
31	13,68117987	2
32	13,83241041	6
33	14,39823268	6
34	14,92821848	8
35	15,58784599	1
36	15,99338094	1
37	16,03096917	6
38	16,39736055	2
39	16,8839223	9
40	17,33926756	1
41	17,55897126	4
42	18,28086164	2
43	18,45280375	2
44	19,14949271	8
45	19,56571107	7
46	19,94904887	10
47	20,20111366	1
48	20,20965141	5
49	20,6304037	6
50	21,13414568	10
51	21,86469127	2
52	22,29829974	7
53	22,51205858	6
54	23,0592911	6
55	23,08431279	4

machinerelease	
1	0,5
2	6
3	0
4	8
5	3,4

processingtime	Machine 1	Machine 2	Machine 3	Machine 4	Machine 5
lottype 1	1,290409444	1000	1000	1,317231005	1,252314822
lottype 2	1,056271826	1000	1000	1000	1000
lottype 3	1,197781311	1,299305751	1,307174898	1,258400586	1,370042205
lottype 4	0,382748097	0,344147457	0,408013504	0,356765643	0,399102334
lottype 5	0,425286713	1000	1000	0,39934594	0,408847187
lottype 6	0,605257085	0,54198281	0,566357588	0,592672037	0,578295196
lottype 7	1000	0,802047632	0,798921832	0,737097208	1000
lottype 8	1000	1000	1,161424759	1,176452777	1000
lottype 9	1000	0,590007762	0,598651119	0,612973907	0,532468513
lottype 10	1,49734915	1,505377642	1,607388586	1,478909624	1,452358213

final	machine 1	machine 2	machine 3	machine 4	machine 5
lottype1	0,00000000	0,00000000	0,00000000	0,62707173	0,00000000
lottype2	0,00000000	0,69956749	0,00000000	0,27827672	0,00000000
lottype3	0,83845367	0,00977321	0,00000000	0,00000000	0,00000000
lottype4	0,00000000	0,73866835	0,97611146	0,81804264	0,00000000
lottype5	0,60313540	0,00000000	0,00000000	0,00000000	0,00000000
lottype6	0,00000000	0,00000000	0,00000000	0,00000000	0,63638817
lottype7	0,03905590	0,50260062	0,00000000	0,00000000	0,37465186
lottype8	0,00000000	0,25086417	0,47990304	0,97453756	0,00000000
lottype9	0,00000000	0,17213084	0,00000000	0,34071769	0,00000000
lottype10	0,00000000	0,46324774	0,62035380	0,00000000	0,00000000

setup [h]	lottype 1	lottype 2	lottype 3	lottype 4	lottype 5	lottype 6	lottype 7	lottype 8	lottype 9	lottype 10
lottype1	0,00000000	0,00000000	0,22103663	0,00000000	0,00000000	0,00000000	0,62707173	0,00000000	0,35529856	0,00000000
lottype2	0,00000000	0,00000000	0,91959805	0,00000000	0,69956749	0,00000000	0,27827672	0,00000000	0,53154960	0,85158521
lottype3	0,00000000	0,00000000	0,00000000	0,83845367	0,00977321	0,00000000	0,00000000	0,00000000	0,06238347	0,00000000
lottype4	0,38355615	0,79407105	0,10432681	0,00000000	0,73866835	0,97611146	0,81804264	0,00000000	0,14634894	0,80737724
lottype5	0,80554320	0,38892394	0,54989083	0,60313540	0,00000000	0,00000000	0,00000000	0,00000000	0,00000000	0,00000000
lottype6	0,20510534	0,22267089	0,74554980	0,00000000	0,00000000	0,00000000	0,00000000	0,63638817	0,51669518	0,00000000
lottype7	0,84900694	0,74114085	0,00000000	0,03905590	0,50260062	0,00000000	0,00000000	0,37465186	0,50658959	0,00000000
lottype8	0,00000000	0,11637756	0,00000000	0,00000000	0,25086417	0,47990304	0,97453756	0,00000000	0,00000000	0,00000000
lottype9	0,32764986	0,00000000	0,12408929	0,00000000	0,17213084	0,00000000	0,34071769	0,00000000	0,00000000	0,99607208
lottype10	0,00000000	0,00000000	0,00000000	0,00000000	0,46324774	0,62035380	0,00000000	0,00000000	0,00000000	0,00000000

initial	lottype1	lottype2	lottype3	lottype4	lottype5	lottype6	lottype7	lottype8	lottype9	lottype10
machine 1	0,20510534	0,22267089	0,74554980	0,00000000	0,00000000	0,00000000	0,00000000	0,63638817	0,51669518	0,00000000
machine 2	0,84900694	0,74114085	0,00000000	0,03905590	0,50260062	0,00000000	0,00000000	0,37465186	0,50658959	0,00000000
machine 3	0,00000000	0,11637756	0,00000000	0,00000000	0,25086417	0,47990304	0,97453756	0,00000000	0,00000000	0,00000000
machine 4	0,32764986	0,00000000	0,12408929	0,00000000	0,17213084	0,00000000	0,34071769	0,00000000	0,00000000	0,99607208
machine 5	0,00000000	0,00000000	0,00000000	0,00000000	0,46324774	0,62035380	0,00000000	0,00000000	0,00000000	0,00000000

Appendix C – Computational Results in detail

1. Product Mix Model

Demand = Upperbound **Profit: 5540.97**	**1,2*Demand = Upperbound** **Profit: 6220.97**	**1,2*Demand = Upperbound Integer** **Profit: 5929**
Produced Amount:	**Produced Amount:**	**Produced Amount:**
1: 1.80688	1: 2,16825	1: 2
2: 3.82492	2: 4,5899	2: 4
3: 4.61242	3: 5,32156	3: 5
4: 15.3746	4: 15,3746	4: 16
5: 9.74953	5: 11,6994	5: 10
6: 3.52509	6: 4,2301	6: 4
7: 4.08741	7: 5,10874	7: 5
8: 4.76234	8: 5,7148	8: 5
9: 5.5125	9: 5,5125	9: 6

2. Material Flow Model – Xpress Results

Number of Lots	Sum of makespan	Runtime	lots(i,m)	setup (i,j,m)	start(i,m)	initial(i,m)	final(i,m)
1	1->0.5	0.3 s	1->3			1->3	1->3
	2->6						
	3->1.30717						
	4->8						
	5-> 3.4						
	∑= 59.2072						
2	1->0.5	0.2 s	1->5	1->2: 5	3.4	1->5	
	2->6		2->5		4.77976		2->5
	3->0						
	4->8						
	5->5.18855						
	∑= 59.6885						
3	1->0.5	0.2s	1->5	1-> 2: 5	3.90343		
	2->6		2->5		5.28324		2->5
	3->0		3-> 5	3-> 1: 5	3.4	3->5	
	4->8						
	5->5.69209						
	∑= 60.192						

4	1->0.5	0.4 s	1->5	1->2: 5	3.90343		
	2->7.30465		2->5		5.28324		2->5
	3->0		3->5	3->1: 5	3.4	3->5	
	4->8		4->2		6	4->2	4->2
	5->5.69209						
	∑= 61.4967						
5	1->0.5	0.7 s	1->3	1-> 4: 3		1->3	
	2->6		2->5		5.43497		2->5
	3->2.25432		3->5	3>5: 5	3.4	3->5	
	4->8		4->3		1.45539		4->3
	5->5.84382		5->5	5->2: 5			
	∑= 62.5981		5->1	5->3: 1	4.18266		
10	1->7.25746	265.0 s	1->3	1->4: 3		1->3	
	2->7.34403		2->1	2->5: 1	0.540653	2->1	
	3->2.25432		3->1		6.87472		3->1
	4->8		4->3		1.45539		4->3
	5-> 3.4		5->1	5->7: 1	1.94722		
	Best bound:		6->2		6.80205		6->2
	68.249		7->1	7->10: 1	3.23763		
	Best solution:		8->2	8-> 6: 2	6	8->2	
	68.2558		9->1	9-> 3: 1	5.58431		
	Gap:		10->1	10-> 9: 1	4.52804		

	0.00996932%						
15	Best bound: 71.7967 Best solution: 78.4026 Gap: 8.42559%	3866.3 s	Limit of complexity				

2.1. Abbreviated Material Flow Problem

Number of Lots	Total Makespan	Runtime	setup (i,j,m)	start(i,m)	initial(i,m)	final(i,m)
1	1->0.5	0.3 s			1->3	1->3
	2->6					
	3->1.30717					
	4->8					
	5-> 3.4					
	∑= 59.2072					
2	1->0.5	0.2 s	1->2: 5	3.4	1->5	
	2->6			4.77982		2->5
	3->0					
	4->8					
	5->5.18866					
	∑= 59.6887					
3	1->0.5	0.3 s	1->2:5	3.90343		
	2->6			5.28324		2->5
	3->0		3->1: 5	3.4	3->5	
	4->8					
	5->5.69209					
	∑= 60.192			1.56908		3->1

4	1->0.5	0.8 s	1->2: 5	3.90343		
	2->7.30465			5.28324		2->5
	3->0		3->1: 5	3.4	3->5	
	4->8			6	4->2	4->2
	5->5.69209					
	∑= 61.4967					
5	1->0.5	3.1 s	1-> 4: 3		1->3	
	2->6			5.43497		
	3->2.25432		3-> 5: 5	3.4	3->5	
	4->8			1.45539		2->5
	5->5.84382		5-> 2: 5	4.18266		4->3
	∑= 62.5981					
10	Best bound: 63.2187 Best solution: 68.8582 Gap: 8.19003%	5317.2 s	Limit of complexity			

3. Material Flow Model – C++ Results

FIFO		**2-Opt**		**GRASP**	
1 Lot					
Makespan 1 : 3.28178 Makespan 2 : 6 Makespan 3 : 0 Makespan 4 : 8 Makespan 5 : 3.4 Total Makespan: 20.6818 8 Objective Function: 60.6818	Machine 1 : 1 Size:1 Machine 2 processes no lots Size:0 Machine 3 processes no lots Size:0 Machine 4 processes no lots Size:0 Machine 5 processes no lots Size:0	Makespan 1 : 3.28178 Makespan 2 : 6 Makespan 3 : 0 Makespan 4 : 8 Makespan 5 : 3.4 Total Makespan: 20.6818 8 Objective Function: 60.6818	Machine 1 : 1 Size:1 Machine 2 processes no lots Size:0 Machine 3 processes no lots Size:0 Machine 4 processes no lots Size:0 Machine 5 processes no lots Size:0	Makespan 1 : 0.5 Makespan 2 : 6 Makespan 3 : 1.30717 Makespan 4 : 8 Makespan 5 : 3.4 Total Makespan: 19.2072 8 Objective Function: **59.2072**	Machine 1 processes no lots Size:0 Machine 2 processes no lots Size:0 Machine 3 : 1 Size:1 Machine 4 processes no lots Size:0 Machine 5 processes no lots Size:0
2 Lots					
Makespan 1 : 3.28178 Makespan 2 : 6 Makespan 3 : 0 Makespan 4 : 8.57148 Makespan 5 : 3.4 Total Makespan: 21.2533 8.57148	Machine 1 : 1 Size:1 Machine 2 processes no lots Size:0 Machine 3 processes no lots Size:0 Machine 4 : 2 Size:1	Makespan 1 : 3.28178 Makespan 2 : 6 Makespan 3 : 0 Makespan 4 : 8.57148 Makespan 5 : 3.4 Total Makespan: 21.2533 8.57148	Machine 1 : 1 Size:1 Machine 2 processes no lots Size:0 Machine 3 processes no lots Size:0 Machine 4 : 2 Size:1	Makespan 1 : 1.56908 Makespan 2 : 6 Makespan 3 : 0 Makespan 4 : 8 Makespan 5 : 4.77004 Total Makespan: 20.3391 8	Machine 1 : 2 Size:1 Machine 2 processes no lots Size:0 Machine 3 processes no lots Size:0 Machine 4 processes no lots

Objective Function: 64.1106	Machine 5 processes no lots Size:0	Objective Function: 64.1106	Machine 5 processes no lots Size:0	Objective Function: **60.3391**	Size:0 Machine 5 : 1 Size:1
3 Lots					
Makespan 1 : 3.28178 Makespan 2 : 7.12187 Makespan 3 : 0 Makespan 4 : 8.57148 Makespan 5 : 3.4 Total Makespan: 22.3751 8.57148 Objective Function: 65.2325	Machine 1 : 1 Size:1 Machine 2 : 3 Size:1 Machine 3 processes no lots Size:0 Machine 4 : 2 Size:1 Machine 5 processes no lots Size:0	Makespan 1 : 3.28178 Makespan 2 : 7.12187 Makespan 3 : 0 Makespan 4 : 8.57148 Makespan 5 : 3.4 Total Makespan: 22.3751 8.57148 Objective Function: 65.2325	Machine 1 : 1 Size:1 Machine 2 : 3 Size:1 Machine 3 processes no lots Size:0 Machine 4 : 2 Size:1 Machine 5 processes no lots Size:0	Makespan 1 : 1.5216 Makespan 2 : 7.30908 Makespan 3 : 0 Makespan 4 : 8.57148 Makespan 5 : 3.4 Total Makespan: 20.8022 8.57148 Objective Function: **63.6595**	Machine 1 : 3 Size:1 Machine 2 : 1 Size:1 Machine 3 processes no lots Size:0 Machine 4 : 2 Size:1 Machine 5 processes no lots Size:0
4 Lots					
Makespan 1 : 3.28178 Makespan 2 : 7.12187 Makespan 3 : 3.22885 Makespan 4 : 8.57148 Makespan 5 : 3.4 Total Makespan: 25.604	Machine 1 : 1 Size:1 Machine 2 : 3 Size:1 Machine 3 : 4 Size:1 Machine 4 : 2 Size:1 Machine 5 processes no lots Size:0	Makespan 1 : 3.28178 Makespan 2 : 7.12187 Makespan 3 : 3.22885 Makespan 4 : 8.57148 Makespan 5 : 3.4 Total Makespan: 25.604	Machine 1 : 1 Size:1 Machine 2 : 3 Size:1 Machine 3 : 4 Size:1 Machine 4 : 2 Size:1 Machine 5 processes no lots Size:0	Makespan 1 : 3.28178 Makespan 2 : 7.30465 Makespan 3 : 0 Makespan 4 : 8 Makespan 5 : 5.27433 Total Makespan: 23.8608 8	Machine 1 : 1 Size:1 Machine 2 : 4 Size:1 Machine 3 processes no lots Size:0 Machine 4 processes no lots Size:0 Machine 5 : 2-3

8.57148 Objective Function: 68.4614		8.57148 Objective Function: 68.4614		Objective Function: 63.8608	Size:2
5 Lots					
Makespan 1 : 3.28178 Makespan 2 : 7.12187 Makespan 3 : 3.22885 Makespan 4 : 8.57148 Makespan 5 : 4.65231 Total Makespan: 26.8563 8.57148 Objective Function: 69.7137	Machine 1 : 1 Size:1 Machine 2 : 3 Size:1 Machine 3 : 4 Size:1 Machine 4 : 2 Size:1 Machine 5 : 5 Size:1	Makespan 1 : 3.28178 Makespan 2 : 7.12187 Makespan 3 : 3.22885 Makespan 4 : 8.57148 Makespan 5 : 4.65231 Total Makespan: 26.8563 8.57148 Objective Function: 69.7137	Machine 1 : 1 Size:1 Machine 2 : 3 Size:1 Machine 3 : 4 Size:1 Machine 4 : 2 Size:1 Machine 5 : 5 Size:1	Makespan 1 : 0.5 Makespan 2 : 7.30465 Makespan 3 : 0 Makespan 4 : 8.57148 Makespan 5 : 6.52579 Total Makespan: 22.9019 8.57148 Objective Function: **65.7593**	Machine 1 processes no lots Size:0 Machine 2 : 4 Size:1 Machine 3 processes no lots Size:0 Machine 4 : 2 Size:1 Machine 5 : 3-1-5 Size:3
10 Lots					
Makespan 1 : 5.21343 Makespan 2 : 8.50589 Makespan 3 : 3.22885 Makespan 4 : 11.3213	Machine 1 : 1-6-10 Size:3 Machine 2 : 3-8 Size:2 Machine 3 : 4 Size:1 Machine 4 : 2-7 Size:2	Makespan 1 : 5.21343 Makespan 2 : 8.50589 Makespan 3 : 3.22885 Makespan 4 : 10.0442	Machine 1 : 1-6-10 Size:3 Machine 2 : 3-8 Size:2 Machine 3 : 4 Size:1 Machine 4 : 7-2 Size:2	Makespan 1 : 6.05355 Makespan 2 : 7.34403 Makespan 3 : 2.52298 Makespan 4 : 10.1196	Machine 1 : 2-5-9- 10 Size:4 Machine 2 : 4-6 Size:2 Machine 3 : 3 Size:1 Machine 4 : 1-8

Makespan 5 : 5.90463 Total Makespan: 34.1741 11.3213 Objective Function: 90.7807	Machine 5 : 5-9 Size:2	Makespan 5 : 5.90463 Total Makespan: 32.897 10.0442 Objective Function: 83.1182	Machine 5 : 5-9 Size:2	Makespan 5 : 4.65231 Total Makespan: 30.6925 10.1196 Objective Function: **81.2904**	Size:2 Machine 5 : 7 Size:1
15 Lots					
Makespan 1 : 6.2697 Makespan 2 : 8.54528 Makespan 3 : 7.74875 Makespan 4 : 12.1732 Makespan 5 : 5.90463 Total Makespan: 40.6415 12.1732 Objective Function: 101.507	Machine 1 : 1-6-10-12 Size:4 Machine 2 : 3-8-14 Size:3 Machine 3 : 4-11-13 Size:3 Machine 4 : 2-7-15 Size:3 Machine 5 : 5-9 Size:2	Makespan 1 : 6.2697 Makespan 2 : 8.54528 Makespan 3 : 7.74875 Makespan 4 : 11.5231 Makespan 5 : 5.90463 Total Makespan: 39.9915 11.5231 Objective Function: 97.6072	Machine 1 : 1-6-10-12 Size:4 Machine 2 : 3-8-14 Size:3 Machine 3 : 4-11-13 Size:3 Machine 4 : 7-2-15 Size:3 Machine 5 : 5-9 Size:2	Makespan 1 : 8.61864 Makespan 2 : 9.98737 Makespan 3 : 6.58733 Makespan 4 : 10.151 Makespan 5 : 8.37133 Total Makespan: 43.7157 10.151 Objective Function: 94.4706	Machine 1 : 3-10-12-15 Size:4 Machine 2 : 1-4-6-8-14 Size:5 Machine 3 : 11 Size:1 Machine 4 : 13 Size:1 Machine 5 : 5-7-2-9 Size:4
25 Lots					
Makespan 1 : 11.3818 Makespan 2 : 10.8128 Makespan 3 :	Machine 1 : 1-6-10-12-16-24 Size:6 Machine 2 : 3-8-14-19-22	Makespan 1 : 11.3818 Makespan 2 : 10.8128 Makespan 3 :	Machine 1 : 1-6-10-12-16-24 Size:6 Machine 2 : 3-8-14-19-22	Makespan 1 : 11.3818 Makespan 2 : 11.0085 Makespan 3 :	Machine 1 : 2-1-6-12-10-9-16-14-24 Size:9 Machine 2 : 8-20-21 Size:3

12.5761 Makespan 4 : 14.5147 Makespan 5 : 9.53662 Total Makespan: 58.822 14.5147 Objective Function: 131.395	Size:5 Machine 3 : 4-11-13-20-23 Size:5 Machine 4 : 2-7-15-18-25 Size:5 Machine 5 : 5-9-17-21 Size:4	12.5761 Makespan 4 : 13.2459 Makespan 5 : 9.53662 Total Makespan: 57.5532 13.2459 Objective Function: 123.783	Size:5 Machine 3 : 4-11-13-20-23 Size:5 Machine 4 : 2-18-15-7-25 Size:5 Machine 5 : 5-9-17-21 Size:4 Machine 1 : 1-6-10-12-16-24 Size:6 Machine 2 : 3-8-14-19-22 Size:5 Machine 3 : 4-11-13-20-23 Size:5 Machine 4 : 2-15-7-18-25 Size:5 Machine 5 : 5-9-17-21 Size:4	12.3087 Makespan 4 : 11.775 Makespan 5 : 12.1472 Total Makespan: 58.6212 12.3087 Objective Function: 120.165	Machine 3 : 13-4-23 Size:3 Machine 4 : 7-11-22 Size:3 Machine 5 : 5-3-15-18-17-19-25 Size:7
35 Lots					
Makespan 1 : 14.7375	Machine 1 : 1-6-10-12-16-24-31	Makespan 1 : 14.7375	Machine 1 : 1-6-10-12-16-24-31	Makespan 1 : 17.3775	Machine 1 : 2-12-10-16-20-24-28-29-

Makespan 2 : 14.3744 Makespan 3 : 14.9646 Makespan 4 : 18.1446 Makespan 5 : 16.8402 Total Makespan: 79.0612 18.1446 Objective Function: 169.784	Size:7 Machine 2 : 3-8-14-19-22-27-32 Size:7 Machine 3 : 4-11-13-20-23-28-33 Size:7 Machine 4 : 2-7-15-18-25-29-34 Size:7 Machine 5 : 5-9-17-21-26-30-35 Size:7	Makespan 2 : 14.3744 Makespan 3 : 14.9646 Makespan 4 : 17.0792 Makespan 5 : 16.8402 Total Makespan: 77.9958 17.0792 Objective Function: 163.392	Size:7 Machine 2 : 3-8-14-19-22-27-32 Size:7 Machine 3 : 4-11-13-20-23-28-33 Size:7 Machine 4 : 2-15-7-18-25-29-34 Size:7 Machine 4 : 2-18-15-7-25-29-34 Size:7 Machine 4 : 18-15-7-2-25-29-34 Size:7 Machine 5 : 5-9-17-21-26-30-35 Size:7	Makespan 2 : 14.3744 Makespan 3 : 6.95899 Makespan 4 : 17.5385 Makespan 5 : 14.7379 Total Makespan: 70.9873 17.5385 Objective Function: 158.68	31-33-35 Size:11 Machine 2 : 8-6-25-26-32 Size:5 Machine 3 : 13 Size:1 Machine 4 : 4-11-9-18-14-17-27-22-34 Size:9 Machine 5 : 3-1-5-15-7-19-23-21-30 Size:9

Abstract

The main topics of this thesis deal with the optimization of the program planning and operation scheduling in the semiconductor manufacturing industry.

The work consists of two separate parts. The first section describes a company called Infineon Technologies AG as an instance of the theoretical background of the semiconductor production process with its specific requirements. The whole procedure underlies great sensitivity and complexity.

As the whole process of a manufacturing flow cannot be taken into consideration, the thesis points out two different real–world problems: a product mix and a material flow problem. Thereby, it is attempted to maximize the profit on one hand and to minimize the total makespan on the other. These two issues are initially discussed in a theoretical way. Moreover, their mathematical formulation is established and in further consequence solved optimally with Xpress.

The product mix subproblem requires solely the implementation in Xpress since the generated results are optimal and the computational time ranges around zero in each run. Therefore, no further comparison with another software implementation is shown.

The generated solution of the material flow problem is compared with heuristically results found through the implementation in C++. These results cannot gain optimality but provide good and feasible solutions for a bigger range of instances in reasonable computation time.

Zusammenfassung

Das zentrale Thema dieser Arbeit behandelt die Optimierung der Programm- und Ablaufplanung in der Halbleiterindustrie.

Die Diplomarbeit besteht aus zwei separaten Teilen. Der erste Abschnitt befasst sich mit einem Unternehmen aus dieser Branche namens Infineon Technologies AG. Dieser internationale Konzern dient als Beispiel für den theoretischen Hintergrund des Halbleiter-Fertigungsprozesses mit seinen spezifischen Anforderungen. Das zugrunde liegende Verfahren zeichnet sich durch große Anfälligkeit der Produkte während des Produktionsprozesses und enorme Komplexität aus.

Da die gesamte Fertigung nicht als Ganzes betrachtet und optimiert werden kann, werden in der Diplomarbeit zwei unterschiedliche Problemstellungen angeführt: ein Produkt-Mix und ein Material-Flow-Problem. Dabei wird einerseits versucht, den Profit zu maximieren, andererseits soll die gesamte Herstellungszeit innerhalb einer Werkstatt minimiert werden. Diese beiden Sachverhalte werden zunächst theoretisch diskutiert und in weiterer Folge wird die mathematische Modellierung mit Xpress optimal gelöst.

Das Produkt-Mix Teilproblem erfordert lediglich die Umsetzung in Xpress, da die generierten Ergebnisse Optimalität aufweisen und die Rechenzeit sich um 0 Sekunden in jedem Durchlauf bewegt. Daher wird kein weiterer Vergleich mit einer anderen Software-Implementierung dargestellt.

Die generierten Lösungen des Materialfluss-Problems aus Xpress werden mit den heuristischen Ergebnissen anhand der Implementierung in C++ verglichen. Diese Ergebnisse erreichen die Optimalität nicht, sondern bieten eine gute und praktikable Lösung für eine größere Auswahl von Fällen in angemessener Rechenzeit.

Curriculum Vitae: Vera Hager

Persönliche Daten

Name:	Vera Hager
Geburtsdatum, -ort:	27.Februar 1984 in Wels
Staatsbürgerschaft:	Österreich
Familienstand:	ledig

Universität und Schulausbildung

10/2004 –12/2009	**Universität Wien**, Betriebswirtschafts Zentrum (BWZ), Wien Studium der Internationalen BWL Spezialisierung: Produktionsmanagement Controlling
09/2002 – 05/2004	**Kolleg** für„Tourismus und Freizeitwirtschaft“ Fachrichtung: Food and Beverage Ausbildungsschwertpunkt: Hotelmanagement, Diplomprüfung mit ausgezeichnetem Erfolg, Kleßheim (Salzburg)
09/1994 – 06/2002	**Wirtschaftkundliches Realgymnasium** der Franziskanerinnen, Wels
09/1990 – 06/1994	**Volkschule** Rainerstraße, Wels

Curriculum Vitae: Karin Spannraft

Persönliche Daten

Name:	Karin Spannraft
Geburtsdatum, -ort:	14.Oktober 1983 in Villach
Staatsbürgerschaft:	Österreich
Familienstand:	ledig

Universität und Schulausbildung

10/2004 –12/2009	**Universität Wien**, Betriebswirtschafts Zentrum (BWZ), Wien Studium der Internationalen BWL Spezialisierung: Produktionsmanagement Internationale Unternehmensführung
09/1999 – 06/2004	**Centrum Humanberuflicher Schulen des Bundes,** Villach Schwerpunkt: 3. lebende Fremdsprache
09/1994 - 07/1999	**BG/BRG St. Martin**, Villach
09/1990 – 07/1994	**Volkschule St. Martin**, Villach