TUCS

Sami Pyöttiälä

Optimization and Measuring Techniques for Collect-and-Place Machines in Printed Circuit Board Industry

Turku Centre *for* Computer Science

TUCS Dissertations
No 190, March 2015

Optimization and Measuring Techniques for Collect-and-Place Machines in Printed Circuit Board Industry

Sami Pyöttiälä

To be presented, with the permission of the Faculty of Mathematics and Natural Sciences of the University of Turku, for public criticism in Tauno Nurmela Hall (Lecture Hall I) on March 21, 2015, at 12 noon.

University of Turku
Department of Information Technology
FI-20014 Turun yliopisto
Finland

2015

Supervisors

Professor Olli Nevalainen, PhD
Department of Information Technology
University of Turku
Turku, Finland

Research director Timo Knuutila, PhD
Technology Research Center
Brahea Center, University of Turku
Turku, Finland

Reviewers

Professor Miklós Krész, PhD
Applications of Informatics Department
University of Szeged
Szeged, Hungary

Professor Martti Penttonen, PhD
Department of Computer Science
University of Eastern Finland
Kuopio, Finland

Opponent

Professor Erkki Mäkinen, PhD
School of Information Sciences
University of Tampere
Tampere, Finland

The originality of this thesis has been checked in accordance with the University of Turku quality assurance system using the Turnitin OriginalityCheck service.

ISBN 978-952-12-3157-5 (print)
ISBN 978-952-12-3158-2 (electronic)
ISSN 1239-1883

Abstract

This thesis considers optimization problems arising in printed circuit board assembly. Especially, the case in which the electronic components of a single circuit board are placed using a single placement machine is studied. Although there is a large number of different placement machines, the use of collect-and-place -type gantry machines is discussed because of their flexibility and increasing popularity in the industry.

Instead of solving the entire control optimization problem of a collect-and-place machine with a single application, the problem is divided into multiple subproblems because of its hard combinatorial nature. This dividing technique is called *hierarchical decomposition*. All the subproblems of the one PCB - one machine -context are described, classified and reviewed. The derived subproblems are then either solved with exact methods or new heuristic algorithms are developed and applied. The exact methods include, for example, a greedy algorithm and a solution based on dynamic programming. Some of the proposed heuristics contain constructive parts while others utilize local search or are based on frequency calculations.

For the heuristics, it is made sure with comprehensive experimental tests that they are applicable and feasible. A number of quality functions will be proposed for evaluation and applied to the subproblems. In the experimental tests, artificially generated data from Markov-models and data from real-world PCB production are used. The thesis consists of an introduction and of five publications where the developed and used solution methods are described in their full detail. For all the problems stated in this thesis, the methods proposed are efficient enough to be used in the PCB assembly production in practice and are readily applicable in the PCB manufacturing industry.

Tiivistelmä

Tässä väitöskirjatutkimuksessa käsitellään piirilevyjen valmistukseen käytettävien komponenttiladontakoneiden ohjauksessa esiintyviä optimointiongelmia. Erikoisesti keskitytään kontekstiin, jossa yksittäisellä ladontakoneella valmistetaan yhdenlaista piirilevytyyppiä. Komponenttiladontakoneita on teollisuudessa käytössä hyvin monenlaisia, mutta tässä tutkimuksessa keskitytään collect-and-place-tyyppisen koneen ohjaukseen. Kyseinen konetyyppi soveltuu monenlaiseen tuotantoon ja on joustavuutensa takia suosittu.

Sen sijaan, että collect-and-place-koneen ohjauksen optimointiongelma yritettäisiin ratkaista yhtenä kokonaisuutena, ongelma on päädytty jakamaan usean osaongelman hierarkiaksi, sillä ongelma on luonteeltaan kombinatorinen ja vaikea ratkaista optimaalisesti vaihtoehtoisten ratkaisujen suuren lukumäärän vuoksi. Tätä ongelmanratkaisuun liittyvää jakotapaa kutsutaan *hierarkkiseksi jakamiseksi*. Kaikki muodostetun ongelmahierarkian osaongelmat kuvaillaan ja luokitellaan. Lisäksi niiden ratkaisumahdollisuuksiin luodaan katsaus. Osaongelmille esitetään joko eksaktiin menetelmään perustuva ratkaisu tai kehitetään ja sovelletaan uusia heuristisia algoritmeja. Käytettyjä eksakteja menetelmiä ovat esimerkiksi ahne algoritmi ja ratkaisujen taulukointiin perustuva dynaamisen ohjelmoinin menetelmä. Kehitetyt heuristiset lähestymistavat kattavat menetelmiä, jotka perustuvat esimerkiksi ominaisuuksien esiintymistiheyksien laskemiseen, paikallishakuun tai ratkaisun muodostamiseen pienemmistä osista rakentamalla.

Heuristiikkojen sovellettavuus ja toteuttamiskelpoisuus on varmistettu kattavilla, kokeellisilla testeillä. Useita erilaisia ratkaisujen laadun evaluointiin käytettäviä funktioita on määritelty ja sovellettu osaongelmiin. Testeissä on käytetty sekä erityisesti tähän tarkoitukseen kehiteltyjen Markov-mallien avulla generoitua, keinotekoista satunnaisdataa että teollisuudesta saatua aitojen piirilevyjen valmistusdataa.

Tämä väitöskirjatutkimus koostuu viidestä osajulkaisusta ja ne yhteenvetävästä johdanto-osuudesta. Kehitetyt ja käytetyt ongelmanratkaisumenetelmät on kuvattu yksityiskohtaisesti osajulkaisuissa. Kaikki tämän tutkimuksen ratkaisumenetelmät ovat tarpeeksi tehokkaita ja soveltuvat käyttöönotettaviksi teollisessa piirilevyjen valmistuksessa.

Acknowledgements

This dissertation represents results of the research work that took a place in my life during the period of eight years. It took almost half of my work time. I am very grateful that I have had this opportunity to study the world of science in the rich guidance of and varied support from great people.

First, I would like to thank both my supervisors, Professor Olli Nevalainen and Research Director Timo Knuutila, of their priceless help and support. This thesis was possible because of their accurate and indomitable guidance. Second, I wish to extend my thanks to Docent Mika Johnsson for co-authoring the publications and providing up-to-date research problems to our research team. Also, I would like to thank the reviewers, Professor Miklós Krész and Professor Martti Penttonen, for their effort and important feedback.

Third, I have the honour to thank all the fellow workers at the department of information technology. I have learned a lot from you. Good memories of the years I was honoured to work, co-operate, laugh and drink coffee with you will last the rest of my life. Especially, I want to thank PhD Jukka Teuhola, Professor Tapio Salakoski, Professor Ville Leppänen, Professor Ari Paasio, Maria Prusila, PhD Olli Luoma, Kari Salonen, PhD Janne Lahtiranta, Mika Murtojärvi, Docent Tapio Pahikkala, D.Sc.(tech) Sami Hyrynsalmi, PhD Jouni Järvinen, Päivi Rastas, Tomi Suovuo, Docent Esa Alhoniemi, Harri Hakonen, PhD Jussi Salmi, PhD Jouni Smed, PhD Mikko Laakso, Jussi Hakokari and PhD Tuomo Saarni. They have helped me a thousand times in many different ways. Actually, the comprehensive list of people to be thanked would include a few dozen names. In addition, I need to thank Marko Lindholm in WinNova Elepro, Rauma, for his volunteer efforts in showing and presenting their production line.

Then, I would like to thank all my friends for indirect and endless support that is impossible to specify. However, special thanks go to Janne Puurunen who helped me also directly with his skills in visualization in my hardest rush. My friends, I feel greatly privileged to live my life among you! This dissertation would not have been written without the wide-ranging relations, sharing and free time activity you are making possible all the time. Note, that $fellowworkers \cap friends \neq \emptyset$.

I must thank the bright summers of midnight sun in the northern wilderness for their stability, stress and rest, and the ability of stopping the passage of time, and the regenerative influence. These kinds of experiences in nature are very helpful in any long term journey and I have had the privilege and health to draw from them for years — with friends and family.

Finally, I want to express my gratitude to my family for the unlimited support, faith and love in every turn and moment. My words and deeds will never be enough. Especially, I would like to thank my beloved wife Tiia who has supported and helped me a lot even with this, and encouraged me to carve out time and space for this thesis from other activities and vice versa.

List of original publications

P1 T. Knuutila, S. Pyöttiälä, and O.S. Nevalainen. Minimizing the number of pickups on a multi-head placement machine. *Journal of the Operational Research Society*, 58(1):115-121, Jan, 2007.

P2 S. Pyöttiälä, T. Knuutila, and O.S. Nevalainen. The selection of nozzles for minimizing the number of pick-ups on a multi-head placement machine. In *Proceedings of the Third International Conference on Group Technology / Cellular Manufacturing 2006*, pages 382-391, Groningen, The Netherlands, July 2006.

P3 T. Knuutila, S. Pyöttiälä, and O.S. Nevalainen. Minimizing the arm movements of a multi-head gantry machine. In *Proceedings of 4th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2007)*, pages 326-335, Angers, France, May 2007.

P4 S. Pyöttiälä, T. Knuutila, M. Johnsson, and O.S. Nevalainen. Solving the feeder assignment on a revolver-head gantry machine. In *Proceedings of 6th International Conference on Informatics in Control, Automation and Robotics, (ICINCO 2009)*, volume 1 - Intelligent Control Systems and Optimization, pages 75-80, Milan, Italy, July 2009.

P5 S. Pyöttiälä, T. Knuutila, M. Johnsson, and O.S. Nevalainen. Minimizing the assembly cycle time on a revolver gantry machine. *Computers & Operations Research* 40, pp. 2611-2624, 2013.

Contents

Chapter 1

Introduction

Electronics has become an inseparable part of everyday life. In the developed countries people normally use several devices everyday, smaller and larger, which are either completely or partly electronic. Electronics can be found in, for example, phones, clocks, stoves, radios, televisions, washing machines, toys, measuring equipment, cars, and computers. The functionality of the electronic parts of devices has been achieved by suitably combining electronic *component*s onto a *printed circuit board (PCB)*. Larger, appropriately functioning entities are then formed from the components with conducting wires on the PCB.

Since electronics is so common it is economically significant for the end user, retailer, as well as for the manufacturer. It is naturally desirable to produce electronic devices as cost effectively as possible while retaining an appropriate level of quality. In this thesis a narrow sector of the manufacturing of electronics, a single step in the manufacturing process of a PCB is considered: placing components on a PCB with a special placement machine and performing the placing as *efficiently* as possible. In this thesis, the collect-and-place machines are discussed to reach higher concreteness. Additionally, this machine type is very popular in the industry.

The entire control problem of a single collect-and-place machine is divided into several subproblems because of its hard combinatorial nature. The subproblems are then solved using exact methods or heuristic approaches. The exact methods include, for example, a greedy algorithm that is shown to produce an optimal solution for the discussed subproblem. For another subproblem, dynamic programming is utilized, also. The optimal solutions and their values of the object function are memorized for the fragments of the subproblem. The final solution is then constructed from these particles.

Some of the heuristics are based on the use of frequency calculations while others apply constructive deduction. For heuristics, it is verified in experimental tests that the proposed algorithms are applicable and feasible. The experimental tests are based on the data of real-world PCB production, or in some cases, data is artificially generated using specialized Markov-models.

In the context of PCB assembly, *efficiency* can mean the shortest possible time spent on placing components on a single PCB, the shortest possible distance traveled by the moving parts of the placement machine or the smallest possible amount of recurring stages. Several functions are proposed for measuring the quality of solutions in heuristics and experimental tests in terms of efficiency. However, *quality* is a complicating matter in optimizing operation efficiency. The quality of a PCB depends on several factors such as the acceleration of the components of machines and the tools used in the manufacturing process.

This thesis is based on five articles in which fundamental subproblems arising from the control of the collect-and-place machines are studied and solved. The developed algorithms are tested using artificially generated data and data from real-world production. The goal of the introduction part of this thesis is to introduce the five articles of this dissertation and to conclude the common purpose of the articles as well as to report the achieved results. As a secondary result, this introduction could be used as a beginners' study material for optimization of collect-and-place machines.

The rest of the introduction is structured as follows. In Chapter 2, the component placement process in general, and especially the collect-and-place -machine and the properties of its operation, are represented. The optimization problems of the single machine are discussed and the research problem along with the main results of this thesis is defined in Chapter 3. Chapter 4 summarizes the publications. Finally, the concluding remarks are written in Chapter 5.

Chapter 2

Properties and control of component placement machines

2.1 Production of printed circuit boards

The production process of a printed circuit board begins when the needed functionality and layout of the PCB is defined by a designer (see the first two phases of Figure 2.1). Usually, computer aided design tools are used for the sake of practicality. The designer chooses the suitable components for the PCB and sets the wires between them so that the desired functionality is achieved. For the placement of the components, the most essential pieces of information of the plan are types, shapes, angles, and positions of the components but also the dimensions of the PCB. There is a lot of detailed information for the other phases of the production process, too. These include data for several automatically controlled machines.

After the layout is ready, the process continues with the construction of the empty physical PCB (the third phase of Figure 2.1). The wires are made, for example, using etching in the factory which is specialized in the production of empty PCBs . As a result the wiring planned by the designer is achieved. It is common that a large amount of empty PCBs of a certain type are etched at a time. Figure 2.2 presents an empty PCB with wires on its surface but which still lacks all the components.

The electronic components are placed and fixed on their proper positions on the PCB in a factory that performs PCB assembly (the fourth phase of Figure 2.1). In addition to the empty circuit board with etched wires, the necessary components

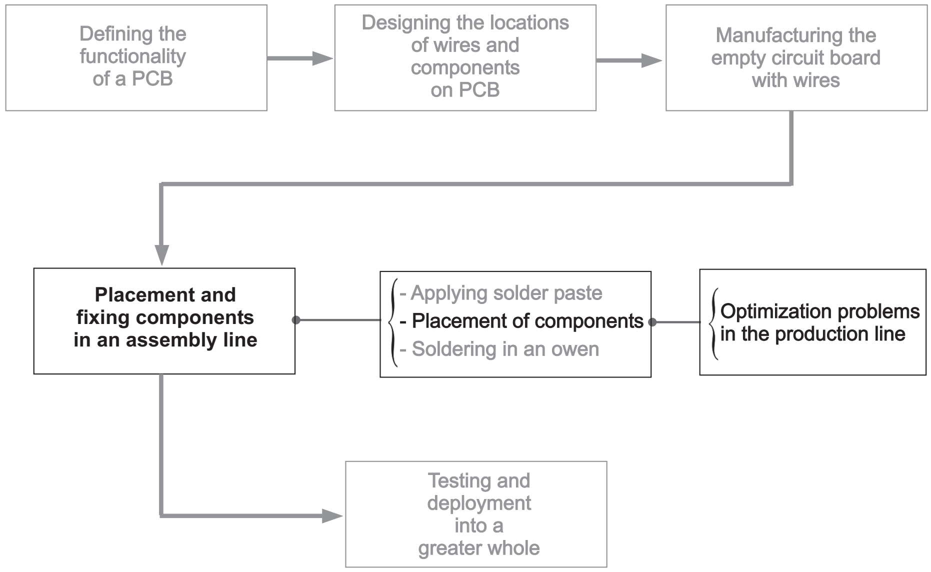

Figure 2.1: The phases of the manufacturing process of a printed circuit board on a high abstraction level. Topics considered in this work are written in boldface.

have to be purchased. The components can be shipped to the factory, for example, in special tape reels that fit straight away into some assembly devices of the production lines. A conventional appearance of such reels is presented in Figure 2.3. Usually, there are only components of a certain type in a single tape reel with a regular spacing in between. The external dimensions, i.e., the shapes of the different components vary. They can also be attached in different angles in the tape reels. Some components can be shipped separately to the factory. In this case the components have to be set up manually in some feeder device before the placement machinery can utilize them.

The components are fixed on the PCB with solder paste that connects the conducting wires of the components into the wires lying on the PCB. The solder paste is applied on the precise placement positions of the components on the PCB with a solder paste spreading machine specially designed for the job. In a typical paste spreading machine a stencil is utilized. The stencil is prepared according to the design of each PCB type and its purpose is to prevent the solder paste from ending up outside the correct placement points. Once the paste is applied the components are placed in the assembly line which can consist of several placement machines. The components are placed exactly on the regions of PCB surface in which paste

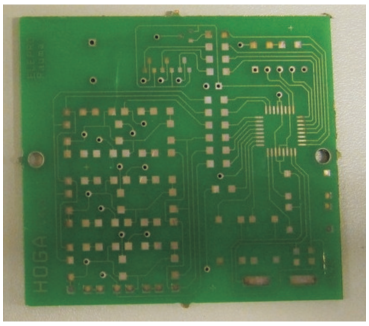

Figure 2.2: An empty circuit board with wires.

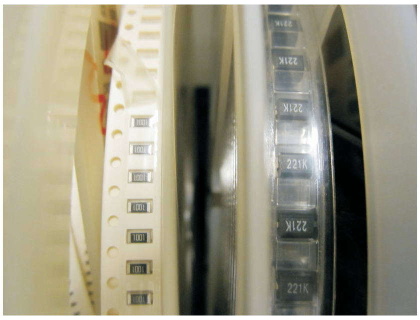

Figure 2.3: Two different types of components in tapes.

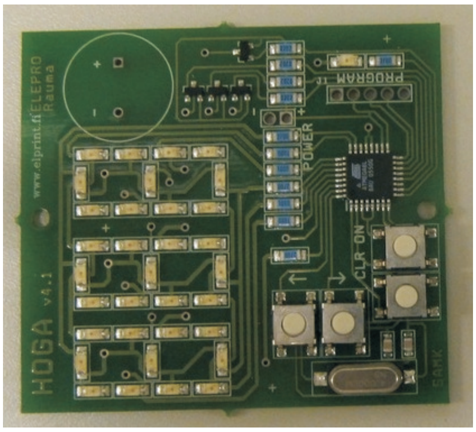

Figure 2.4: Components placed on a PCB.

is spread. Often, in the end of the assembly line there is an oven where the PCB is heated to a certain temperature. After heating the PCB is let to cool off and the paste fixes the components on the PCB permanently. Finishing operations include cleaning, checking and testing of the product. A finished PCB is presented in Figure 2.4. Finally, a ready batch of PCBs is transported into another production facility where they are attached as an essential part of a greater whole (the last phase of Figure 2.1).

Functions and the components of the PCBs that are designed for different electronic devices vary significantly as well as their sizes and prices. For example, the motherboard of a personal computer is larger and needs notably more components than is found from a wrist device which monitors the heart rate of an athlete. The dimensions of PCBs vary from few millimeters to half a meter. During the assembly process, the number of fixed components can be even several thousands [52]. Usually, numerous copies of the PCB of a certain type are produced. In the biggest batches, several assembly lines or even many factories are used to produce the same PCBs.

2.2 Placement machines in PCB assembly

Mass-produced PCBs are manufactured in highly automated assembly lines. An example of a very simple and minimal assembly line is presented in Figure 2.5. First, the supplying device for empty PCBs (A in Figure 2.5) passes a copy of the clean board with ready-etched wiring on the conveyor belt which carries it to the soldering paste spreading machine (B). The solder paste is then applied on the surface of an empty PCB one board at a time.

From the solder paste spreading the PCB proceeds to the component placement machine (C) via a conveyor belt. There can be several placement machines one after another on a single assembly line. Beforehand, a human operator has set up the component tape reels or some other component feeder technology in the feeder units of the placement machines based on the data in the manufacturing plan of the PCB which is also called a PCB recipe.

The only purpose of the placement machinery is to pick up the right components from the feeders and place them in their proper positions defined in the PCB recipe. However, before the necessary components can be picked up and placed, the order of the placements has to be defined. In addition, the organization of the component tape reels in the feeder units has to be defined for the machines and the nozzles must be selected into the placement head and into the nozzle magazine.

If there are multiple lines or multiple placement machines on the line it has to be decided, which components and placements are handled with each line or machine. In this thesis, the control of a single placement machine is studied. This includes multiple interconnected decisions such as the order of the component tape reels in the feeder unit, the usage policy of tools which are used to grab components, the composition of the component pickup, placement tours of the machine and so on. These decisions are observed from the point of through-put efficiency of the machine. Notice, that even if there are multiple machines in the line the problems relating a single machine have to be solved before production can begin.

Figure 2.5: A production line in which a conveyor belt connects an empty board supply station (A), a solder paste spreading machine (B), a placement machine (C) and oven (D). The model of the placement machine is Siemens Siplace 80 F4

After the components are placed, the PCB is conveyed to an oven (D) in which the PCB is heated to the required temperature. In the cooling off -phase the solder paste connects the components to the wires of the PCB and fixes them permanently at their positions. Thus, the PCB has acquired the features that were designed and some finalizing activities may be applied, such as cleaning and testing. Sometimes part of the checking can be done before the PCB goes into the oven. If quality problems are detected, at least the most expensive components are easier to detach for reuse if they are not fixed yet. Some of the components may be so inexpensive that their manual cleaning for reuse is not worthwhile. The production of the PCB ends here, but the product has to be packed or attached into some device, marketed and sold.

In this thesis, a narrow (but difficult) part of the PCB production is studied: how to make a single placement machine produce a single PCB as quickly as possible. The influence of the human operators is omitted, and only the actions of the placement machine are considered. While there are various types of placement machines, here the focus is in so called gantry machines with a multiple placement spindles, mostly with a revolver-type organization of spindles. In Figure 2.6 such multi-spindle revolver gantry placement machine is pictured.

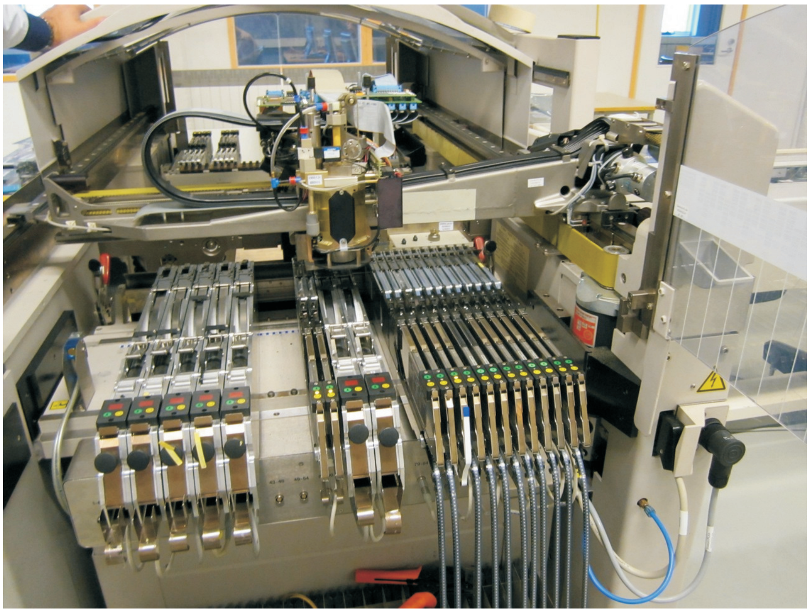

Figure 2.6: Component placement machine (Siemens Siplace 80 F4) equipped with both the pick-and-place- and the collect-and-place-heads. The collect-and-place-head is a revolver-type (see Figure 2.9) and has 12 spindles for nozzles.

2.3 Types of placement machines

According to Crama et al. [13] problems emerging in placement machine control vary greatly depending on the machine type. Even two quite similar types of machines may have fundamental differences that make problem definitions and solution models incompatible and are not directly portable between the machine types. The primary differences between placement machines divide the machine types into five categories. Here, the machine types are described briefly. For an extensive classification of different placement machine types, see Ayob et al. [4, 5].

*Dual-delivery machine*s have two gantry-type placement heads which operate in turns. One head picks components into its nozzles while another head places components on a PCB. There are multiple nozzles in an annular placement head that moves on the (x, y)-plane above the PCB and feeder during its operation. There are two conveyor belts and own feeders for both heads which pick up components on their own sides of the machine. For a more accurate description of the machine type, see for example [61, 2].

In *chip-shooter machine*s, also known as *turret machine*s, a PCB is fixed into the table that is able to move on the (x, y)-plane. Components are picked up and placed by a rotating turret that has multiple nozzles on it. The components are supplied by a feeder unit that moves on the x-axis. The turret picks up components from the feeder unit simultaneously with the placement operation of the nozzle of opposite position. There are numerous studies in which these machines are discussed, for example [8, 13, 18, 24, 31, 19].

In *multi-station machine*s, there are multiple identical assembly stations in a sequential orientation. The stations are connected by a conveyor belt. In the production process of a PCB each station of the machine performs a certain part of the placements. The problems for this machine type are studied, for example, in [21, 17].

*Collect-and-place machine*s are especially studied in this thesis. These machines have a moving placement head of multiple nozzles and while the feeder unit and PCB are stationary. The placement head collects components from the feeder unit and transports and places them on a PCB. The machine has either a rotating revolver head or an in-line head. The collect-and-place machine type is described in greater detail in Section 2.6.

Pick-and-place machine is a special case of a collect-and-place machine. It has only one nozzle in the placement head. This machine type is introduced in Subsection 2.5.1. So called *radial placement machine*s belong to the class of pick-and-place machines. [50, 51]

2.4 The production control problems in an assembly line

The whole optimization problem relating to printed circuit board assembly lines can be divided into parts so that one can solve the resulting simpler problems [56]. This process is called as the *hierarchical decomposition* of the problem [14]. Problems of various levels exist depending on the point-of-view the PCB assembly process is observed [62]. Johnsson et al. [28] and Smed [55] propose a matrix for the problem classification that splits the problem domain by the number of PCB types to be produced and machines to be used. The number of produced PCB types can be either 1 or N and in production one or multiple machines (M) can be used.

Using this approach, each subproblem of any hierarchical decomposition can be assigned into some of the following four categories:

- 1 PCB type - 1 machine
- N PCB types - 1 machine
- 1 PCB type - M machines
- N PCB types - M machines.

In general, there are multiple component placement machines in a factory. These machines are used to manufacture a set of different PCB types. For a contract manufacturer, this set may be very large. On this production control level, there are problems such as how the production lines are organized by the existing placement machines, or which PCBs are produced in which assembly line. Often, the goal is to balance the workload between the lines, and also between the machines in a single line, so that no machine stays idle. Further, the scheduling of PCB jobs should be done in order to fulfill the given deadlines. There are cases in which the production times can be speed up by grouping the PCB jobs so that the idle time of an assembly line while changing the product from one PCB type to another is minimized. These kind of line balancing and job grouping problems are studied, for example, by Scholl and Becker [54] and Smed et al. [57, 58], and the scheduling of independent jobs with non-identical parallel machines is discussed by Balin [6].

The problems related to a single placement machine are divided into following two categories: multiple produced PCB types and single PCB type. First, in the case of multiple PCB types, the feeder setup and tool switching performed manually by a human operator have to be considered when the PCB type to be manufactured changes. Again, the goal is to minimize the idle time of the machine. Second, the lowest level of problems is the case where one PCB type is assembled with a single machine. In this case the self-evident goal is to organize the component tape reels in the feeder unit, decide the nozzle usage policy and the pickup and placement sequences so that the throughput of the machine is maximized. Stated otherwise, the goal is to minimize the *assembly cycle time (ACT)* of a single PCB to be produced. The assembly cycle time is defined and discussed in greater detail in Subsection 3.4.

All the categorized problems described above are highly dependent on each other and they should be treated as an entirety to be sure that the solution is as good as possible. However, it is known that, in the case of a gantry machine that uses only three nozzles and the component-to-nozzle compatibility is known, minimizing the travel length by dividing the placements into clusters of three components (which can be transported at a time) is admitted as an NP-hard problem [14] (page 19). In general, it is impossible to make a single placement machine perform optimally even if there are less than 200 component placements in the PCB recipe. By decomposing the entire problem hierarchically, in the case of a single PCB type and a single machine, the resulting subproblems can be solved separately. Some of the subproblems can be solved even optimally, for example the *minimal length pickup sequence problem* [33], while for other ones the heuristic or genetic algorithms and variants of local search are applied. Using the hierarchical approach it is clear that the found solution is not necessarily optimal or maybe not even close to optimal but it is the best one can do in a practical time.

In the research considering the control of the component placement machinery the approach of the hierarchical decomposition has been used by many authors. For example, Ahmadi et al. [1] use it for optimizing the performance of dual-delivery machines, and Crama et al. [14, 16] propose in the case of a single PCB recipe the hierarchical decomposition for a collect-and-place-machine of three nozzles and for a turret-type machine. Crama et al. also propose the use of a hierarchical approach for problems on a higher hierarchy level of multiple PCB products and placement machines. For the turret-type machine Ellis et al. [20] have applied constructive heuristics. They suggest add-on heuristics which enhance the results of the constructive heuristics and these can be seen as a variant of the local search.

For *revolver head* collect-and-place-machines, Grunow et al. [22] use hierarchical decomposition of the machine context. They solve the feeder assignment problem and propose constructive heuristics to determine the placement sequence which is treated as *vehicle routing problem* in micro-dimensions. For inline collect-and-place-machines, Lee et al. [36] determine the placement sequence using the *nearest neighbor* heuristic for the *traveling salesman problem* which appears in the PCB assembly. They solve the problems relating to the feeder assignment problem by the means of dynamic programming. Their approach is based on the hierarchical decomposition.

Contrary to many authors, a solution can be searched for several of these subproblems at a time. This approach makes the search space much wider, but if the subproblems are chosen carefully the solutions are sometimes better than if the same subproblems were solved separately. For example, Kulak et al. [35] propose a method based on genetic algorithms for solving the joint problem of the feeder assignment and placement sequencing for a revolver head collect-and-place machine. Their method turned out to outperform the hierarchical approach proposed in [22] when measured by the assembly cycle time.

Note that while there are subproblems of various levels in the PCB assembly process, finding as good as possible solution for the problems of one PCB — one placement machine usually speeds up the production.

The premise in this thesis is to develop algorithmic methods for a collect-and-place machine. These methods can yield such a machine setup, and pickup and placement order that the assembly cycle time of a single PCB recipe is minimized. The setup includes the order of the component tape reels in the feeder unit and the selection of nozzles into the spindles of the placement head. In this study, some of these problems are solved optimally using exact methods, while the others are approached with heuristics. The feasibility of the heuristic solutions is verified by experimental tests. For all the problems stated in this thesis, the methods proposed are efficient enough to be used in the PCB assembly production in practice.

2.5 Collect-and-place -type placement machine

There are numerous manufacturers of placement machines, for example, ASM Pacific, Fuji, Juki, Mydata, Panasonic, Philips, PMJ, Samsung, Sanyo, Sony, Universal and Yamaha. The manufacturers have produced several different models and variants of machines for the use of the PCB industry. For classifications of different placement machine types, see Ayob et al. [4, 5], Pham et al. [43] and Tirpak et al. [60].

In this thesis, the focus is on the collect-and-place gantry machines. The different types of placement machines differ from each other so fundamentally that it depends strongly on the machine type what kind of problems exist in their control. Thus, the solution methods proposed for a problem of a certain machine type are not necessarily directly applicable for the corresponding problem of a machine of

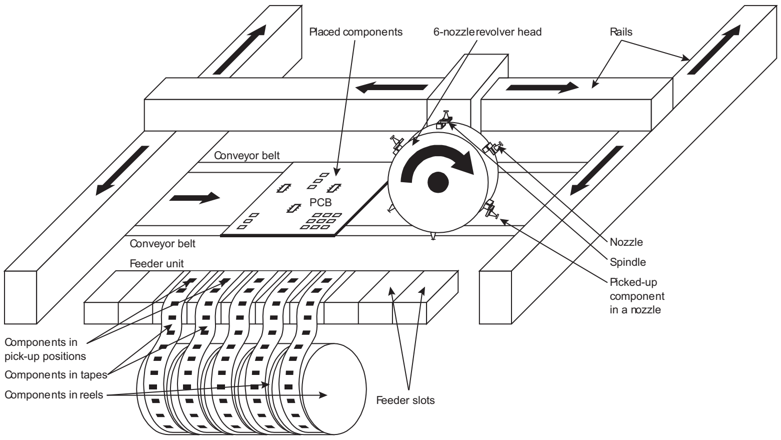

Figure 2.7: The scheme of a revolver head gantry machine.

some other type [13]. The machines have a lot of equivalent features but there are differing properties, also. However, all details of machine operations do not have to be known when solving subproblems of higher levels. For example, in order to solve the *job grouping* problem it may be sufficient to simply know whether the placement machine is able to place all the allocated components of the PCB job group, see Smed et al. [58] and Salonen et al. [53]. In some cases the time factors of the machines have to be approximated in order to manage a subproblem.

2.5.1 Pick-and-place machine

A special case of collect-and-place machines is the pick-and-place machine which is able to carry only one component at a time. The optimization problems of this machine type have been studied by Ball & Magazine [7] and Leipälä & Nevalainen [37], for example. Ball and Magazine decompose the problems of the production line into three parts that depend on each other. The pickup and placement problem for a single machine is solved by treating it as so called *rural postman problem* (which is an NP-hard problem [7]). Leipälä and Nevalainen solve the pick-and-place sequence problem for a fixed permutation of component tape reels in the feeder treating it as a *three-dimensional asymmetric traveling salesman problem*. For a fixed pick-and-placement sequence, Leipälä and Nevalainen propose a solution in which the feeder assignment problem is converted to a *quadratic assignment problem* [9]. Although the pure pick-and-place-type machine is already outdated, these problems exist also in current machinery. For example, in some collect-and-

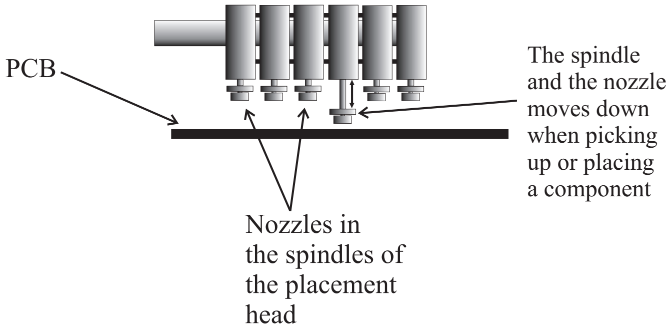

Figure 2.8: In-line head in which the spindles are side-by-side each other.

place machines there is a pick-and-place-head which is used for large components. Lately, the pick-and-place machines are studied by Hardas et al. [23].

2.6 Operations of a revolver head gantry machine

There are two types of collect-and-place machines, *in-line* and *revolver* head machines. The main parts of the revolver head machine are represented and named in Figure 2.7. In an in-line head machine *spindles* (and therefore the nozzles) are side by side in a row (see Figure 2.8). In revolver head machines the spindles are positioned in an annular way [3]. A revolver head is shown in Figures 2.7 and 2.9. The type of the placement head has elementary influence on the operation of the machine. For example, with the in-line head it is possible to perform a so called *gang pickup*, in which several nozzles grab a component from different slots simultaneously [41]. However, for a revolver head machine only the nozzle at the six o'clock position of the head is capable to pickup or place a component. There are also differences between the revolver machines. It is common that the revolver can rotate in one (forward) direction only. This property strongly affects the pickup and placement order.

Both in-line and revolver head machines are named as collect-and-place machines as a separation from pick-and-place and turret machines. The main parts of the revolver head gantry machine are presented in Figure 2.7. In this machine, the placement head is attached to a gantry or a buck mechanism. Using the step motors in the placement head and in the gantries the revolver head can travel on the (x, y)-plane. Because the machine has two independent motors the travel dis-

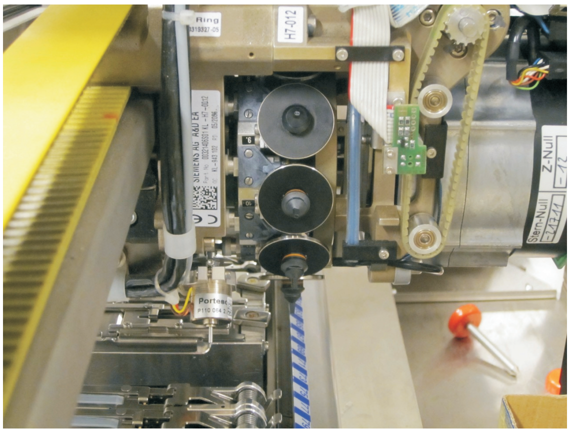

Figure 2.9: A revolver head and nozzles attached to it. The feeder slots marked with blue and white numbers are seen under the head.

tance between two points on the (x, y)-plane is defined by Chebyschev-metrics provided the motors operate at the same speed [7], i.e., the distance between two points (x_1, y_1) and (x_2, y_2) is given by $d_\infty = max\{|x_2 - x_1|, |y_2 - y_1|\}$. When the speeds are different the times of the x- and y-movements $t_x(|x_2 - x_1|)$ and $t_y(|y_2 - y_1|)$ should be used in the d_∞-formula. A step motor has a certain acceleration, too. The masses of the picked up components have an influence on the acceleration, and therefore to the travel times, and this should be taken into account in the most exact simulations. In addition, there is an own motor for rotating the revolver equipped with the spindles.

Nozzles that are used to grab and place the components are attached to spindles. Typically, there are about 6-30 spindles in the revolver head. Some machine models are capable of changing nozzles automatically using nozzle bank, see Figure 2.10. The machine has its own nozzle bank for its pick-and-place-head.

On the side (or both sides) of the machine there is a feeder unit which has storage slots for component tape reels, see Figure 2.11. The number of available feeder slots varies typically between 20-40 depending on the design of the placement equipment. On the left side of the Figure 2.11, there are two special tape reel holders that can be used to feed wide components which occupy more than one storage slot from the feeder unit. Note that one can put several component reels of the same component type in the feeder unit and in this way increase the effectiveness of the placement operations of frequently used components.

Figure 2.10: The nozzle bank of Siemens Siplace 80 F4 and nozzles in it.

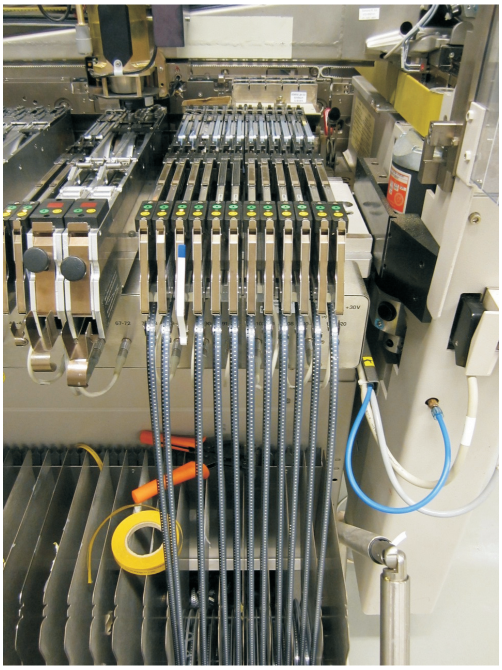

Figure 2.11: Component tape reels loaded into the feeder unit. There are two tape reel holders for wide component reels on the left.

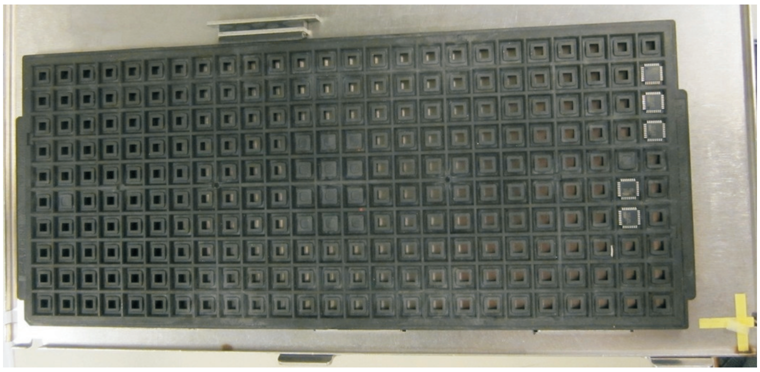

Figure 2.12: A waffle pack feeder that is used with a pick-and-place-head.

In some machine models it is possible to put even multiple component reels available on the same feeder slot using a special tape holder. In that case the components of different type are still supplied each by their own reel, one reel per type, but they are picked up from the same slot. In addition, there are machines in which components can be fed from a waffle pack tray (Figure 2.12) or from a plastic tube (Figure 2.13) using a vibration device (Figure 2.14) that brings them within the range of the placement head.

Nozzles are used to grab components. Because the size, mass and shape vary greatly between component types multiple nozzle types are needed. Usually, the component-to-nozzle compatibility is mentioned in the data that describes the PCB recipe. For the component of a certain type there can be, for example, three suitable nozzle types and the best placement accuracy is achieved only with the nozzle of the first type. The two latter can be used if the placement accuracy for the component has a greater tolerance. The placement head can be equipped with multiple copies of a certain nozzle type because the usage frequency of different nozzles may vary considerably.

The conveyor belt transports the empty circuit board (or incomplete if some of the components are already placed) on a table located in the center of the placement machine. Then, the machine starts to work cyclically following the instructions given in the control program. First, the revolver head moves on the feeder slots where it collects some components into the nozzles. As mentioned earlier, only the nozzle that points down (the nozzle that is at the six o'clock position) can pick up a component. After a component is picked up the revolver rotates the next suitable nozzle into the position in which it can grab the next component. While the revolver rotates, any tape reel in the feeder can move the tape forward (if necessary) so that it is possible to pick up the next component. At the same time the placement head can move on the (x, y)-plane above another feeder slot or stay motionless. This is possible because the machine has independent step motors for all these tasks.

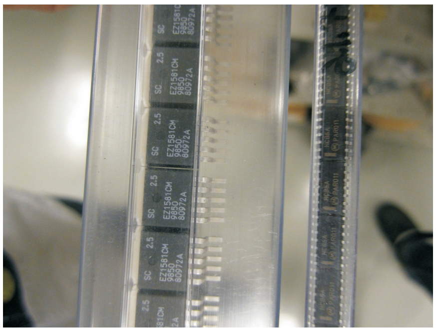

Figure 2.13: A plastic tube that is used with vibration feeder of Figure 2.14.

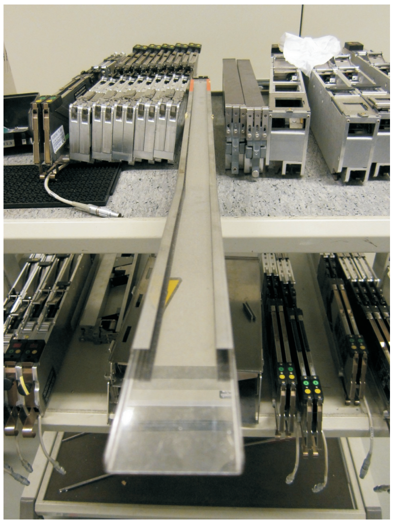

Figure 2.14: A vibration feeder device in which the components slide within the range of revolver head.

Once placement head has collected all the components determined in the control program from the feeder unit into its nozzles it moves on the camera (Figure 2.15) where the stances of the components are verified. For some machine models the camera is integrated into the placement head so that a separate visit to the camera station is not needed.

After the components are collected and their stances in nozzles are verified and corrected the placement head moves on the PCB and places the components on it one by one. Also in this phase, the revolver rotates a step after each placement to get the next nozzle armed with a component to point down. Again, the placement head moves on the (x, y)-plane to the next placement position, and part of the movement can be done at the same time the revolver is rotating. Finally, when all the nozzles are empty the revolver head returns back above the feeder unit. This cycle is iterated until all components determined in the PCB recipe are placed. The

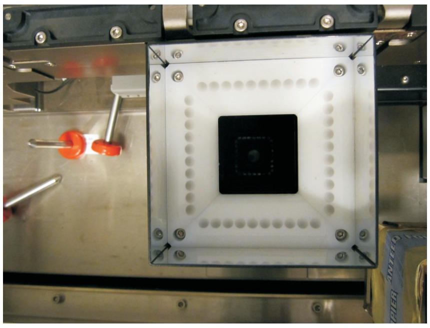

Figure 2.15: A camera unit for the pick-and-place head of Siemens Siplace 80 F4 that has to be visited separately during the placement process.

conveyor belt then transports the PCB forward on the production line and brings a new empty PCB on the placement table for the next iteration of the same cycle.

Chapter 3

Optimization for collect-and-place -machine

3.1 Combinatorial optimization and complexity

Many of the optimization problems in the control of a single placement machine are *combinatorial optimization (CO)* -problems [27]. In CO-problems discrete choices are made and the goal is to find an optimal solution from a finite or a countably infinite set of alternatives. The quality is measured by an object function, a *cost*, that gives a numerical value for the quality of solutions [40]. All the subproblems relating to the control of a collect-and-place machine should be solved jointly in order to guarantee an optimal solution of the joint control problem. However, many of the known problems that resemble the subproblems arising from the context of collect-and-place machines have been proved to be NP-hard, for example *quadratic assignment problem* and *travelling salesman problem*. In fact, several subproblems even in the hierarchical decomposition of a pick-and-place-machine are generalizations of NP-hard combinatorial optimization problems [62, 37, 22].

3.2 Single PCB type, single machine control problems

Intuitively, the aim is to always perform the assembly job as fast as possible so that the placement accuracy meets the quality requirements of the product. Applying the hierarchical decomposition, some of the subproblems can be solved optimally using some exact algorithm or a simple brute force method. For other subproblems heuristic methods are applied. Sometimes it is advantageous to solve the subproblems in turns, iteratively [37, 46]. Finally, the solutions of separate problems are combined and the whole solution builds up a feasible control program for the placement machine. The larger the batches of a PCB type are, the longer the production planning may last.

Next, the problems relating to the class of *1 PCB type - 1 placement machine* [28] are presented briefly. The machine type is a collect-and-place machine with either in-line or revolver head.

Nozzle problems

In the collect-and-place machine there are multiple spindles in which the component handling nozzles are attached. It depends on a component shape what kind of nozzle can be used to pick it up. The assembly of any PCB requires numerous different components that all have their own nozzle requirements. Often, the component can be handled with 1-3 nozzles of different types. The placement accuracy may vary by the nozzle type and the most suitable nozzle is preferred. The nozzle grabs on a component using vacuum. Next, optimization problems arising in the nozzle usage are described.

- **What kind of nozzles at least should be selected into the placement head?** Answering this is straightforward if there is a unique nozzle requirement for every component. The problem becomes harder if there are multiple alternatives of suitable nozzles for each component. Note, that the solution may be different depending on the objectives. Two natural objectives are minimization of the cycle time and the highest possible quality. [47, 48]

- **Which nozzle types are duplicated in the placement head?** If the number of the mandatory nozzle types is less than the nozzle capacity of the placement head, which types of nozzles are selected into the free spindles? [33, 47, 48, 49]

- **In which order the nozzles should be attached to the placement head?** The feasible order of the nozzles in the placement head depends greatly on the permutation of the component tape reels in the feeder unit. With revolver head it may or may not be possible to rotate the revolver in both directions which has effect on the optimal nozzle assortment. This problem is discussed by Pyöttiälä et al. [46] for a revolver head machine. The preferences for considering this problem differ significantly between in-line and revolver machines. Further, there may be empty spindles in the placement head because some components are too large to fit into the placement head if the nozzles were in adjacent spindles.

- **How are nozzle changes organized if they are required during the placement job?** Some machine models can change nozzles automatically, while others need a manual procedure. Typically, the time spent in nozzle changes is relatively high and they should therefore be avoided in general. In a manual change, the operation of the placement machine is stopped for a relatively long period, and therefore manual changes are not possible in mass production. The placement head may utilize automatically [12] and manually changeable nozzles at the same time. In some in-line heads a *gang nozzle change*, where multiple nozzles are changed at a time, may be possible. Nozzle change operations of in-line machine have been studied by Crama et al. [14], Sun et al. [59] and Park et al. [42]. The problem of nozzle changes is closely related to *tool management problem* studied by Crama et al. [15]. The nozzle changes for collect-and-place machine are studied by Knuutila et al. [34].

- **What types of nozzles are selected into the nozzle bank of the machine?** If nozzle changes are required by the job what kind of combinations of nozzles are enabled? [34] If the nozzle changing is automatic the capacity of the nozzle bank is limited. In manual change the capacity of the bank is directed by economical issues.

If all components of the PCB can be picked up and placed using a nozzle of the single type it is straightforward to select the nozzle of the same type into all spindles. This is called a *universal nozzle* and its use simplifies many subproblems and makes it easier to find suitable solutions. For collect-and-place machines, the approach of universal nozzles is used, for example, in [22, 26, 42, 35, 59], while the requirements of various nozzle types have been taken into account in [14, 62, 36, 63, 33, 48, 32, 45, 46].

Feeder problems

The feeder unit is the part that supplies components for the placement head of the machine. Usually, the preparation of the feeder unit is done manually by a human operator. There are numerous types of feeder units depending on the manufacturer of the placement machine. The feeder unit is positioned on the side of the machine. There can be more than one feeder unit in a placement machine. Often, the components are supplied in component tape reels, but different technologies are described in Section 2.6. In some machine models, multiple feeders can be changed at a time by using a changeover table. The component tape reels can be prepared into one changeover table while another is attached in the use of the machine. The tables can be swapped when needed and the placement machine offline time remains then short. The tape reels can also be changed one at a time. The feeder unit has a maximum capacity of feeder slot positions that can keep the component tape reels. The reels of large components may occupy more than one feeder slot which otherwise could hold, for example, two reels of narrower components. The following problems concern feeders.

- **In which slots the component tape reels are positioned?** Generally, this problem and its variations are called as the *feeder assignment problem*. The problem depends on the nozzle types and their order in the placement head but also on the placement order and positions of the components. Even the type of the used placement head has an effect on the optimal solution of this problem. For in-line-machines, it is advantageous to favor gang pick-ups [59], while with revolver head it is important that the component types, which lie near each other on the PCB, are neighbors in the feeder unit [46]. The latter requires that the placement order is organized by assigning the components to be placed near each other into the same load of the placement head. The feeder assignment problem of the collect-and-place machines is studied in [14, 36, 22, 45, 46, 35, 38]. The corresponding problem for the pick-and-place machine is discussed in [37, 25, 39]. In the pick-and-place machine the adjacency of the components on the PCB does not affect as much as in the case of collect-and-place machine.

- **What types of components are duplicated in multiple slots of the feeder unit?** If components of a certain type are consumed considerably it may be advantageous to put them available in more than only one feeder slot. This can give benefits also in organizing the gang pickups. In addition, the duplication of the component pickup places gives ability to change feeder reels in the duplication slots without stopping the production in some machine models. For turret type machines, duplication of component reels is discussed by Crama et al. in [13].

- **What component types should be assigned to the feeder slots keeping two or three component reels in the slot position?** In addition, if there are only a limited number of feeder devices of double or triple reels where are they assigned in the feeder unit? Using this type of devices still needs further research. Note, that these special feeder devices are available for some machine types, only.

Pickup and placement problems

The optimal order in which the components should be picked up and placed depends strongly on the arrangement of the component tape reels in the feeder unit, the placement points of the components on the PCB and the types of nozzles in the placement head. The machine type may force the pickup and the placement orders to be identical. If the manufactured PCB is of small dimensions, the movements done in the pickup phase on the feeder may dominate the assembly cycle time. It is then necessary to concentrate on minimizing the length of the movements on the feeder area.

- **How to divide the component placements into smaller partitions that fit in the placement head at once?** These partitions are called the *subjobs* of a PCB assembly job or *subtours*. The number of components that the placement head can keep in its nozzles at the same time is limited by its nozzle capacity. In addition, the components have to be compatible with the nozzles types. The methods for constructing this kinds of partitions is studied, for example, in [14, 22, 35, 46].

- **In which order are the components picked up from the feeder unit?** With in-lines machine several components can be picked at a time if gang pickups [11] are possible. Constraints for a gang pickup are mostly related to the match of distances of nozzles in the head and the corresponding distances of the component reels in the feeder unit. If the machine has a revolver head, the limits in the direction of rotation of the revolver may affect the ordering of the pickups. Further, it is typical that the revolver can rotate at most 360° in a single pickup tour. Methods for the pickup ordering are proposed in [14, 22, 35, 32, 46].

- **In which order are the components placed on the PCB?** Again, the machine design may allow that the placement order can differ from the pickup order [32] or that these two orders must be identical [22, 35, 46]. Methods for placement ordering are proposed in [10, 30], too.

In the above problem descriptions the pickup and placement ordering together with partitioning the placements into subjobs define the mutual order of the partitions, too. Alternatively, the mutual order of the subtours and the pickup and placement order inside each subtour (of a task block) could be raised as problems like done by Grunow et al. [22]. In addition, if there are multiple nozzles suitable for each component type, the usage policy of the nozzles has to be solved [46]. If some component types can be picked up from multiple slots of the feeder unit the exact position of each pickup has to be defined.

3.3 Research problem

In Section 3.2, the problems appearing in the control of a single collect-and-place machine were described briefly. The most fundamental research problem of the present thesis can be defined as follows.

- **How to solve in a feasible way the problems arising in the control of a single collect-and-place machine in the case of a single PCB type when quality of the solution is measured by the time consumed to the assembly job?**

When the production time, or assembly cycle time, of a single PCB is minimized the amount of manufactured circuit boards in a fixed period is maximized. In theory, a naive algorithm that always finds the optimal configuration for the production (by exhaustive search) can be written, but this kind of method works for the problems of unrealistically small size, only. Thus, heuristic methods have to be used for most of the subproblems of hierarchical decomposition. In order to compare the composed configurations and the control programs generated by solution algorithms the measurement methods have to be defined. Often, the factor to be minimized is the length of head movement, the time consumed for production or its simplification like the number of pickup rounds [48]. In addition, the following adjunct question should be answered:

- **What kind of measuring techniques can be used for comparing the efficiency of different machine configurations or the pickup and placement ordering?**

In general in the PCB assembly, the essential object function to be minimized is the time consumed in production. However, in some cases the use of time for measurements may be impossible because of the technical properties of the solution method. For example, using hierarchical decomposition of the above control problems, it is impossible to calculate the exact assembly cycle time when the decision of the nozzle selection is made because the placement order cannot be determined yet.

3.4 Measuring the quality of candidate solutions

For each configuration and control program for manufacturing a single PCB with a single placement machine the efficiency of the composed configuration and operation sequence have to be calculated in order to compare them. The most realistic values for these setups are achieved only by uploading the control program into the placement machine, setting the component tape reels into the feeders and performing the job by the machine.

It is sometimes possible that the machine performs the job in a simulation mode in which the component placements are not done for real. The placement time can then be observed from the machine clock. When doing this, the machine has to run in the speed which is required for the needed precision of the placement positions. If the operation speeds of the motors are raised to their upper limits the precision may suffer too much.

The reliability of the final product depends often on the precision of the placements. The acceptable amount of failed products is defined by economic arguments. However, in this thesis, the placement precision, the requirements for quality of the manufactured product and direct economic questions are not discussed in detail. A certain quality is taken as a premise which allows the placement machines to operate by a certain speed which is then used in the calculations of the assembly cycle time. The assembly cycle time is generally accepted as an important measurement method. The faster a single PCB is produced, the higher is the throughput of the machinery. Thus, the investment yields the highest possible profit.

In practice, simulations with real placement machinery cannot be used for comparison of the control programs because it would take too much time and would be too expensive. Instead of the actual placement machines, an exact simulation or simpler approximations are used. Kallio et al. [29] have proposed a generic model for simulation that is based on measured time factors of real machinery. Their model achieves a good accuracy. The time factors of the parallel operation of several machine components has to be taken into account in measuring and modeling. For example, in calculation of the movement times for the placement head the use of Chebychev-metric may be the proper choice.

The time used to manufacture a single copy of a PCB of a certain type is called *assembly cycle time (ACT)*. The use of as precise and detailed time calculations as possible increases the reliability of the results of control program comparisons. The times obtained from the simulations can then be validated with a real machine. Unfortunately, the manufacturers of placement machines do not provide accurate information of the speeds of their products in various operation situations. However, a reliable estimate for the completion of a batch of PCBs is always important to get and one has to cope with more inaccurate simulations when solving the different subproblems of the machine control optimization.

Instead of ACT calculations one may compare the different candidate solutions indirectly by calculating the length of movements made by the placement head. In a purely distance based comparison, the resulting values are somewhat inaccurate but still applicable. For example, it is then not taken into consideration that the placement head could move a short distance while the revolver rotates or the feeding device of the component tape reel moves the tape forward.

In some cases the details of a machine operation are not known accurately enough that an exact simulation would be possible. The simulation can then be implemented only partly and the time consumed or distance traveled by the rest

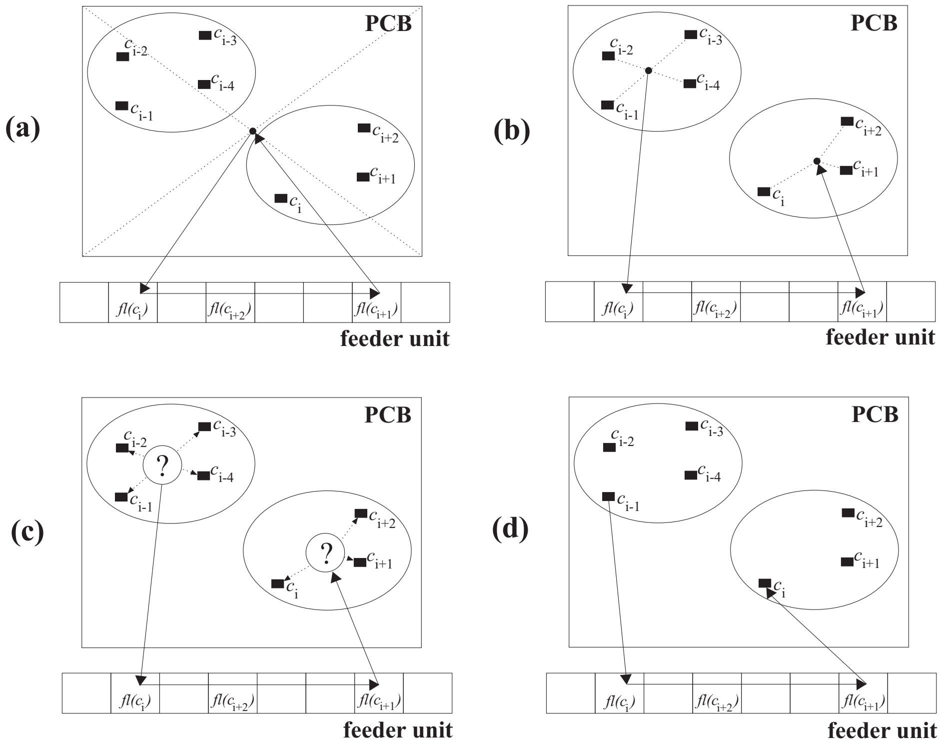

Figure 3.1: Different methods for estimating and defining the distance traveled between the feeder and the PCB. A component placement point on the PCB is denoted with c_j and the feeder location of c_j is given by $fl(c_j)$.

of the operations must be approximated [32]. For example, four different ways to approximate the distance traveled between the feeder and the PCB are given in Figure 3.1. Here, either the exact sequence of placement operations is not known or an estimate for the distance is sufficient. In case (a) of Figure 3.1, the center of the circuit board is used as the start and end point of a tour. In (b) the exact start and end points are replaced by the average values of the placement points for every subtour. The points can be replaced with randomly selected points of the subtours like in (c). In case (d) only the first and last placement points are known, but the exact distances can be calculated based on these values. Notice, that the pickup sequences in Figure 3.1 present sequences where the typical limitations for the revolver head machine are considered.

In the hierarchical decomposition, the construction of the solution for the whole machine has to begin from some simple starting point, for example, the selection of the nozzles into the spindles. In this situation, it is possible that the final positions of the component tape reels in the feeder unit are still not known. Then the comparison of the different nozzle settings can be based on the number of the operation cycles of the machine for random pickup sequences as done by Pyöttiälä et al. [48].

In hierarchical problem decomposition any subproblem can be solved again iteratively after the independent or higher-level problems have been solved once. It is also possible to change the object function before the second time of solving. Often, this improves the whole solution that is built from the solutions of subproblems. For example, the problem of selecting the nozzles can be solved again after the positions of the components on the feeder unit and the pickup and placement sequences have been determined. Instead of using the number of operation cycles ACT can then be used in the comparisons of different nozzle assortments.

A model for calculating the assembly cycle time

The following notation can be used for calculating the ACT of a collect-and-place machine equipped with a revolver head. It is assumed that the machine configuration and the machine control program have been fixed. The notation summarizes the various notations for the different problems discussed in the publications of this thesis.

Definition 1.1. ***PCB Assembly Setup (PASSE).*** Let us denote

- $CT = \{ct_1, \ldots, ct_m\}$, the set of component types.
- $C = \{c_1, \ldots, c_n\}$, the recipe of a PCB, *i.e.* the set of components and their locations on the PCB, where $c_i = (t_i, (x_i, y_i))$ such that $t_i \in CT$, $(x_i, y_i) \in R$, and $R \subset \mathbb{R}^2$. R stands for the rectangular area of the PCB.
- $NT = \{nty_1, \ldots, nty_l\}$, the set of nozzle types.
- $cl : C \mapsto R$, a function giving the location of a component c_i on the PCB; $cl(c_i) = cl(t_i, (x_i, y_i)) = (x_i, y_i)$.
- $ct : C \mapsto CT$, the type of each component c_i; $ct(c_i) = t_i$.
- $fl : CT \mapsto \mathbb{R}^2$, the location of the component of a certain type in the feeder unit (in other words, fl defines the feeder setup).

- $nt : CT \mapsto NT$, the nozzle requirement of a certain component type in the PCB recipe.
- $\alpha = (n_1, \ldots, n_k)$, the nozzle arrangement of the revolver head, *i.e.* so-called *arm*, where $n_i \in NT$ for $i = 1 \ldots k$ and k is the size of the head.
- $o^6(\alpha)$ gives the index of the nozzle that is operable (points at 6 o'clock) in α. Note that, $1 \leq o^6 \leq k$ and indexing is cyclic.
- (x_{pp}, y_{pp}) is a special *park position* from which the revolver head starts and where it returns at the beginning and in the end of a placement job of a certain PCB.
- $d((x_1, y_1), (x_2, y_2)) = max(|x_1 - x_2|, |y_1 - y_2|)$, the (Chebychev) distance between two locations, (x_1, y_1) and (x_2, y_2).
- $tt : \mathbb{R} \mapsto \mathbb{R}$, the time required by the revolver head to *travel* a certain distance.
- $re : C \times \alpha \times o^6 \mapsto \mathbb{N}$ gives for a given arm α and 6 o'clock -position the number of rotation steps required to turn the revolver head so that the next suitable empty nozzle can pick up a certain component c.
- $rf : C \times \alpha \times o^6 \mapsto \mathbb{N}$ gives for arm α and 6 o'clock -position the number of rotation steps required to rotate the revolver head so that the next nozzle holding component c can place it.
- $rt : \mathbb{N} \mapsto \mathbb{R}$, the time required by the revolver head to rotate a certain number of steps forward.
- pt, a constant time for a single pickup or placement operation.

A permutation of the elements of the PCB recipe C is called a *job* W and it determines the order in which the components are picked up from the feeder unit and placed on the PCB. It is a natural assumption for a revolver head (but not for in-line heads) that the orderings for pickups and placements are identical. Each element of set C occurs exactly once in W. Job W must be partitioned into p separate subjobs or tasks, $W =< W_1 \cdot W_2 \cdots W_p >$ because the whole job of a practical size does not fit into the arm at once. (If pickup and placement orders may differ in a placement machine type the pickup order related to W is denoted by W^o.)

Definition 1.2 Let W be a job, α an arm of k spindles (i.e., places for keeping nozzles), and let $W_i = (c_1^i, c_2^i, \ldots, c_s^i)$ be a partition of W, where c_j^i is the jth placement of the ith partition of W and $s \leq k$. We say that α *can pick up* W_i if and only if $nt(W_i) = (nt(c_1^i), nt(c_2^i), \ldots, nt(c_s^i))$ is a *subsequence* of α. We then say that α is *feasible* for W_i.

Definition 1.3 Given arm α and job $W =< W_1 \cdot W_2 \cdots W_p >$, we say that α *can execute* W if and only if α can pick up each W_i. We then say also that α is *feasible for* W.

Using the previous definitions the following three object functions can now be defined.

Definition 1.4 [The number of pickup phases (NPP)]. Let us suppose that job W, arm α, and functions ct, nt are given, and α can execute W. Then, for any partition for $W =< W_1 \cdot W_2 \cdots W_p >$, such that α can pick up each W_i separately, the number of partitions

$$NPP(W, \alpha, ct, nt) = p \tag{3.1}$$

is called the *number of pickup phases*.

Definition 1.5 [The length of a pickup path (LPP)]. Suppose that we are given a fixed job W of length n and the associated pickup order W^o, arm setting α of size k and the functions cl, ct, fl, nt and d as defined in PASSE. Further, it is assumed that α can execute W. The length of pickup path of W^o is then

$$LPP(W^o, \alpha, \ldots, d) = MOV + \sum_{i=1}^{|W^o|-1} d(fl(ct(c_i)), fl(ct(c_{i+1}))) \tag{3.2}$$

where MOV represents the length of the movements between the feeder and the board. Different possibilities to define MOV are presented in Figure 3.1.

Definition 1.6 [The assembly cycle time (ACT)]. Suppose that we are given a fixed job W of length n, arm setting α of size k and the functions cl, ct, fl, nt, d, tt, re, rf, rt, pt as defined in PASSE. Further, it is assumed that α can execute W in p pickup and placement tours (subjobs) and that the revolver head (arm) is initially located at the feeder park position (x_{pp}, y_{pp}) of the machine. The assembly cycle time of W is then

$$ACT(W, \alpha, \ldots, pt) = cost_0 + \sum_{i=1}^{p-1} \left(cost_i \right) + cost_p, \tag{3.3}$$

where $cost_0$ is the time for traveling from the park position of the placement head to the feeder slot of the first component of W, $cost_i$ is the time for the ith subjob and $cost_p$ is the time for the last subjob and traveling from the last placement to the park position.

These costs are of the form

$$cost_0 = tt(d((x_{pp}, y_{pp}), fl(ct(c_1^1)))), \tag{3.4}$$

$$\begin{aligned} cost_i = pt + &\sum_{j=1}^{|W_i|-1} (cost_{i,j}^{pickup}) + cost_i^{travel\_PCB} + pt \\ &+ \sum_{j=1}^{|W_i|-1} (cost_{i,j}^{place}) + cost_i^{travel\_feeder}, \end{aligned} \tag{3.5}$$

and

$$\begin{aligned} cost_p = pt + &\sum_{j=1}^{|W_p|-1} (cost_{p,j}^{pickup}) + cost_p^{travel\_PCB} + pt \\ + &\sum_{j=1}^{|W_p|-1} (cost_{p,j}^{place}) + tt(d(cl(c_{|W_p|}^p), (x_{pp}, y_{pp}))). \end{aligned} \tag{3.6}$$

Equation (3.4) gives the travel time of the placement head from the park position to the feeder location of the first component to be placed. Equation (3.5) gives the time of the pickups and the placements for the ith subjob of W and (3.6) states the same for the last subjob. This subjob ends at the park position.

More accurately, $cost_i$ (3.5) consists of the pickup time of the first component of W_i, the time consumed in picking up all the components of W_i, the travel time from feeder onto the PCB, the placement time of c_1^i, the time for placing the other components of W_i, and finally the travel time from the PCB back to the feeder area.

Parameter pt stands for the *pickup* and *placement* time. In Equation (3.5), the first occurrence of pt is the pickup time of the first component and the second occurrence stands for the placement time of a component.

Furthermore, the cost of a single pickup of the j'th non-first component in the i'th partition of W is

$$cost_{i,j}^{pickup} = max\Bigg\{tt\Bigg(d\Bigg(fl(ct(c_j^i)), fl(ct(c_{j+1}^i))\Bigg)\Bigg),$$

$$rt\Bigg(re(c_{j+1}^i, \alpha, o^6(\alpha))\Bigg)\Bigg\} + pt. \quad (3.7)$$

The cost of traveling from the feeder slot location of the last picked up component of the i'th partition to the placement location of the first component of the i'th partition is

$$cost_i^{travel\_PCB} = tt(d(fl(ct(c_{|W_i|}^i)), cl(c_1^i)). \quad (3.8)$$

The cost of a single placement of the j'th component in i'th partition of W is

$$cost_{i,j}^{place} = max\Bigg\{tt\Bigg(d\Bigg(cl(c_j^i), cl(c_{j+1}^i)\Bigg)\Bigg),$$

$$rt\Bigg(rf(c_{j+1}^i, \alpha, o^6(\alpha))\Bigg)\Bigg\} + pt. \quad (3.9)$$

Finally, the cost of traveling from the placement location of the last component of the i'th partition to the feeder slot location of the first component of the $(i+1)$'th partition is

$$cost_i^{travel\_feeder} = tt(d(cl(c_{|W_i|}^i), fl(ct(c_1^{i+1})). \quad (3.10)$$

Note, that revolver head rotations and (x, y)-travels are simultaneous and therefore one has to consider which one of travelling and rotating takes a longer time.

The term $cost_{i,j}^{pickup}$ in Equation (3.7) is the maximum of the travel time between the feeder slots of two consecutive components in a job, the time that it takes to rotate the head, summed up with the pickup time of one component (pt). Here i stands for the partition ordinal and j is the ordinal of the component in the i'th partition. The term $cost_{i,j}^{place}$ (Equation (3.9)) states the same for placements as Equation (3.7) for the pickups. Equations (3.8) and (3.10) stand for the traveling time between the feeder and the PCB (and vice versa).

The definitions of NPP, LPP and ACT follow definitions given in the publications of this thesis. NPP is used in [33, 48], LPP in [32] and ACT in [45, 46]. The definitions of the different measurement methods with a review presented in Section 3.4 answer the adjunct question of the research problem stated in Section 3.3.

3.5 Problem definitions

In Section 3.2 we briefly presented the problems emerging in printed circuit board assembly when a single PCB type is produced using a single placement machine. Next, we give the formal definitions of the problems discussed in the publications of this thesis. The naming of the problems has been changed more consistent in some cases. The original name appearing in the publications is given in parentheses after the new name.

Problem 1. Minimal number of pickup phases problem (MNPP).
(Also called as the minimal length pickup sequence problem (MLP) [33].)
Suppose that recipe C, job W, feasible arm α of size k, functions ct and nt as in definition PASSE are given. Find a partition into subjobs $W = < W_1 \cdot W_2 \cdots W_p >$ such that $NPP(W, \alpha, ct, nt)$ is minimal. The minimal number of the pickup phases is denoted by $MNPP(W, \alpha, ct, nt)$.

Problem 2. Optimal nozzle selection problem (ONS). [33, 48]
Suppose that recipe C, job W, arm capacity k, NT, at least k nozzles of every required type in a nozzle bank, functions ct and nt as in definition PASSE are given. Especially, $k \geq |NT|$. Assign k suitable nozzles (i.e. fix the nozzle selection) into arm α so that α is feasible and that the minimal number of pickup phases $MNPP(W, \alpha, ct, nt)$ is minimal.

Problem 3. Length of a pickup path for single phase problem (LPPS).
(Also called as the single pickup event problem (SPE) [32])
Suppose that recipe C, job W, arm α of size k which can pick up W (at once) and functions cl, ct, nt, d as in definition PASSE are given. Find a pickup sequence W^o and compute the associated length of the pickup path $LPP(W^o, \alpha, \ldots, d)$.

Problem 4. Minimal length of the pickup path problem (MLPP).
(Also called as the minimal pickup sequence problem (MPS) [32])
Suppose that recipe C, job W, feasible arm α of size k and functions cl, ct, nt, d as

in definition PASSE are given. Find a pickup sequence $W^o =< W_1^o \cdot W_2^o \cdots W_p^o >$ for W such that the length of the pickup path $LPP(W^o, \alpha, \ldots, d)$ is minimal.

Problem 5. Optimal Feeder Assignment Problem (OFA). [45, 46] Suppose that recipe C, job W and its partition $W =< W_1 \cdot W_2 \cdots W_p >$, feasible arm α of size k and functions cl, ct, nt, d, tt, re, rf, rt, pt as in definition PASSE are given. Assign component types to feeder slots (function fl) so that assembly cycle time $ACT(W, \alpha, \ldots, pt)$ (equation (3.3)) is minimized.

Problem 6. Optimal Placement Sequence Problem (OPSE). [46] Suppose that recipe C, component locations in feeder fl, feasible arm α of size k and functions cl, ct, nt, d, tt, re, rf, rt, pt as in definition PASSE are given. Find job W^* (a permutation of the elements of C) and its partition $W^* =< W_1^* \cdot W_2^* \cdots W_p^* >$ so that assembly cycle time $ACT(W^*, \alpha, \ldots, pt)$ is minimal (over all possible orderings).

Problems 1 and 2 are the nozzle problems presented in Section 3.2, Problem 5 is one of the feeder problems of Section 3.2, and Problems 3, 4 and 6 represent the pickup and placement sequence related problems of Section 3.2.

3.6 About the solution methods, tests and results

For the problems MNPP, LPPS and MLPP defined in Section 3.5 feasible exact algorithms exist and are proposed in the publications [33] and [32] of this thesis. For the rest of the problems, ONS, OFA and OPSE, the heuristic algorithms are proposed and compared in publications [48], [45] and [46]. These algorithms include both new and previously published methods.

3.6.1 Types of heuristic approaches

Since many of the problems proposed in 3.5 are hard, solutions cannot be found by exact methods for realistic sized problems. Optimal solutions can be found only for small instances using exact methods. The running time of exact algorithms quickly becomes impractical when the problem size increases. Typically, heuristics and different versions of local search are developed and used to solve the problems of realistic size. These methods are suitable for optimization problems emerging in the control of placement machinery, also. In these control problems, quality and completeness are sacrificed as a trade-off for speed when heuristics are applied.

The non-exact methods in this thesis fall into the following categories:

1. The heuristics first analyzes the entire input data and then uses the gathered information to the construction of a solution.

2. The heuristics scans through input data and immediately starts to construct the solution more or less greedily and possibly applying some randomization.

3. The local search algorithm starts with an initial solution and a quality function and tries to improve the solution using local search. (This may include random choices, also.)

An algorithm in the last category can get its initial solution from heuristics of type 1 or 2. The methods proposed in the literature mostly fall in these categories, also. For example, the popular *genetic algorithm* can be seen as a kind of local search with random choices.

3.6.2 About the data in experimental tests

To make sure that the proposed heuristics and other non-exact methods are applicable and feasible, comprehensive experimental tests are needed. The reliability of the tests depends on the used test data. For exact methods, the tests are needed to certify that the running times of their implementations are acceptable for problems of practical size, but also to clarify the effect of the variable parameters.

The best practice to organize the experimental tests is to use data from real-world PCB production. When using realistic data one has to ensure that it does not originate from any special case.

The data from the real-world PCB production is used in the test runs of Publications 4 and 5. In Publication 5, it is noted that the great diversity of the test cases hides more or less the quality differences of results of the proposed heuristics for OFA and OPSE. It was argued that the differences could be disclosed by running the tests with more homogeneous data which is partially artificial. This data was generated by deconstructing the information from the original PCBs and then constructing a great number of homogeneous PCBs which dimensions, the number of components and other properties are close to each other. The construction was done randomly by applying certain, justifiable limits.

As contrast for the real-world data, fully artificial test data are used in the experimental tests of Publications 1, 2 and 3. In Publication 1, the simple random data is used. However, if the test data have been generated randomly from an uniform distribution of different values the heuristics that generate their solutions in the same way may have a certain advantage. In Publications 2 and 3, a more sophisticated method is therefore applied. Specialized *Markov-models* are able to generate random data that imitates, for example, the occurrences of component types on the PCB better than the simple random data does. The generation of the test data by Markov-models is discussed in Publication 2 and [44].

3.6.3 Main results of thesis

Next, the main results of the thesis are summarized briefly.

For Problem 1, MNPP, a greedy algorithm is proposed in [33]. Although the algorithm is quite simple and fast (time complexity $\Theta(|W|k)$, it is proved in [33] that it gives a minimal number of pickup phases. This simple result has significance because it enables the study of other subproblems from a reliable basis. The studying of the problem gives a better understanding of the nature of the subproblem, for example, in Figure 3.2 it is illustrated how increasing the arm size affects the minimal number of pickup phases. The proposed algorithm for MNPP is fast enough for practical purposes, too.

Problem 2, ONS, is studied in [48]. Three different heuristics are proposed for Problem 2, ONS, and their solutions are compared against each other, and, for smaller problem instances, against the optimal solution found by a brute force method. A simple and fast local search is used for the improvement of the results of the heuristic procedures. The two heuristics that were based on frequencies of the nozzle requirements of the job W outperformed the third heuristic in which the nozzle selections were based on the analysis of the lack of suitable nozzles. In the experimental tests with the hardest problem instances generated, the frequency-based heuristics gave about 10% greater numbers of pickup phases, as illustrated in Figure 3.3. After the local search step was applied on the solutions of the frequency-based methods, the means of the numbers of pickup phases were only 2.6% greater than the mean of optimal solutions found using brute force.

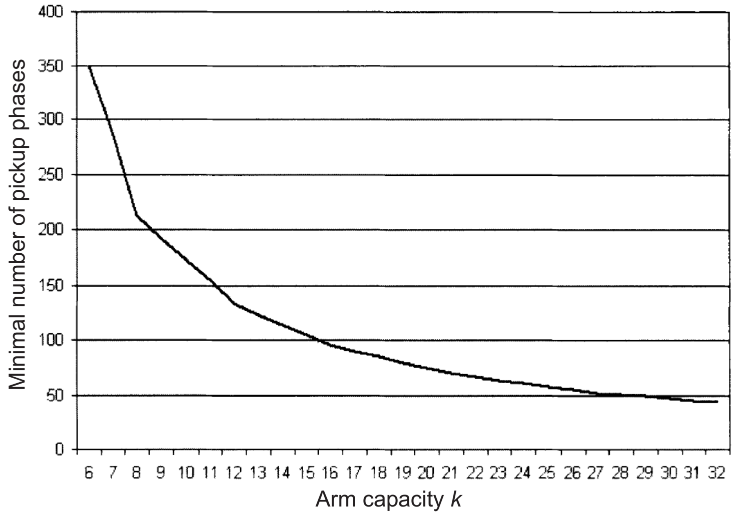

Figure 3.2: Minimal number of pickup phases as a function of the arm capacity k (fixed, artificially generated job W, $|W| = 1024$, $|NT| = 4$) [33].

Thus, the frequency-based methods with local search are applicable in practice because of the quality of their results and their fast execution.

Problems 3 and 4, LPPS and MLPP, are solved in [32] using exact methods. The overall length of the pickup sequence for a placement job W and arm contents α are minimized using the *dynamic programming* approach. The decisions include finding an optimal partitioning of W into subjobs and defining the optimal pickup order of the components in each subjob. Dynamic programming applies to this problem context well. Optimal partitions are constructed from optimal partitions of a smaller size. The iterations of same calculations are avoided by memorizing the optimal partitions for all possible subjobs. The running time of the solution algorithm is feasible even with the job length of 3000 components, see Figure 3.4.

The Problem 5, OFA, is studied in [45] and [46]. The proposed solution methods that minimize the ACT are based on heuristics. The results of the methods were compared to each other and to a theoretical lower bound. The heuristic algorithm that turned out to be the best is proposed in [46] (named as *feeder assignment by best k*). It analyzes the neighborhood of the component placement locations in a PCB recipe and derives the solution for the feeder assignment problem from that information. Again, the method is fast and simple. However, it yields good solutions and outperforms the methods proposed in the literature for a collect-and-place

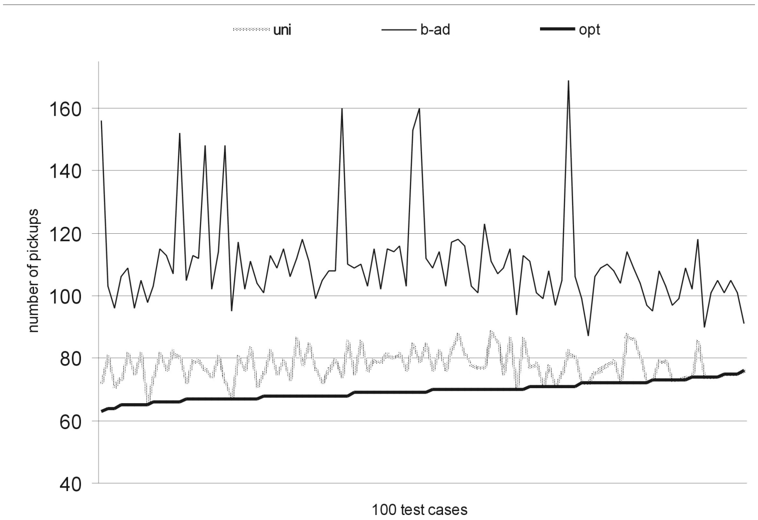

Figure 3.3: Heuristics based on uniform distribution of nozzle requirements (*uni*) and on the analysis of the most needed lacking nozzle type (*b-ad*) compared against each other and optimal solutions when solving ONS. The results of 100 test cases are sorted into an ascending order of the number of pickup phases of optimal solutions. (Parameters in test cases: $|W| = 400$, $k = 12$, $|NT| = 6$) [48].

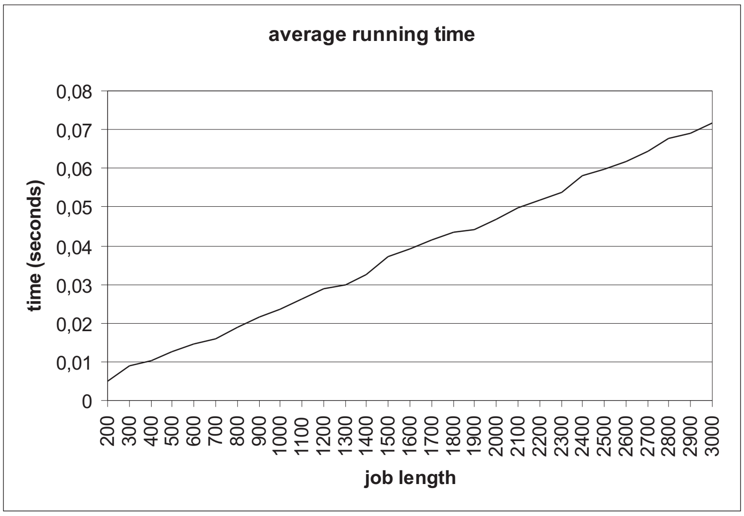

Figure 3.4: The running times of a dynamic programming algorithm for MLPP as a function of the job size. The results are averages of 100 repetitions for each artificially generated test job. (Parameters in test cases: $|CT| = 20$, $|NT| = 7$, $k = 14$) [32].

machine. The experimental tests were based on realistic data and an accurate modeling of the operation of the revolver head machine. The statistical descriptive and the results of ANOVA of the ACT comparisons are shown in Table 3.1. Statistically significant differences were found in [46] (ANOVA, Sig. 0.000), see row 5 in Table 3.1(c) for details.

For solving the Problem 6, OPSE, four constructive heuristics were proposed. Each of these heuristics are composed of the following three components or phases.

1. **Select the starting component.** Select *a starting component* from the set of *unassigned* components (PCB recipe) that have not yet been attached to any partition of the placement sequence. Attach the starting component to the partition that is currently being constructed, mark the nozzle which can pick it up as occupied and remove the component from the set of unassigned components.

2. **Collect more components.** Collect *more* unassigned *components* from the PCB recipe which fit into the free nozzles of the arm. Remove the selected components from the set of unassigned components and keep the accounting

Table 3.1: Results of ANOVA when comparing the ACTs of five heuristics for OFA problem. [46]

a)

Descriptives

ACT

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
1	500	43,47673	2,22856550	,09966448	43,2809120	43,6725394	37,35805	51,14267
2	500	43,80643	2,29719824	,10273383	43,6045859	44,0082742	37,24749	51,91672
3	500	43,70770	2,31718253	,10362755	43,5040995	43,9112997	36,31631	51,15767
4	500	43,79085	2,25892220	,10102207	43,5923653	43,9893274	37,33640	50,71047
5	500	42,00379	2,11216726	,09445899	41,8182068	42,1893795	35,96975	48,33475
Total	2500	43,35710	2,34504953	,04690099	43,2651302	43,4490678	35,96975	51,91672

b)

ANOVA

ACT

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1179,352	4	294,838	58,553	,000
Within Groups	12563,292	2495	5,035		
Total	13742,644	2499			

c)

Multiple Comparisons

Dependent Variable: ACT
Tamhane

(I) alg	(J) alg	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
1	2	-,32970436	,14313367	,195	-,7313148	,0719061
	3	-,23097387	,14377649	,683	-,6343885	,1724407
	4	-,31412067	,14191007	,240	-,7122972	,0840559
	5	1,47293250*	,13731537	,000	1,0876456	1,8582194
2	1	,32970436	,14313367	,195	-,0719061	,7313148
	3	,09873049	,14592090	,999	-,3106997	,5081607
	4	,01558369	,14408226	1,000	-,3886878	,4198552
	5	1,80263686*	,13955909	,000	1,4110508	2,1942229
3	1	,23097387	,14377649	,683	-,1724407	,6343885
	2	-,09873049	,14592090	,999	-,5081607	,3106997
	4	-,08314680	,14472086	1,000	-,4892104	,3229168
	5	1,70390637*	,14021830	,000	1,3104694	2,0973434
4	1	,31412067	,14191007	,240	-,0840559	,7122972
	2	-,01558369	,14408226	1,000	-,4198552	,3886878
	3	,08314680	,14472086	1,000	-,3229168	,4892104
	5	1,78705317*	,13830387	,000	1,3989913	2,1751150
5	1	-1,4729325*	,13731537	,000	-1,8582194	-1,0876456
	2	-1,8026369*	,13955909	,000	-2,1942229	-1,4110508
	3	-1,7039064*	,14021830	,000	-2,0973434	-1,3104694
	4	-1,7870532*	,13830387	,000	-2,1751150	-1,3989913

*. The mean difference is significant at the .05 level.

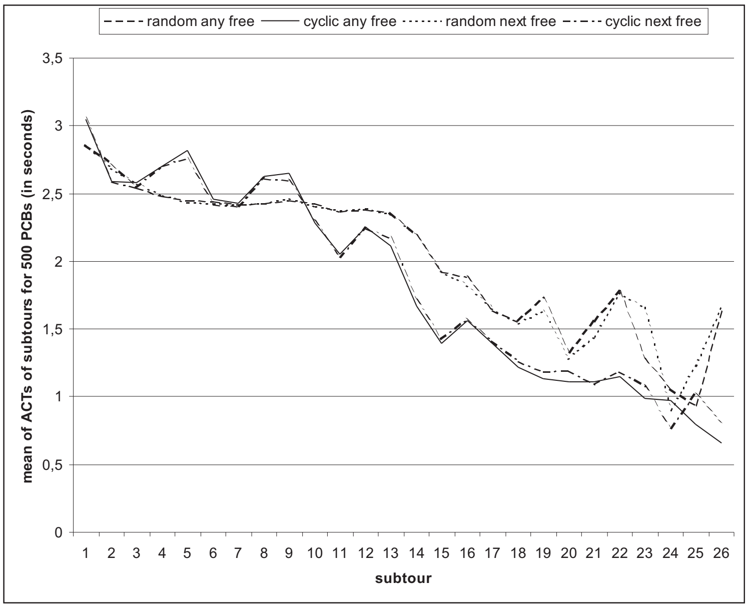

Figure 3.5: The development of the mean ACTs of subjobs with the four heuristics for OPSE [46].

of the free and occupied nozzles up-to-date.

3. **Concatenate the placement sequence.** Build a new partition from the components that have been selected in phases 1 and 2, and append it to the end of the placement sequence.

The four heuristics developed in [46] differ from each other in the ways how phases 1 and 2 are done. The third phase is identical in all the algorithms. In the experimental tests the two heuristics that perform the phase 1 in the same way and selected the first component of each subjob in so-called *cyclic turns* outperformed the methods that use a random starting component (ANOVA, Sig. 0.000) [46].

In addition, the values of the ACTs of the late subjobs tend to decrease with the methods based on cyclic selection in phase 1. However, this is not the case with the methods using randomly starting components (see Figure 3.5).

Chapter 4

Summary of publications

4.1 Minimizing the number of pickups on a multi-head placement machine

In the first publication [33], the *minimal length pickup sequence problem* (MLP) for a collect-and-place machine is defined and studied. The placement head of the machine operates in cycles of four phases. First, it collects components from a feeder unit. Then, the head travels on the PCB. Third, the head places the components collected in the first phase. Finally, the placement head travels back on the feeder unit to collect more components. The length of the pickup sequence is defined as the number of phases in which the head collects components. A greedy method for solving the minimal length pickup problem, for a fixed job and fixed nozzle assortment in the placement head, is proposed and proved to be optimal.

The MLP is then used to define *optimal nozzle selection problem* (ONS) in which the contents of the placement head (or arm) should be assigned for a fixed job so that MLP is minimized. The optimal solutions of the greedy method for MLP is utilized when solving ONS using brute force for small problem instances.

4.2 The selection of nozzles for minimizing the number of pick-ups on a multi-head placement machine

In the second publication [48], the ONS problem for the collect-and-place machine defined in [33] is discussed. Since the brute force method is feasible only for small problem instances of ONS a heuristic approach is used. Three new heuristic algorithms are proposed and compared to each other in experimental tests. Two of these are based on frequencies of the nozzle requirements of the placement job. In the third heuristic lacks of suitable nozzles are analyzed and the nozzle selection is derived from that fact. A local search method that improves the quality of the results of the heuristics is developed, too. The results of the heuristics added with the improvement technique are compared to optimal results for small problem instances. In order to achieve harder test cases, the data used in experimental tests is random data which has been generated with Markov models. In tests the frequency-based heuristics outperformed the third heuristic (best addon -heuristic). The frequency-based heuristics augmented with local search yield solutions close to optimal.

4.3 Minimizing the arm movements of a multi-head gantry machine

In the third publication [32], the length of the path that the placement head travels during the pickup phase in the collect-and-place machine is studied and minimized. The *single pickup event problem* (SPE) and the *minimal pickup sequence problem* (MPS) are defined. In the SPE, the length of the path that the placement head travels in a single pickup phase is calculated for a fixed subjob of placements and a fixed nozzle assortment in the placement arm. Then, it is shown that SPE can be solved exactly. In MPS the whole placement job, the arm contents and the feeder assignment are fixed. The problem is to find such partition of the job into subjobs that the overall length of the travel path of the placement head is minimized. A number of different ways to define the travel path are proposed for the cases where the accurate operation of the placement head is not known. A simple brute force method for MPS is proposed. It is pointed out that the greedy method of [33] is not always optimal if the object function is the length of travel path instead of the number of pickup phases. Finally, the MPS is solved using the dynamic programming approach.

4.4 Solving the feeder assignment on a revolver-head gantry machine

The fourth publication [45] discusses the *optimal feeder assignment problem* (OFA) for a collect-and-place machine, in which the nozzle selection in the placement arm, the placement order, partition into subjobs and positions are given. The goal is to find such order of component reels in the slots of the feeder unit that minimizes the *assembly cycle time* (ACT) of the job. The ACT is defined mathematically. In order to solve the OFA, three new heuristics, one existing and a theoretic lower bound are proposed. These are then compared to each other in experimental tests that are based on realistic PCB assembly data. However, no statistically significant differences were found between the heuristics.

4.5 Minimizing the assembly cycle time on a revolver gantry machine

In the fifth publication [46], the OFA and the *optimal placement sequence problem* (OPSE), for a collect-and-place machine, are studied. A new heuristic is proposed for OFA. The new method is based on the analysis of the neighborhoods of the component placements on a PCB. In experimental tests with data based on realistic PCBs this approach turned out to outperform all the heuristics discussed in [45].

In the OPSE, the PCB recipe, the feeder assignment and the nozzle selection in the placement arm are fixed and the task is to find such a pickup and placement order that minimizes the ACT. Four new heuristics are proposed. There are two phases in these heuristics that are altered when composing the new heuristics. The selection of a starting component can be either cyclic or random for each subjob generation. After that the rest of the components into the arm can be collected by emphasizing either nozzle order and distance on a PCB area or distance only. In experimental tests the selection of the starting component was found significant. In experimental tests, the heuristic algorithms utilizing the cyclic starting component selection were found better than the methods of the random starting components with data based on realistic PCBs.

Chapter 5

Conclusion

In this thesis, the optimization of a collect-and-place machine used in printed circuit board assembly is studied. Especially, the focus is on the case in which a single PCB is produced with a single machine. However, the entire control problem concerning the machine is too complex to be solved in a single step. By using a hierarchical decomposition the entire control problem of the machine is divided into multiple subproblems which depend strongly from each others. All the subproblems of the 1 PCB - 1 machine -context are described and then classified and reviewed.

Some of the described subproblems of Section 3.2 are studied and solved. The solutions apply exact methods when possible. The rest of the studied problems are solved by developing heuristic algorithms and being convinced with experimental tests that they are applicable and feasible. A number of different quality functions were proposed and used in these tests.

For the problems that are not studied in this thesis or are solved using heuristics in the present study, the development should continue. Especially, for the OPSE problem more brilliant algorithms should be composed. At least a kind of local search would certainly improve the solutions found by constructive heuristics.

Another fundamental topic of criticism concerns the accuracy of the machine model used in the present studies. In particular, a certain set of parameters and properties are taken as an assumption in the experimental tests. This leads into a specialized case of the simulated machine type and it is unsure whether the observations and conclusions can be generalized to all machines with similar design. To be convinced of this, one should have a very good knowledge and accurate understanding of the machines offered by numerous different manufacturers. Then it could be possible to derive the major commonalities and develop definitely universal methods for the problems.

Bibliography

[1] J. Ahmadi, S. Grotzinger, and D. Johnson. Component allocation and partitioning for dual delivery placement machine. *Operations Research, 36, 2, March-April 1988, pp. 176-191*, 1988.

[2] R.H. Ahmadi and P. Kouvelis. Staging problem of a dual delivery pick-and-place machine in printed circuit card assembly. *Operations Research, 42, 1, January-February 1994, pp. 81-91*, 1994.

[3] K. Altinkemer, B. Kazaz, M. Köksalan, and H. Moskowitz. Optimization of printed circuit board manufacturing: Integrated modeling and algorithms. *European Journal of Operational Research 124, pp. 409-421*, 2000.

[4] M. Ayob, P. Cowling, and G. Kendall. Optimisation for surface mount placement machines. In *Proceedings of the IEEE ICIT'02, Bangkok 11-14 December 2002, pp. 498-503*, 2002.

[5] M. Ayob and G. Kendall. A survey of surface mount device placement machine optimisation: Machine classification. *European Journal of the Operational Research 186, pp. 893-914*, 2008.

[6] S. Balin. Non-identical parallel machine scheduling using genetic algorithm. *Expert Systems with Applications 38, pp. 6814-6821*, 2011.

[7] M. Ball and M. Magazine. Sequencing of insertions in printed circuit board assembly. *Operations Research 36(2), pp. 192-201*, 1988.

[8] J.F. Bard, R.W. Clayton, and T.A. Feo. Machine setup and component placement in printed circuit board assembly. *International Journal of Flexible Manufacturing Systems 6, pp. 5-31*, 1994.

[9] R.E. Burkard. Quadratic assignment problems. *European Journal of Operational Research 15, pp. 283-289*, 1984.

[10] E.K. Burke, P.I. Cowling, and R. Keuthen. Effective local and guided variable neighbourhood search methods for the asymmetric travelling salesman problem. In *E.J.W. Boers et al. (Eds.) EvoWorkshop 2001, LNCS 2037, pp. 203-212*, 2001.

[11] Y.M. Chen and C.-T. Lin. A particle swarm optimization approach to optimize component placement in printed circuit board assembly. *International Journal of Advanced Manufacturing Technology 35, pp. 610-620*, 2007.

[12] Y.M. Chen and C.-T. Lin. Optimizing the operation sequence of a multihead surface mounting machine using a discrete particle swarm optimization algorithm. *Journal of Artificial Evolution and Applications, Volume 2008, pp. 1-8*, 2008.

[13] Y. Crama, O.E. Flippo, J. van de Klundert, and F.C.R. Spieksma. The component retrieval problem in printed circuit board assembly. *International Journal of Flexible Manufacturing Systems 8, pp. 287-312*, 1996.

[14] Y. Crama, A.W.J. Koleen, A.G.Oerlemans, and F.C.R. Spieksma. Throughput rate optimization in the automated assembly of printed circuit boards. *Annals of Operations Research 26, pp. 455-80*, 1990.

[15] Y. Crama and J. van de Klundert. Worst-case performance of approximation algorithms for tool management problems. *Naval Research Logistics 46, 5(1999), pp. 445-62*, 1999.

[16] Y. Crama, J. van de Klundert, and F.C.R. Spieksma. Production planning problems in printed circuit board assembly. *Discrete Applied Mathematics 123, pp. 339-361*, 2002.

[17] P. Csaszar, T.M. Tirpak, and P.C. Nelson. Optimization of a high-speed placement machine using tabu search algorithms. *Annals of Operations Research 96, 1-4, pp. 125-147*, 2000.

[18] A. Dikos, P.C. Nelson, T.M. Tirpak, and W. Wang. Optimization of high-mix printed circuit card assembly using genetic algorithms. *Annals of Operations Research, Volume 75, pp. 303-324*, 1997.

[19] K.P. Ellis, J.E. Kobza, and F.J. Vittes. Development of a placement time estimator function for a turret style surface mount placement machine. *Robotics and Computer Integrated Manufacturing 18, pp. 241-254*, 2002.

[20] K.P. Ellis, F.J. Vittes, and J.E. Kobza. Optimizing the performance of a surface mount placement machine. *IEEE Transactions on Electronics Packaging Manufacturing 24, 3, July, pp. 160-170*, 2001.

[21] M. Grunow, H.-O. Günther, and M. Schleusener. Component allocation for printed circuit board assembly using modular placement machines. *International Journal of Production Research 41, 6, pp.1311-1331*, 2003.

[22] M. Grunow, H.-O. Günther, M. Schleusener, and I.O. Yilmaz. Operations planning for collect-and-place machines in PCB assembly. *Computers & Industrial Engineering 47, pp.409-429*, 2004.

[23] C.S. Hardas, T.L. Doolen, and D.H. Jensen. Development of a genetic algorithm for component placement sequence optimization in printed circuit board assembly. *Computers & Industrial Engineering 55, pp. 165-182*, 2008.

[24] W. Ho and P. Ji. Component scheduling for chip shooter machines: a hybrid genetic algorithm approach. *Computers & Operations Research 30, pp. 2175-2189*, 2003.

[25] W. Ho and P. Ji. A hybrid genetic algorithm for component sequencing and feeder arrangement. *Journal of Intelligent Manufacturing 15, pp. 307-315*, 2004.

[26] W. Ho, P. Ji, and Y. Wu. A heuristic approach for component scheduling on a high-speed PCB assembly machine. *Production Planning & Control 18, 8, pp. 655-665*, 2007.

[27] M. Johnsson. Operational and tactical level optimization in printed circuit board assembly. *TUCS Dissertations 16, May 1999*, 1999.

[28] M. Johnsson and J. Smed. Observations on pcb assembly optimization. *Electronic Packaging & Production 41(5), pp.38-42*, 2001.

[29] K. Kallio, M. Johnsson, and O.S. Nevalainen. Estimating the operation time of flexible surface mount placement machines. *Production Engineering, Research and Development, 1-10*, 2012.

[30] R. Keuthen. Heuristic approaches for routing optimisation. *School of Computer Science and Information Technology, University of Nottingham*, 2003.

[31] C. Klomp, J. van de Klundert, F.C.R. Spieksma, and S. Voogt. The feeder rack assignment problem in PCB assembly: A case study. *Int.J.Production Economics 64, pp. 399-407*, 2000.

[32] T. Knuutila, S. Pyöttiälä, and O.S. Nevalainen. Minimizing the arm movements of a multi-head gantry machine. In *Proceedings of 4th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2007), pp. 326-335*, 2007.

[33] T. Knuutila, S. Pyöttiälä, and O.S. Nevalainen. Minimizing the number of pickups on a multi-head placement machine. *Journal of the Operational Research Society 58(1), pp. 115-121, Jan*, 2007.

[34] T. Knuutila, T. Suomi, S. Emet, M. Johnsson, and O.S. Nevalainen. Organizing the nozzle magazine of a gantry-type PCB assembly machine. *International Journal of Advanced Manufacturing Technology 68(5-8), pp. 1189-1202*, 2013.

[35] O. Kulak, I.O. Yilmaz, and H.-O. Günther. PCB assembly scheduling for collect-and-place machines using genetic algorithms. *International Journal of Production Research 45, 17, pp. 3949-3969*, 2007.

[36] S.H. Lee, H.B. Lee, and T.H. Park. A hierarchical method to improve the productivity of a multi-head surface mounting machine. In *Proc. of the IEEE International conference on Robotics and Automation*, 1999.

[37] T. Leipälä and O.S. Nevalainen. Optimization of the movements of a component placement machine. *European Journal of Operational Research 38, pp. 167-177*, 1989.

[38] S. Li, C. Hu, and F. Tian. Enhancing optimal feeder assignment of the multi-head surface mounting machine using genetic algorithms. *Applied Soft Computing 8, pp. 522-529*, 2008.

[39] T.S. Loh, S.T.S. Bukkapatnam, D. Medeiros, and H. Kwon. A genetic algorithm for sequential part assignment for PCB assembly. *Computers & Industrial Engineering 40, pp. 293-307*, 2001.

[40] G. Magyar. On solution approaches for some industrially motivated combinatorial optimization problems. *TUCS Dissertations 23, March 2000*, 2000.

[41] G. Magyar, M. Johnsson, and O.S. Nevalainen. On solving single machine optimization problems in electronics assembly. *Journal of Electronics Manufacturing 9, 4, pp. 249-267*, 1999.

[42] T.-H. Park and N. Kim. A dynamic programming approach to PCB assembly optimization for surface mounters. *International Journal of Control, Automation, and Systems 5, 2, April 2007, pp. 192-199*, 2007.

[43] D.T. Pham, S. Otri, and A. Haj Darwish. Application of the bees algorithm to PCB assembly optimisation. In *Proceedings of Innovative Production Machines and Systems (IPROMS2007)*, 2007.

[44] S. Pyöttiälä, T. Knuutila, M. Johnsson, and O.S. Nevalainen. Improving the pick-ups of components on a gantry-type placement machine. *TUCS Technical report, No. 692*, 2005.

[45] S. Pyöttiälä, T. Knuutila, M. Johnsson, and O.S. Nevalainen. Solving the feeder assignment on a revolver-head gantry machine. In *Proceedings of 6th international conference on informatics in control, automation and robotics, (ICINCO 2009), volume 1 - Intelligent Control Systems and Optimization, pp. 75-80*, 2009.

[46] S. Pyöttiälä, T. Knuutila, M. Johnsson, and O.S. Nevalainen. Minimizing the assembly cycle time on a revolver gantry machine. *Computers & Operations Research 40, pp. 2611-2624*, 2013.

[47] S. Pyöttiälä, T. Knuutila, and O.S. Nevalainen. Improving the pick-ups of components on a gantry-type placement machine. In *5th International Conference on Technology and Automation, Thessaloniki, Greece*, 2005.

[48] S. Pyöttiälä, T. Knuutila, and O.S. Nevalainen. The selection of nozzles for minimizing the number of pick-ups on a multi-head placement machine.

In *Proceedings of the Third International Conference on Group Technology / Cellular Manufacturing 2006 Conference (GTCM2006), pp. 382-391, Groningen, The Netherlands*, 2006.

[49] Cs. Raduly-Baka, T. Knuutila, M. Johnsson, and O.S. Nevalainen. Selecting the nozzle assortment for a gantry-type placement machine. *OR Spectrum, 30, pp. 493-513*, 2008.

[50] Cs. Raduly-Baka, T. Knuutila, M. Johnsson, and O.S. Nevalainen. Construction of component tapes for radial placement machines. *Computers & Operation Research 3, pp. 1488-1499*, 2009.

[51] Cs. Raduly-Baka, T. Knuutila, M. Johnsson, and O.S. Nevalainen. Organizing the operation of radial machine sequencers for multiple PCB types. *Computers & Industrial Engineering 61, 4, pp. 1003-1010*, 2011.

[52] K. Salonen. Setup optimization in high-mix surface mount PCB assembly. *TUCS Dissertations 109, October 2008*, 2008.

[53] K. Salonen, J. Smed, M. Johnsson, and O. Nevalainen. Grouping and sequencing PCB assembly jobs with minimum feeder setups. *Robotics and Computer-Integrated Manufacturing 22, 4, pp. 297-305*, 2006.

[54] A. Scholl and C. Becker. State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research 168, pp. 666-693*, 2006.

[55] J. Smed. Production planning in printed circuit board assembly. *TUCS Dissertations 36, January 2002*, 2002.

[56] J. Smed, M. Johnsson, T. Johtela, and O. Nevalainen. Techniques and applications of production planning in electronics manufacturing systems. *TUCS Technical report, No. 320*, 1999.

[57] J. Smed, M. Johnsson, M. Puranen, T. Leipälä, and O. Nevalainen. Job grouping in surface mounted component printing. *Robotics and Computer Integrated Manufacturing 15, 1, pp. 39-49*, 1999.

[58] J. Smed, K. Salonen, M. Johnsson, T. Johtela, and O. Nevalainen. Grouping PCBs with minimum feeder changes. *The International Journal of Flexible Manufacturing Systems 15, pp. 19-35*, 2003.

[59] D.-S. Sun, T.-E. Lee, and K.-H. Kim. Component allocation and feeder arrangement for a dual-gantry multi-head surface mounting placement tool. *International Journal of Production Economics 95, pp. 245-264*, 2005.

[60] T.M. Tirpak, P.K. Mohapatra, P.C. Nelson, and R.R. Rajbhandari. A generic classification and object-oriented simulation toolkit for SMT assembly equipment. *IEEE Transactions on Systems, Man, and Cybernetics, Part A, pp. 104-122*, 2002.

[61] T.M. Tirpak, P.C. Nelson, and A.J. Aswani. Optimization of revolver head SMT machines using adaptive simulating annealing (ASA). In *Proceedings of Int'l Electronics Manufacturing Technology Symposium IEEE/CPMT, pp. 214-220*, 2000.

[62] P.J.M. van Laarhoven and W.H.M. Zijm. Production preparation and numerical control in PCB assembly. *International Journal of Flexible Manufacturing Systems 5, pp. 187-207*, 1993.

[63] Y. Zeng, D. Ma, Y. Jin, and J. Yan. Hierarchical planning for a surface mounting machine placement. *Journal of Zhejiang University 5(11), pp. 1449-1455*, 2004.

Chapter 6

Publications

Publication 1

T. Knuutila, S. Pyöttiälä, and O.S. Nevalainen. Minimizing the number of pickups on a multi-head placement machine. *Journal of the Operational Research Society*, 58(1), pages 115-121, Jan, 2007.

Publication 2

S. Pyöttiälä, T. Knuutila, and O.S. Nevalainen. The selection of nozzles for minimizing the number of pick-ups on a multi-head placement machine. In *Proceedings of the Third International Conference on Group Technology / Cellular Manufacturing 2006*, pages 382-391, Groningen, The Netherlands, July 2006.

Publication 3

T. Knuutila, S. Pyöttiälä, and O.S. Nevalainen. Minimizing the arm movements of a multi-head gantry machine. In *Proceedings of 4th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2007)*, pages 326-335, Angers, France, May 2007.

Publication 4

S. Pyöttiälä, T. Knuutila, M. Johnsson, and O.S. Nevalainen. Solving the feeder assignment on a revolver-head gantry machine. In *Proceedings of 6th International Conference on Informatics in Control, Automation and Robotics, (ICINCO 2009)*, volume 1 - Intelligent Control Systems and Optimization, pages 75-80, Milan, Italy, July 2009.

Publication 5

S. Pyöttiälä, T. Knuutila, M. Johnsson, and O.S. Nevalainen. Minimizing the assembly cycle time on a revolver gantry machine. *Computers & Operations Research* 40, pages 2611-2624, 2013.

Turku Centre for Computer Science
TUCS Dissertations

1. **Marjo Lipponen**, On Primitive Solutions of the Post Correspondence Problem
2. **Timo Käkölä**, Dual Information Systems in Hyperknowledge Organizations
3. **Ville Leppänen**, Studies on the Realization of PRAM
4. **Cunsheng Ding**, Cryptographic Counter Generators
5. **Sami Viitanen**, Some New Global Optimization Algorithms
6. **Tapio Salakoski**, Representative Classification of Protein Structures
7. **Thomas Långbacka**, An Interactive Environment Supporting the Development of Formally Correct Programs
8. **Thomas Finne**, A Decision Support System for Improving Information Security
9. **Valeria Mihalache**, Cooperation, Communication, Control. Investigations on Grammar Systems.
10. **Marina Waldén**, Formal Reasoning About Distributed Algorithms
11. **Tero Laihonen**, Estimates on the Covering Radius When the Dual Distance is Known
12. **Lucian Ilie**, Decision Problems on Orders of Words
13. **Jukkapekka Hekanaho**, An Evolutionary Approach to Concept Learning
14. **Jouni Järvinen**, Knowledge Representation and Rough Sets
15. **Tomi Pasanen**, In-Place Algorithms for Sorting Problems
16. **Mika Johnsson**, Operational and Tactical Level Optimization in Printed Circuit Board Assembly
17. **Mats Aspnäs**, Multiprocessor Architecture and Programming: The Hathi-2 System
18. **Anna Mikhajlova**, Ensuring Correctness of Object and Component Systems
19. **Vesa Torvinen**, Construction and Evaluation of the Labour Game Method
20. **Jorma Boberg**, Cluster Analysis. A Mathematical Approach with Applications to Protein Structures
21. **Leonid Mikhajlov**, Software Reuse Mechanisms and Techniques: Safety Versus Flexibility
22. **Timo Kaukoranta**, Iterative and Hierarchical Methods for Codebook Generation in Vector Quantization
23. **Gábor Magyar**, On Solution Approaches for Some Industrially Motivated Combinatorial Optimization Problems
24. **Linas Laibinis**, Mechanised Formal Reasoning About Modular Programs
25. **Shuhua Liu**, Improving Executive Support in Strategic Scanning with Software Agent Systems
26. **Jaakko Järvi**, New Techniques in Generic Programming – C++ is more Intentional than Intended
27. **Jan-Christian Lehtinen**, Reproducing Kernel Splines in the Analysis of Medical Data
28. **Martin Büchi**, Safe Language Mechanisms for Modularization and Concurrency
29. **Elena Troubitsyna**, Stepwise Development of Dependable Systems
30. **Janne Näppi**, Computer-Assisted Diagnosis of Breast Calcifications
31. **Jianming Liang**, Dynamic Chest Images Analysis
32. **Tiberiu Seceleanu**, Systematic Design of Synchronous Digital Circuits
33. **Tero Aittokallio**, Characterization and Modelling of the Cardiorespiratory System in Sleep-Disordered Breathing
34. **Ivan Porres**, Modeling and Analyzing Software Behavior in UML
35. **Mauno Rönkkö**, Stepwise Development of Hybrid Systems
36. **Jouni Smed**, Production Planning in Printed Circuit Board Assembly
37. **Vesa Halava**, The Post Correspondence Problem for Market Morphisms
38. **Ion Petre**, Commutation Problems on Sets of Words and Formal Power Series
39. **Vladimir Kvassov**, Information Technology and the Productivity of Managerial Work
40. **Frank Tétard**, Managers, Fragmentation of Working Time, and Information Systems

41. **Jan Manuch**, Defect Theorems and Infinite Words
42. **Kalle Ranto**, Z_4-Goethals Codes, Decoding and Designs
43. **Arto Lepistö**, On Relations Between Local and Global Periodicity
44. **Mika Hirvensalo**, Studies on Boolean Functions Related to Quantum Computing
45. **Pentti Virtanen**, Measuring and Improving Component-Based Software Development
46. **Adekunle Okunoye**, Knowledge Management and Global Diversity – A Framework to Support Organisations in Developing Countries
47. **Antonina Kloptchenko**, Text Mining Based on the Prototype Matching Method
48. **Juha Kivijärvi**, Optimization Methods for Clustering
49. **Rimvydas Rukšėnas**, Formal Development of Concurrent Components
50. **Dirk Nowotka**, Periodicity and Unbordered Factors of Words
51. **Attila Gyenesei**, Discovering Frequent Fuzzy Patterns in Relations of Quantitative Attributes
52. **Petteri Kaitovaara**, Packaging of IT Services – Conceptual and Empirical Studies
53. **Petri Rosendahl**, Niho Type Cross-Correlation Functions and Related Equations
54. **Péter Majlender**, A Normative Approach to Possibility Theory and Soft Decision Support
55. **Seppo Virtanen**, A Framework for Rapid Design and Evaluation of Protocol Processors
56. **Tomas Eklund**, The Self-Organizing Map in Financial Benchmarking
57. **Mikael Collan**, Giga-Investments: Modelling the Valuation of Very Large Industrial Real Investments
58. **Dag Björklund**, A Kernel Language for Unified Code Synthesis
59. **Shengnan Han**, Understanding User Adoption of Mobile Technology: Focusing on Physicians in Finland
60. **Irina Georgescu**, Rational Choice and Revealed Preference: A Fuzzy Approach
61. **Ping Yan**, Limit Cycles for Generalized Liénard-Type and Lotka-Volterra Systems
62. **Joonas Lehtinen**, Coding of Wavelet-Transformed Images
63. **Tommi Meskanen,** On the NTRU Cryptosystem
64. **Saeed Salehi**, Varieties of Tree Languages
65. **Jukka Arvo**, Efficient Algorithms for Hardware-Accelerated Shadow Computation
66. **Mika Hirvikorpi**, On the Tactical Level Production Planning in Flexible Manufacturing Systems
67. **Adrian Costea**, Computational Intelligence Methods for Quantitative Data Mining
68. **Cristina Seceleanu**, A Methodology for Constructing Correct Reactive Systems
69. **Luigia Petre**, Modeling with Action Systems
70. **Lu Yan**, Systematic Design of Ubiquitous Systems
71. **Mehran Gomari**, On the Generalization Ability of Bayesian Neural Networks
72. **Ville Harkke**, Knowledge Freedom for Medical Professionals – An Evaluation Study of a Mobile Information System for Physicians in Finland
73. **Marius Cosmin Codrea**, Pattern Analysis of Chlorophyll Fluorescence Signals
74. **Aiying Rong**, Cogeneration Planning Under the Deregulated Power Market and Emissions Trading Scheme
75. **Chihab BenMoussa**, Supporting the Sales Force through Mobile Information and Communication Technologies: Focusing on the Pharmaceutical Sales Force
76. **Jussi Salmi**, Improving Data Analysis in Proteomics
77. **Orieta Celiku**, Mechanized Reasoning for Dually-Nondeterministic and Probabilistic Programs
78. **Kaj-Mikael Björk**, Supply Chain Efficiency with Some Forest Industry Improvements
79. **Viorel Preoteasa**, Program Variables – The Core of Mechanical Reasoning about Imperative Programs
80. **Jonne Poikonen**, Absolute Value Extraction and Order Statistic Filtering for a Mixed-Mode Array Image Processor
81. **Luka Milovanov**, Agile Software Development in an Academic Environment
82. **Francisco Augusto Alcaraz Garcia**, Real Options, Default Risk and Soft Applications
83. **Kai K. Kimppa**, Problems with the Justification of Intellectual Property Rights in Relation to Software and Other Digitally Distributable Media
84. **Dragoş Truşcan**, Model Driven Development of Programmable Architectures
85. **Eugen Czeizler**, The Inverse Neighborhood Problem and Applications of Welch Sets in Automata Theory

86. **Sanna Ranto**, Identifying and Locating-Dominating Codes in Binary Hamming Spaces
87. **Tuomas Hakkarainen**, On the Computation of the Class Numbers of Real Abelian Fields
88. **Elena Czeizler**, Intricacies of Word Equations
89. **Marcus Alanen**, A Metamodeling Framework for Software Engineering
90. **Filip Ginter**, Towards Information Extraction in the Biomedical Domain: Methods and Resources
91. **Jarkko Paavola**, Signature Ensembles and Receiver Structures for Oversaturated Synchronous DS-CDMA Systems
92. **Arho Virkki**, The Human Respiratory System: Modelling, Analysis and Control
93. **Olli Luoma**, Efficient Methods for Storing and Querying XML Data with Relational Databases
94. **Dubravka Ilić**, Formal Reasoning about Dependability in Model-Driven Development
95. **Kim Solin**, Abstract Algebra of Program Refinement
96. **Tomi Westerlund**, Time Aware Modelling and Analysis of Systems-on-Chip
97. **Kalle Saari**, On the Frequency and Periodicity of Infinite Words
98. **Tomi Kärki**, Similarity Relations on Words: Relational Codes and Periods
99. **Markus M. Mäkelä**, Essays on Software Product Development: A Strategic Management Viewpoint
100. **Roope Vehkalahti**, Class Field Theoretic Methods in the Design of Lattice Signal Constellations
101. **Anne-Maria Ernvall-Hytönen**, On Short Exponential Sums Involving Fourier Coefficients of Holomorphic Cusp Forms
102. **Chang Li**, Parallelism and Complexity in Gene Assembly
103. **Tapio Pahikkala**, New Kernel Functions and Learning Methods for Text and Data Mining
104. **Denis Shestakov**, Search Interfaces on the Web: Querying and Characterizing
105. **Sampo Pyysalo**, A Dependency Parsing Approach to Biomedical Text Mining
106. **Anna Sell**, Mobile Digital Calendars in Knowledge Work
107. **Dorina Marghescu**, Evaluating Multidimensional Visualization Techniques in Data Mining Tasks
108. **Tero Säntti**, A Co-Processor Approach for Efficient Java Execution in Embedded Systems
109. **Kari Salonen**, Setup Optimization in High-Mix Surface Mount PCB Assembly
110. **Pontus Boström**, Formal Design and Verification of Systems Using Domain-Specific Languages
111. **Camilla J. Hollanti**, Order-Theoretic Mehtods for Space-Time Coding: Symmetric and Asymmetric Designs
112. **Heidi Himmanen**, On Transmission System Design for Wireless Broadcasting
113. **Sébastien Lafond**, Simulation of Embedded Systems for Energy Consumption Estimation
114. **Evgeni Tsivtsivadze**, Learning Preferences with Kernel-Based Methods
115. **Petri Salmela**, On Commutation and Conjugacy of Rational Languages and the Fixed Point Method
116. **Siamak Taati**, Conservation Laws in Cellular Automata
117. **Vladimir Rogojin**, Gene Assembly in Stichotrichous Ciliates: Elementary Operations, Parallelism and Computation
118. **Alexey Dudkov**, Chip and Signature Interleaving in DS CDMA Systems
119. **Janne Savela**, Role of Selected Spectral Attributes in the Perception of Synthetic Vowels
120. **Kristian Nybom**, Low-Density Parity-Check Codes for Wireless Datacast Networks
121. **Johanna Tuominen**, Formal Power Analysis of Systems-on-Chip
122. **Teijo Lehtonen**, On Fault Tolerance Methods for Networks-on-Chip
123. **Eeva Suvitie**, On Inner Products Involving Holomorphic Cusp Forms and Maass Forms
124. **Linda Mannila**, Teaching Mathematics and Programming – New Approaches with Empirical Evaluation
125. **Hanna Suominen**, Machine Learning and Clinical Text: Supporting Health Information Flow
126. **Tuomo Saarni**, Segmental Durations of Speech
127. **Johannes Eriksson**, Tool-Supported Invariant-Based Programming

128. **Tero Jokela**, Design and Analysis of Forward Error Control Coding and Signaling for Guaranteeing QoS in Wireless Broadcast Systems
129. **Ville Lukkarila**, On Undecidable Dynamical Properties of Reversible One-Dimensional Cellular Automata
130. **Qaisar Ahmad Malik**, Combining Model-Based Testing and Stepwise Formal Development
131. **Mikko-Jussi Laakso**, Promoting Programming Learning: Engagement, Automatic Assessment with Immediate Feedback in Visualizations
132. **Riikka Vuokko**, A Practice Perspective on Organizational Implementation of Information Technology
133. **Jeanette Heidenberg**, Towards Increased Productivity and Quality in Software Development Using Agile, Lean and Collaborative Approaches
134. **Yong Liu**, Solving the Puzzle of Mobile Learning Adoption
135. **Stina Ojala**, Towards an Integrative Information Society: Studies on Individuality in Speech and Sign
136. **Matteo Brunelli**, Some Advances in Mathematical Models for Preference Relations
137. **Ville Junnila**, On Identifying and Locating-Dominating Codes
138. **Andrzej Mizera**, Methods for Construction and Analysis of Computational Models in Systems Biology. Applications to the Modelling of the Heat Shock Response and the Self-Assembly of Intermediate Filaments.
139. **Csaba Ráduly-Baka**, Algorithmic Solutions for Combinatorial Problems in Resource Management of Manufacturing Environments
140. **Jari Kyngäs**, Solving Challenging Real-World Scheduling Problems
141. **Arho Suominen**, Notes on Emerging Technologies
142. **József Mezei**, A Quantitative View on Fuzzy Numbers
143. **Marta Olszewska**, On the Impact of Rigorous Approaches on the Quality of Development
144. **Antti Airola**, Kernel-Based Ranking: Methods for Learning and Performace Estimation
145. **Aleksi Saarela**, Word Equations and Related Topics: Independence, Decidability and Characterizations
146. **Lasse Bergroth**, Kahden merkkijonon pisimmän yhteisen alijonon ongelma ja sen ratkaiseminen
147. **Thomas Canhao Xu**, Hardware/Software Co-Design for Multicore Architectures
148. **Tuomas Mäkilä**, Software Development Process Modeling – Developers Perspective to Contemporary Modeling Techniques
149. **Shahrokh Nikou**, Opening the Black-Box of IT Artifacts: Looking into Mobile Service Characteristics and Individual Perception
150. **Alessandro Buoni**, Fraud Detection in the Banking Sector: A Multi-Agent Approach
151. **Mats Neovius**, Trustworthy Context Dependency in Ubiquitous Systems
152. **Fredrik Degerlund**, Scheduling of Guarded Command Based Models
153. **Amir-Mohammad Rahmani-Sane**, Exploration and Design of Power-Efficient Networked Many-Core Systems
154. **Ville Rantala**, On Dynamic Monitoring Methods for Networks-on-Chip
155. **Mikko Pelto**, On Identifying and Locating-Dominating Codes in the Infinite King Grid
156. **Anton Tarasyuk**, Formal Development and Quantitative Verification of Dependable Systems
157. **Muhammad Mohsin Saleemi**, Towards Combining Interactive Mobile TV and Smart Spaces: Architectures, Tools and Application Development
158. **Tommi J. M. Lehtinen**, Numbers and Languages
159. **Peter Sarlin**, Mapping Financial Stability
160. **Alexander Wei Yin**, On Energy Efficient Computing Platforms
161. **Mikołaj Olszewski**, Scaling Up Stepwise Feature Introduction to Construction of Large Software Systems
162. **Maryam Kamali**, Reusable Formal Architectures for Networked Systems
163. **Zhiyuan Yao**, Visual Customer Segmentation and Behavior Analysis – A SOM-Based Approach
164. **Timo Jolivet**, Combinatorics of Pisot Substitutions
165. **Rajeev Kumar Kanth**, Analysis and Life Cycle Assessment of Printed Antennas for Sustainable Wireless Systems
166. **Khalid Latif**, Design Space Exploration for MPSoC Architectures

167. **Bo Yang**, Towards Optimal Application Mapping for Energy-Efficient Many-Core Platforms
168. **Ali Hanzala Khan**, Consistency of UML Based Designs Using Ontology Reasoners
169. **Sonja Leskinen**, m-Equine: IS Support for the Horse Industry
170. **Fareed Ahmed Jokhio**, Video Transcoding in a Distributed Cloud Computing Environment
171. **Moazzam Fareed Niazi**, A Model-Based Development and Verification Framework for Distributed System-on-Chip Architecture
172. **Mari Huova**, Combinatorics on Words: New Aspects on Avoidability, Defect Effect, Equations and Palindromes
173. **Ville Timonen**, Scalable Algorithms for Height Field Illumination
174. **Henri Korvela**, Virtual Communities – A Virtual Treasure Trove for End-User Developers
175. **Kameswar Rao Vaddina**, Thermal-Aware Networked Many-Core Systems
176. **Janne Lahtiranta**, New and Emerging Challenges of the ICT-Mediated Health and Well-Being Services
177. **Irum Rauf**, Design and Validation of Stateful Composite RESTful Web Services
178. **Jari Björne**, Biomedical Event Extraction with Machine Learning
179. **Katri Haverinen**, Natural Language Processing Resources for Finnish: Corpus Development in the General and Clinical Domains
180. **Ville Salo**, Subshifts with Simple Cellular Automata
181. **Johan Ersfolk**, Scheduling Dynamic Dataflow Graphs
182. **Hongyan Liu**, On Advancing Business Intelligence in the Electricity Retail Market
183. **Adnan Ashraf**, Cost-Efficient Virtual Machine Management: Provisioning, Admission Control, and Consolidation
184. **Muhammad Nazrul Islam**, Design and Evaluation of Web Interface Signs to Improve Web Usability: A Semiotic Framework
185. **Johannes Tuikkala**, Algorithmic Techniques in Gene Expression Processing: From Imputation to Visualization
186. **Natalia Díaz Rodríguez**, Semantic and Fuzzy Modelling for Human Behaviour Recognition in Smart Spaces. A Case Study on Ambient Assisted Living
187. **Mikko Pänkäälä**, Potential and Challenges of Analog Reconfigurable Computation in Modern and Future CMOS
188. **Sami Hyrynsalmi**, Letters from the War of Ecosystems – An Analysis of Independent Software Vendors in Mobile Application Marketplaces
189. **Seppo Pulkkinen**, Efficient Optimization Algorithms for Nonlinear Data Analysis
190. **Sami Pyöttiälä**, Optimization and Measuring Techniques for Collect-and-Place Machines in Printed Circuit Board Industry

Turku Centre *for* Computer Science

Joukahaisenkatu 3-5 B, 20520 Turku, Finland | www. tucs.fi

University of Turku
Faculty of Mathematics and Natural Sciences
- Department of Information Technology
- Department of Mathematics and Statistics

Turku School of Economics
- Institute of Information Systems Science

Åbo Akademi University
Division for Natural Sciences and Technology
- Department of Information Technologies

ISBN 978-952-12-3158-2
ISSN 1239-1883

Sami Pyöttiälä

Optimization and Measuring Techniques for Collect-and-Place Machines in Printed Circuit Board Industry