# A CASE STUDY: IMPROVEMENT OF COMPONENT PLACEMENT SEQUENCE OF A TURRET STYLE SMT MACHINE

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MEHMET SAVAŞ ÇENGEL

IN PARTIAL FULFILLMENT OF THE REQUIRMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

DECEMBER 2006

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Çağlar Güven
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Pelin Bayındır
Co-Supervisor

Assoc. Prof. Dr. Levent Kandiller
Supervisor

**Examining Committee Members**

| | |
|---|---|
| Assoc. Prof. Dr. Sedef Meral | (METU, IE) |
| Assoc. Prof. Dr. Levent Kandiller | (METU, IE) |
| Assoc. Prof. Dr. Haldun Süral | (METU, IE) |
| Assist. Prof. Dr. Pelin Bayındır | (METU, IE) |
| Aydın Kaynarca( M.S.) | (KAREL) |

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Last name : Mehmet Savaş Çengel

Signature :

# ABSTRACT

# A CASE STUDY: IMPROVEMENT OF COMPONENT PLACEMENT SEQUENCE OF A TURRET STYLE SMT MACHINE

Çengel, Mehmet Savaş

M.S. Department of Industrial Engineering

Supervisor: Assoc. Prof. Dr. Levent Kandiller

Co-supervisor: Assis. Prof. Dr. Z. Pelin Bayındır

December 2006, 119 pages

This study aims to improve component placement sequencing of a number of PCBs produced on a turret style SMT machine. After modeling the problem and having found that an optimal solution to the real PCB problem is hard to be achieved because of the concurrent behavior of the machine and the PCB design parameters, two heuristics are developed by oversimplifying the problem down to TSP. Performance of the heuristics and the lower bounds is evaluated by comparing the results with the optimal solution for two sets of randomly generated PCBs. The heuristic solutions are also compared with the lower bounds and the current implementation for the real PCBs. It is found out that the heuristics improve the current efficiency figures of the company.

Keywords: TSP, heuristics, optimization, PCB assembly.

# ÖZ

## VAKA ÇALIŞMASI: TARET TİPİ YÜZEY MONTE MAKİNASINDA BİLEŞEN YERLEŞTİRME SIRALAMASININ İYİLEŞTİRİLMESİ

Çengel, Mehmet Savaş

Yüksek Lisans, Endüstri Mühendisliği Bölümü

Tez Yöneticisi: Doç. Dr. Levent Kandiller

Yardımcı Tez Yöneticisi: Yrd. Doç. Dr. Z. Pelin Bayındır

Aralık 2006, 119 sayfa

Bu çalışma, taret tipi yüzey monte makinesi tarafından üretilen bir gurup baskıdevre kartının parça yerleştirme sıralamasının iyileştirilmesini amaçlamaktadır. Modellenen problemde, gerçek kartlara ait en iyi sonucun, makinenin ardışık ve birbirine bağımlı hareketleri ile baskıdevre kartın tasarım verilerinden kaynaklanan karmaşıklık sonucu, kabul edilen sürelerde elde edilememesi sonrasında iki sezgisel yöntem geliştirilmiştir. Bu sezgisel yöntemler, problemin gezgin satıcı problemine dönüştürülmesi ile oluşturulmuştur. Sezgisel yöntemlerin ve bunların kıyaslanması ve oluşturulan alt sınırların performansını değerlendirebilmek için iki küme rassal baskıdevre kartına ait en iyi sonuçlar bulunmuştur. Ayrıca, gerçek kartlar için bulunan sezgisel sonuçlar ve belirlenen alt sınırlar, bu kartların halihazırdaki üretim hızı ile karşılaştırılmış ve sezgisel yöntemlerin makine verimini geliştirme düzeyi ortaya konmuştur.

Anahtar Kelimeler: Gezici Satıcı Problemi, sezgisel yöntemler, optimizasyon,
Baskıdevre kart montajı.

To My Parents and My Wife

# ACKNOWLEDGMENTS

It was a pleasure and honor to study with Assoc. Prof. Dr. Levent Kandiller and Asist. Prof. Dr. Pelin Bayındır. With their helps, encourageous attitude and endless support I have been succeeded. I know I can not thank them enough, although these few words can express my appreciation to them.

SMT production was an unknown to me, I would like to thank to Bugra Karahan who taught all the concepts, definitions and the technology behind the production environment. His patient and kind approach helped me so much in defining the complex problem. I would like to thank to Taner Durucan, who has given all his support, even he has shared some of my work during my absence in the company I worked as the study continued. Aydin Kaynarca has helped me endlessly, I would like to thank to him and wishing him luck for the future.

Technical assistance of Oguz Solyali regarding the CONCORDE program is acknowledged.

My family always had faith in me; I would like to thank to my mother and father for their full support even in my worst times. Finally; I would like to thank to my wife who cares for and loves me all the time.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

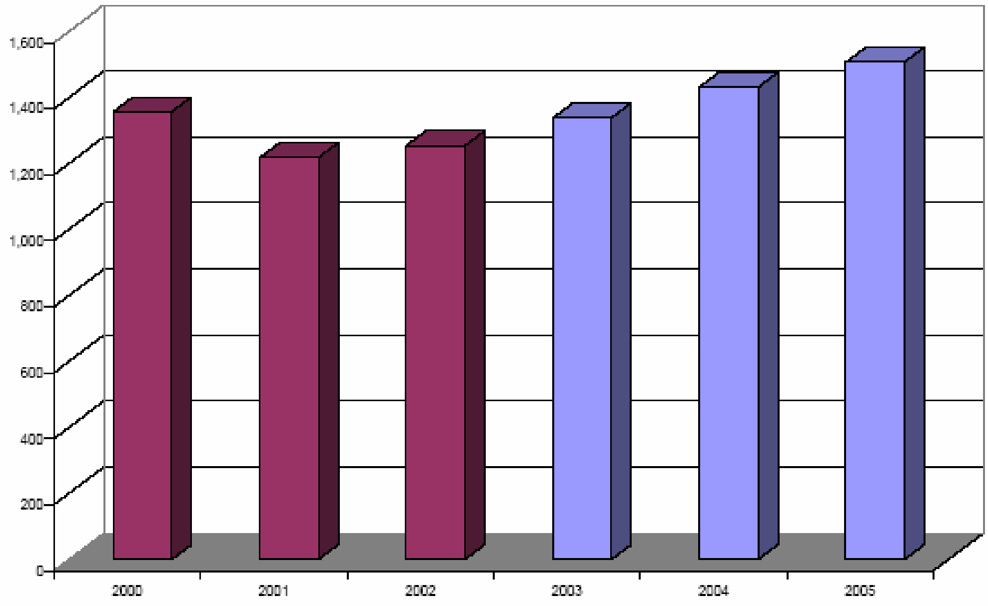
# LIST OF ABBREVIATIONS

| | |
|---|---|
| B&B | : Branch and Bound Algorithm. |
| CONCORDE | : Computer code for the symmetric traveling salesman problem (TSP) and some related network optimization problems. |
| CPLEX | : Linear programming optimizer developed in the C programming language. |
| DPP | : Dynamic pick–and –place point. |
| FPP | : Fixed pick–and –place point. |
| FSA | : Feeder Slot Allocation. |
| HSCS | : High speed chip shooter. |
| MWMP | : Minimum weight matching problem. |
| PCB | : Printed circuit board. |
| PCTSP | : Precedence constrained traveling salesman problem. |
| QAP | : Quadratic assignment problem. |
| SCARA | : Single compliance robot for assembling. |
| SMT | : Surface mount technology. |
| TRAVEL | : A software that uses a combination of 2-opt, 3-opt and Lin-Kernighan algorithms to solve traveling salesman problem. |
| TSP | : Traveling salesman problem. |

# CHAPTER 1

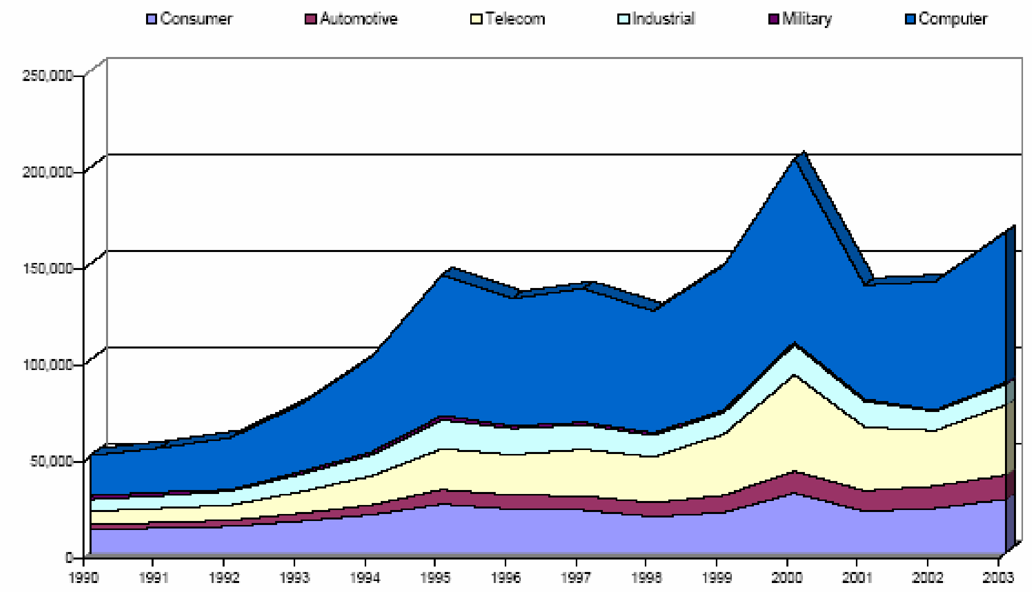
# INTRODUCTION

World's electronics production increases every year due to high volume usage in daily life, and technology (see Figure 1.1). As a key input to the electronics industry, semi-conductors trade can be seen as a lead indicator of market trends. Figure 1.2 shows the historical pace of the sector for different market segments.



Source: Reed Electronics Group (2003) *Electronics Market Outlook Presentation.*

Figure 1.1. World electronics market, 2000 to 2005, million USD [32]

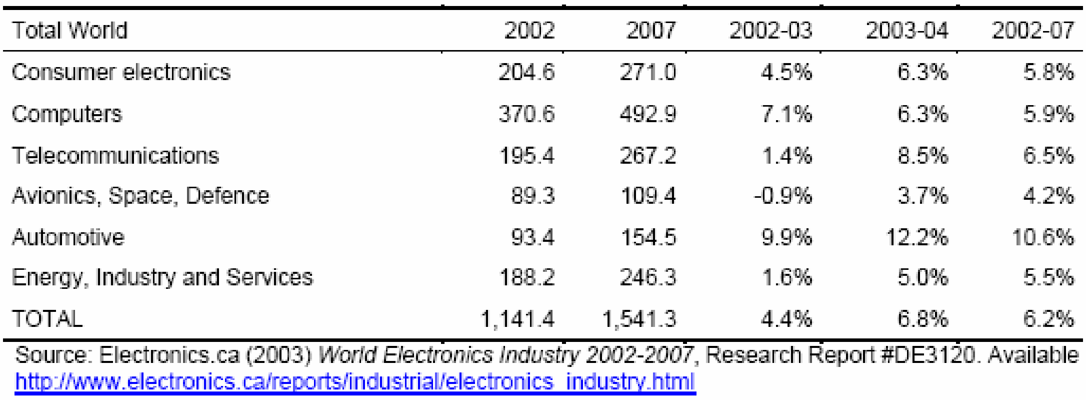


Source: World Semiconductor Trade Statistics. OECD (2004) *Information Technology Outlook 2004*, OECD, Paris.

Figure 1.2 – Worldwide semiconductor market by segment, 1990 to 2003 (million USD) [32]

Increases in the sales also imply the increase in the usage of the electronics in other areas of production (see Table 1.1).

Table 1.1. World production by sectors, 2002-2007 (billion EUROs and percent growth) [32]



| Total World | 2002 | 2007 | 2002-03 | 2003-04 | 2002-07 |
|---|---|---|---|---|---|
| Consumer electronics | 204.6 | 271.0 | 4.5% | 6.3% | 5.8% |
| Computers | 370.6 | 492.9 | 7.1% | 6.3% | 5.9% |
| Telecommunications | 195.4 | 267.2 | 1.4% | 8.5% | 6.5% |
| Avionics, Space, Defence | 89.3 | 109.4 | -0.9% | 3.7% | 4.2% |
| Automotive | 93.4 | 154.5 | 9.9% | 12.2% | 10.6% |
| Energy, Industry and Services | 188.2 | 246.3 | 1.6% | 5.0% | 5.5% |
| TOTAL | 1,141.4 | 1,541.3 | 4.4% | 6.8% | 6.2% |

Source: Electronics.ca (2003) *World Electronics Industry 2002-2007*, Research Report #DE3120. Available http://www.electronics.ca/reports/industrial/electronics_industry.html

Since the electronic products have become more critical inputs in terms of both cost figures and functional effectiveness, new technologies have been developed in order to speed up the manufacturing lines as well as the functional output of an electronic semi-finished good. Surface mount technology (SMT) is one of these technologies. It involves the use of tiny components to be placed so that tighter layouts can be achieved on the printed circuit board (PCB). SMT products mainly lead to finished goods of smaller sizes. It also brings an advantage in terms of manufacturing performance, since SMT production, compared to the prior production techniques like through hole, axial insertion, can produce higher amounts in less time. Being widely used for more than 20 years [1], certain improvements in SMT manufacturing technology have been achieved. General process flow for SMT manufacturing is depicted in Figure 1.3. The main steps are: the placement of the components on a pre-pasted PCB (screen printing process: solder paste is printed over PCB by a print-screen machine – see Figure 1.4) and passing it through an oven (re-flow oven – see Figure 1.4) on a basis of a heating profile. The placement process can be divided into subprocesses according to the component characteristics (such as component class, shape, packaging type) while assembly line balance is taken into account.

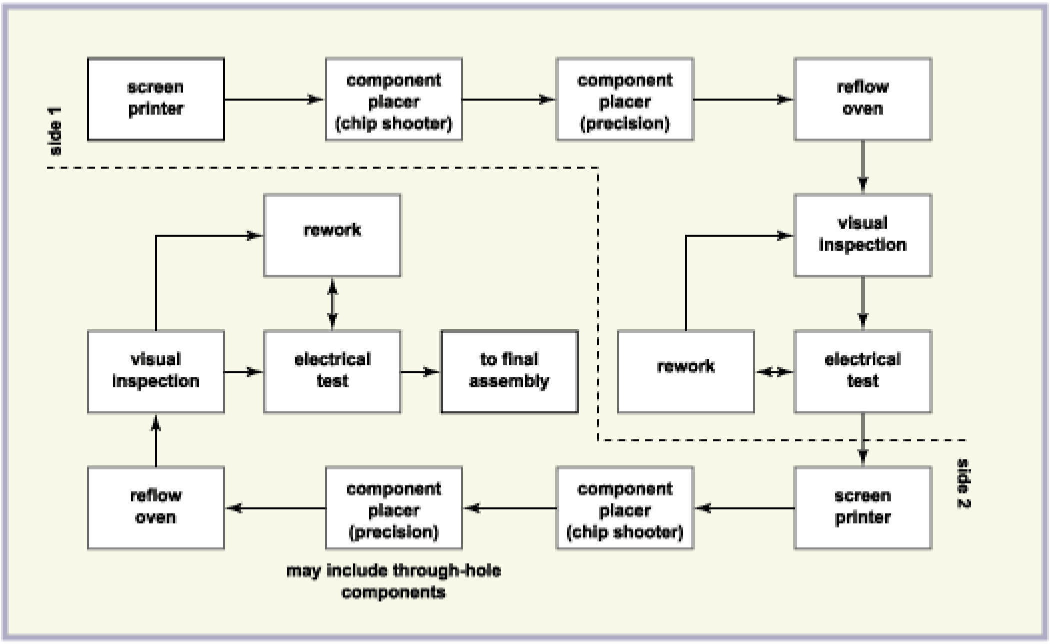


Figure 1.3 – General process flow for SMT manufacturing line

A surface mounting line is usually the vital part of the electronics production and costs more than a million USD constituting the main investment for most of the electronics companies. SMT line feeds other processes such as optical inspection,

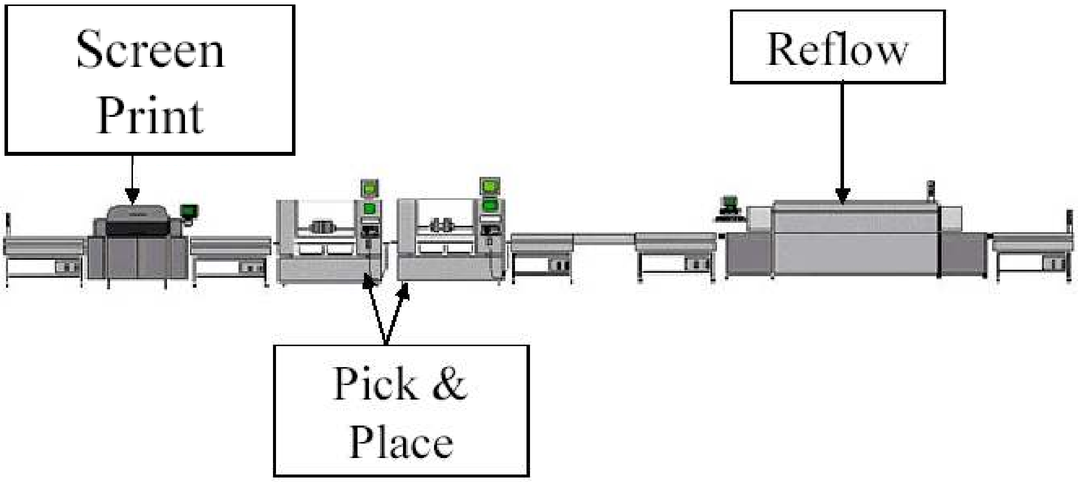


Figure 1.4 – Automated SMT line

assembly line, functional test, in-circuit test, mechanical assembly, packaging. In most of the production environments, the SMT line is the bottleneck process. Our study focuses on pick and place SMT machines, which are the most expensive part of an SMT assembly line in terms of investment. These machines are capable of processing in high volumes (components/hour) and placing the components on the PCBs with higher precision. Therefore, full utilization of these machines is desirable in this manufacturing environment. Utilization, classically measured as the usage of the machine per shift/day (in terms of hours/day), is an inadequate performance measure since what matters is the amount of PCBs manufactured and/or components placed per shift/day.

The main motivation of this study is to investigate ways of increasing productivity of the placement process based on the scientific method. We explore the problem environment using the following terms and definitions.

Performance of an SMT manufacturing is the result of a production plan that considers the main issues below [2]:

- characteristics of the equipment (layout, number of machines, details of the operating mode, etc.),
- characteristics of the product mix (diversity of PCB types, batch sizes, etc.),
- managerial policies like the frequency of setups or the willingness to redesign the lines on a regular basis.

An SMT machine consists of three main units: a worktable, a feeder carrier, and a pick and place device. The ***worktable*** is the place where the PCB is attached during the placement operations. The ***feeder carrier*** is the unit through which the component feeders are fed. The ***pick and place device*** is the mechanism that picks the components from the feeder carrier and places the component on the PCB that is attached to the worktable. The way these main units work together may differ; for instance, the worktable may be fixed or movable in X or X-Y direction; the feeder carrier may be fixed or move along X-axis while pick and place mechanism can move in Y direction or X-Y directions to pick up the components from the feeders or a set of heads that can pick and place components may rotate 360 degrees. Machine types can be classified according to the picking and/or placing mechanism: dual delivery placement machine, multi-station placement machine, turret style placement machine, multi-head placement machine and sequential pick-and-place machines. These are the types of machines mainly studied in the relevant literature and an interested reader is referred to the survey in [3].

Four factors are considered in processing PCBs: component location, component type, PCB size and component placement angle. These input data determine the setup for each PCB type, which is transformed into a machine code that specifies the mechanical movements for the sequence of pick up and placement of the components. Diversity of PCB products requires grouping as a family, since each PCB family uses the same feeder setup. Small batch sizes of PCB constitute another reason for forming families to save the setup time/cost of feeder changes.

As multiple SMT lines exist and a variety of PCB types have to be manufactured, the manufacturer has to decide on the product mix that an SMT line can handle according to the above factors. Since each PCB type uses a different set of component types and the machine types might have different characteristics (a machine can mount a given range of component types), assigning the PCB types into SMT lines have to be done as the first design decision. As the groups are formed, the setup strategy for the SMT line has to be covered and a sequence of the groups has to be formed. Assuming that the long term decisions (on SMT line investments and setup policy) are already made and the shop floor and product mix are fixed, Crama et al. [2] build up a perfect hierarchical approach to cover the decisions regarding the planning of PCB manufacturing. The decisions to be made should cover the answers to eight sub-problems (SP) as given below:

SP1. assignment of PCB types to product families and product families to machine groups (cells or lines).

SP2. allocation of component feeders to machines.

SP3. for each PCB type, partition of the set of component locations on the board type, indicating which components are to be placed by each machine.

SP4. for each machine group, sequence of the PCB types, indicating the order the board types are to be produced on these machines.

SP5. for each machine, location of feeders on the carrier.

SP6. for each pair consisting of a machine and a PCB type, component placement sequence; that is, a sequence of the placement operations to be performed by the machine on this board type.

SP7. for each pair consisting of a machine and a PCB type, component retrieval plan; that is, for each component on the board, a rule indicating from which feeder this component should be retrieved.

SP8. for each pair consisting of a machine and a PCB type, motion control specification; that is, for each component, a specification of where the pick-and-place device should be located when it picks or places the component.

In this thesis, we focus on the problem of component placement sequence for a given machine and PCB type, that is SP6 under Crama's [2] hierarchy. The study is motivated from a real–life situation observed in one of the leading electronics companies in Turkey. The main objective of the study is to improve the operations of an SMT machine. The problem is finding the best tour that minimizes the component placement sequencing of PCBs given the machine specifications and managerial decisions. Since the underlying optimization problem is difficult to solve (optimal solution could not be found in 14 days for a small sized PCB, where the optimization tool evaluates only 37% of the possible solutions in the branch&cut tree in that time period), we propose heuristic approaches. The performance of the heuristics is tested for real PCBs that the company produces. The results are compared with both the optimal solutions of the restricted number of real cards based instances and randomly generated problem instances under a modified machine specifications (which also reduces the computational effort required to get the optimal solutions). In order to make a performance assessment based on the problem instances for which the optimal solution could not been obtained in a reasonable time, we proposed two lower bounding schemes. The performance of the heuristics is compared with the current placement sequence for the company's machine and PCB pairs as well as the heuristic results are compared with the lower bounds. The computational study on the PCBs that are currently on the product mix of the company shows that the heuristics result in the placement sequences better than the current practise. It is found out that the proposed heuristics find near optimal results for the randomly generated PCBs. However, the computational study with these randomly generated PCBs shows that there can be a high gap between lower bounds and the optimal solution values.

The thesis includes five chapters. In Chapter 2, a literature survey on SMT machine optimization techniques and different approaches to the SP6 problem are presented following the framework provided by Crama et. al. [2]. In Chapter 3, a detailed description of the problem environment, the problem definition, the proposed heuristic approaches and the lower bounding schemes are given. Chapter 4 involves a computational study that is carried out to assess the performance of the heuristics and lower bounding schemes. In Chapter 5, a summary of the work done is given and future research directions are addressed.

# CHAPTER 2

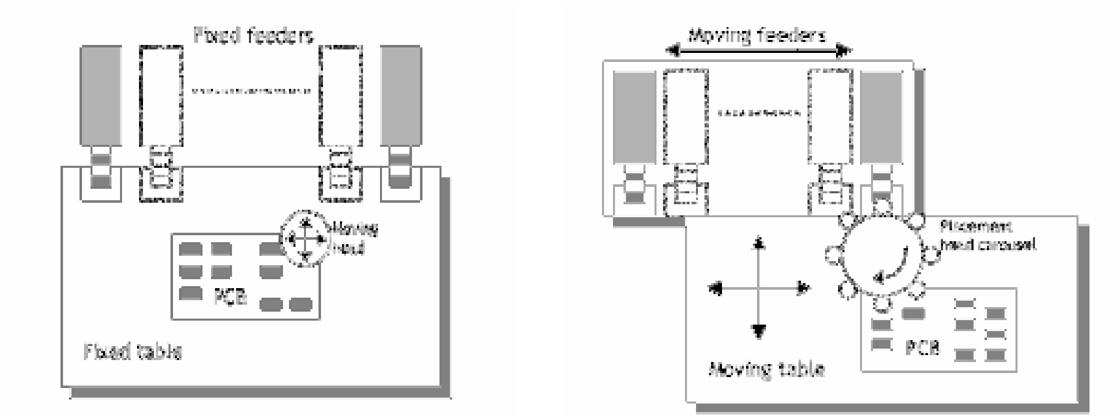
# LITERATURE REVIEW

## 2.1 Background

A family setup is made for an SMT machine assigned to a group of PCBs. This strategy (as explained by McGinnis et al. [5] and Ammons et al. [11]) eliminates the need for machine setups between the manufacturing of different PCB types at the cost of potentially increasing the assembly time for each individual PCB type in the family. In order to apply a family setup strategy, one has to solve the grouping problem that determines the setup policy for a machine (i.e. feeder placement). Then, for each PCB in the family, the placement sequence of the components with respect to the fixed setup (i.e. the location of feeders) is planned. In the related literature, the studies about the family setup policy focus on the problem of grouping PCBs in order to minimize the setups while the placement sequencing for each PCB is also optimized as well.

The researchers have tried to overcome the complexity of the problem by generalizing the problem or simplifying it by making some assumptions. While some researchers have tried to decompose the overall problem into subproblems to reach an approximately optimal solution, the problem is directly attacked to be solved by means of heuristic methods (in some studies).

SMT machines are of various types and can be classified with respect to the mechanical structure as follows [4]: (i) single compliance robot for assembling (SCARA) (pick and place machines), (ii) cartesian/gantry and (iii) high speed chip shooter (HSCS) (turret head) [4]. Operational characteristics of the machines is another classification criterion: concurrent vs. sequential machines. The distinction comes from the machine cycle as defined in [5], concurrent machines operate a series of consecutive operations beginning with a component pick and ending with a component placement, while sequential machine cycle consists of only one pick and one placement. From this point of view, sequential machines (i.e. pick and place machines, see Figure 2.1.a) are those whose machine cycle involves exactly one component (the same component is picked and immediately placed) while concurrent machines (i.e. rotary turret machines, see Figure 2.1.b) are those whose cycle involves the pick of one component type and the placement of a component that is previously picked.



2.1.a Pick and place machine (sequential) 2.1.b Rotary turret machine (concurrent)

Figure 2.1 – SMT Machine types

Other classifications are fixed pick-and-place point (FPP) vs. dynamic pick-and-place point (DPP) [6], insertion vs. pick-and-place vs. rotary turret machines [7], turret-head vs. pick-and-place vs. pick-and-place with rotary head [8], multi-head vs. high speed chip shooter machine (HCHS) vs. robotic arm placement machine [9]. In order to fully understand the features of the machines and to design new optimization methods for these machines, the classification criteria have to be more detailed in terms of operational activities and concurrent characteristics of the machines. Thus we follow dual-delivery, multi-station, turret-type, multi-head and sequential pick-and-place classification [3] as the framework in this study.

After various studies concerning the problems faced in electronics production have been conducted; Crama et al. [2] presented a detailed hierarchy where planning problems in electronics production are divided into eight subproblems. These are:

SP1. assignment of PCB types to product families and product families to machine groups (cells or lines).

SP2. allocation of component feeders to machines.

SP3. for each PCB type, partition of the set of component locations on the board type, indicating which components are to be placed by each machine.

SP4. for each machine group, sequence of the PCB types, indicating the order the board types are to be produced on these machines.

SP5. for each machine, location of feeders on the carrier.

SP6. for each pair consisting of a machine and a PCB type, component placement sequence; that is, a sequence of the placement operations to be performed by the machine on this board type.

SP7. for each pair consisting of a machine and a PCB type, component retrieval plan; that is, for each component on the board, a rule indicating from which feeder this component should be retrieved.

SP8. for each pair consisting of a machine and a PCB type, motion control specification; that is, for each component, a specification of where the pick-and-place device should be located when it picks or places the component.

SP1 deals with the grouping problem, SP2-SP4 deal with problems to be solved for each group, SP5-SP8 are related to a particular machine type with a single PCB type. The difficulty with the above hierarchy is the tight relationship of subproblems with each other. The way to overcome this problem is the selection of a group of subproblems to be dealt with while the other subproblems are solved beforehand. The operational decisions are seen in the subproblems SP5-SP8 where one makes decisions on the manufacturing process, as the others can be viewed as mid-term decisions (SP4) and results of long term decisions given by top management (SP1-SP3). Although these strategic decisions affect the performance of the manufacturing system, focusing on the manufacturing variables in order to optimize the assembly (SP5-SP8) as other decisions are made beforehand is seen more realistic.

In the literature, it is seen that most researchers focus on SP5 and SP6 of Crama's hierarchical framework. Motion control (SP8) subproblem is seen as a part of preparation for production [10], and researchers are interested in the machine types having no motion control problem. Motion control is related to the movable heads that can move in X-Y direction with a PCB carrier moving on X direction, where machines having fixed pick and place points have motion control problem related to the table movement in X-Y directions which can be formulated into the component placement sequencing, i.e., turret-type SMT placement machines. Thus, there are less studies covering the motion control subproblem compared to those related to SP5 and SP6.

Component retrieval (SP7) subproblem is even less studided in the literature, because it requires the assignment of one component type to several feeders (duplication of component types) which might be seen as a managerial decision to make before operational planning takes place. In addition, it is related to the operational characteristics of the SMT placement machines. For most sequential machines, it can be modeled and solved within the placement sequencing subproblem. The same holds true for the turret-type SMT placement machine if the start of a pick activity coincides with the start of a place activity.

Single machine single PCB type problems have to deal with the operational characteristics of the SMT machine, hence model formulation, assumptions and solution techniques vary according to the characteristics of the machine considered [2], [9], [14]. There are many studies on this problem in the literature. Feeder arrangement, placement sequencing and component retrieval are the basic concerns of these studies, as SP8 (motion control) problem in Crama et al [2]'s classification is included in the placement sequencing subproblem. As a number of components (of different types with respect to size and shape) have to be located on specific points on a PCB, an optimal solution should handle all these concerns together. It is seen in the previous works that solving component placement sequencing alone is a traveling salesman problem (TSP).

TSP is defined as follows: given $n$ cities and their intermediate distances, find a shortest route traversing each city exactly once [33]. By defining locations on a PCB as cities and the time between placements (or insertions) as distances, the component placement sequencing problem can be formulated as a TSP [16]. The classical formulation of TSP is:

$$\begin{aligned}
&\min \\
&\sum_{l=1}^{n}\sum_{k=1}^{n} d_{lk}x_{lk} \\
&\text{s.t.} \\
&\sum_{k=1}^{n} x_{lk} = 1 \qquad \text{for } l = \{1,2,...,n\} \\
&\sum_{l=1}^{n} x_{lk} = 1 \qquad \text{for } k = \{1,2,...,n\} \\
&\sum_{l\in S}\sum_{k\in S} x_{lk} \le |S|-1 \quad \text{for all } S \subseteq \{2,...,n\} \\
&x_{lk} = 0 \text{ or } 1 \quad \text{for all } l,k
\end{aligned}$$

Although powerful tools to solve the TSP have been developed [31], as the amount of locations on a PCB increase, the optimal tour is hard to be found. The authors address that TSP is an NP-Hard optimization problem that is difficult to solve for some practical instances in a reasonable time [7], [9], [13], [15], [27], [30]. The determination of the distance between locations on a PCB, e.g. on turret style machine types, forces the entrance of new decision variables into the formulation because of the concurrent behavior of the movements (which will be discussed in Chapter 3). When other subproblems are included, such as feeder assignment problem, mathematical programming approach for the overall design problem induces a computational burden [12]. Given that the optimal solution is hard to be found, researchers construct heuristics to obtain near optimal solutions and heuristic solutions are compared with the lower bounds. Heuristic algorithms that attempt to find feasible solutions step-by-step are called construction heuristics while algorithms that modify or improve some given starting solution are called improvement heuristics [34]. Lawler [35] discusses some useful construction heuristics such as nearest neighbor, furthest insertion point, arbitrary insertion point as well as improvement heuristics such as 2-opt, 3-opt and or-opt for TSP. He also provides comparisons on the quality of the solutions. Another extensive review on heuristics for TSP is by Applegate et al. [36]. They schematically explain the algorithms on pseudo-codes.

The literature review related to the placement sequencing subproblem and feeder assignment subproblem is presented in Sections 2.2 and 2.3, respectively. Some authors study feeder arrangement and placement sequencing subproblems together; they use a solution technique such that first one of the subproblems is solved separately (using heuristics), then based on this initial solution another heuristics is applied to the other subproblem. The solutions are tried to be improved in an iterative manner. These studies are reviewed in Section 2.4.

## 2.2. Placement Sequencing Subproblem:

TSP formulation is used in placement sequencing subproblem by almost all of the researchers, however they all obtain near optimal solutions.

Chan and Mercier [16] study a special insertion machine that feeds the head (movable on Z axis) with the components and a PCB table moving on X-Y axis. They assume that the feeder movement is faster than the PCB table. Since the components are arranged beforehand for the header the component placement sequencing problem boils down to a TSP. The model is solved by TRAVEL software using a combination of 2-opt, 3-opt and Lin-Kernighan algorithms.

Moyer and Gupta [17] study the subproblem on a rotary turret machine. For the traveling time between locations, Euclidean metric is used instead of Tchebycheff's metric. It is assumed that the table movement time is longer than the feeder exchange time. The placement sequencing subproblem is formulated as a two dimensional TSP. For the solution, an initial placement sequence is taken as input and a pairwise exchange algorithm is applied. The initial sequence is generated via three methods: random selection, increasing component type identifier; and sorting the components over the PCB according to the components' location point definitions; X and then Y coordinates. The initial placement sequence is improved by exchanging the components according to the proposed pairwise exchange algorithm.

Duman and Or [18] study a specific machine where the components are ready to be picked. The machine in concern limits the placement sequencing since the placement head can damage the PCB depending on the sequence. The formulation is a special kind of TSP named as precedence constrained TSP (PCTSP), where a constraint regarding the prevention of component damage is added. Tchebycheff's distance is considered in this study. For the solution, a two step heuristic is proposed. First an approximate TSP solution is found, then a heuristic is applied for damage reduction. The performance of the proposed method is tested on a real production environment. It is shown that component damage is decreased and the throughput figures are increased.

## 2.3. Feeder Arrangement Subproblem:

Ahmadi [19] assumes that the head and the PCB table move in a constant time. As placing of a component depends on the maximum of this constant time and feeder movement time, he claims that the feeder arrangement would affect the assembly time more than the placement sequence. The shortest path of a set of nodes containing the possible locations of the feeders in the feeder carriers is found as the solution.

Some researchers have tried to solve the problem by solving component sequencing subproblem first or by fixing the component sequence. Leipala and Nevalainen [20] and Francis et al. [21] formulate the subproblem as a quadratic assignment problem (QAP). Foulds and Hamacher [22] model the subproblem as a bin location assignment which is formulated as a single facility location problem.

Crama [23] studies the subproblem on two turret rotary machines aiming to balance the workload by assignment of feeders and finding a placement sequence for each PCB. It is assumed that a certain component type may be carried on two feeders. The feeders that can carry the components are found by a clustering heuristic. An arbitrary assignment of feeders for each machine’s slot is made as initial assignment. The placement time for the assignment is estimated by assuming that the feeders move from left to right; i.e., for each PCB the components in each feeder is finished first, then the feeder moves to the next. After estimation of placement time for each PCB with the feeder assignment, workload balance is done by using two heuristics that consider the exchange of feeder types with their component clusters between the machines. An iterative search is performed as a heuristic and the placement times are estimated for each assignment cycle.

Moyer and Gupta [4] solve the feeder assignment subproblem on a rotary turret machine for a given placement sequence. A QAP formulation, where the departments are component feeders and possible department locations are feeder slot locations, is applied. Feeder Slot Allocation (FSA) heuristic considers the flow between any pair of components and a weighting scheme. In order to compare results of FSA heuristic pairwise exchange algorithm is considered. Although pairwise exchange algorithm gives better result, FSA solves the problem in a shorter time.

Sun [14] aims to maximize a component’s simultaneous pick activity, i.e., a component type’s placement is tried to be arranged such that picking activity for that component type again is minimized. Using a genetic algorithm, feeder traveling time is reduced.

## 2.4. Feeder Arrangement and Placement Sequencing

The feeder arrangement and placement sequencing subproblems are interrelated. The setup strategy for a machine type might necessitate the component feeders to be fixed at a certain position. In other words, since feeder arrangement might take longer times compared to production of a single PCB for a large product mix, it would be reasonable to assume that feeder arrangement is given. Single setup strategy uses a fixed feeder arrangement for the production of a family of PCBs. The feeders are arranged such that all components required to produce each PCB in that family are available at that machine. This strategy can be seen in two ways: unique setup strategy (the family consists of one PCB, i.e. high volume/low diversity of production) and family setup strategy (the family has more than one PCB, i.e. low volume/high diversity of production) [5].

Due to machine characteristics, component feeder movement might take much longer time than placement of components. This assumption is seen in some papers presented in Section 2.3, since the assumption forces one to deal with only the feeder arrangement subproblem if a placement sequence is given. It should be noted that this assumption might not be valid for the machinery developed in recent years and the strength of the assumption must be tested for different placement sequencing instead of considering one case of sequences.

Another scope that affects the description of the problem lies in the functional performance of a machine. Note that the feeder movement and placement time for a component depends on the head, i.e., the time interval for a placement may be described as a sum of two events. This may change the optimization model such that the feeder arrangement might simply be included in the TSP formulation. On the other hand, for concurrent type of machines, the updated model is harder since feeder movement and component placement sequencing seem to occur independently. Sequential machines' feeder arrangement subproblem might be embedded into the formulation of the placement sequencing subproblem.

It is stated by several authors ([2], [3]) that the feeder arrangement affects placement sequencing since the time to pick a component by a head lags the placement operation performed by the same head or vice versa.

As being the pioneering work on PCB assembly optimization, Ball and Magazine [15] divide the overall problem into two parts for a sequential pick & place SMT machine: feeder arrangement and component insertion sequence. Feeder assignment is modelled as a linear assignment problem as the total placement time of all the components retrieved from a given feeder is roughly approximated. For placement sequencing problem, a network of nodes, where each node starts at a feeder location and end at a component location is developed. Movements between the nodes are described as nodes that are formed as movements from feeders to components (placing operation), and nodes that are formed as movements from components to feeders (picking operation).

Placing operation is named as “required” because as the component types are assigned to the feeders, these movements have to be done, since the head must move from the feeder (one component type has one feeder assigned) containing the component type to each component location. Picking operation is named as “non-required” and it depends on the placement sequencing. The authors model the problem as a special case of traveling salesman problem (TSP): Rural Postman Problem or Stacker Crane problem. The network of possible nodes and the types of possible movements are defined, by applying an Eulerian tour algorithm, a closed path between the nodes that minimizes the distance traveled by the placement head is found. The metrics used in the formulation affects the solution. If the placement head moves only in one orthogonal direction at a time, the Manhattan metric is used for measuring the length of the movements and one can find an optimal solution in polynomial time. If the head can move in both of X and Y-directions independently, Tchebycheff’s metric can be applied for the distance calculations, but the solutions to be obtained are then suboptimal. However, in the latter case, Ball and Magazine [15] prove that the maximum error of the solution is bounded.

Leipala and Nevalainen [20] deal with placement sequence and feeder arrangement separately on a rotary turret machine with two heads and a moving feeder carriage. Placement sequence is modelled as a three dimensional asymmetric TSP problem and the solution is found using a heuristic that is a version of furthest insertion point heuristic for an initial feeder assignment. This initial placement sequence is used to solve the feeder arrangement subproblem which is modeled as a QAP. A heuristic that exchanges component feeders pairwise is proposed for QAP. Iterations of the two heuristics continue until the feeder arrangement results with no improvement in placement sequence.

Leu [24] considers insertion and sequential pick and place machines. In insertion machines, the components to be placed are sequenced into a feeder tape by a sequencer beforehand, i.e. there is no feeder arrangement problem. The placement problem is just the TSP. For sequential pick and place machine, the problem can be modeled as a rural postman, linear assignment or QAP. For a rotary turret machine, a genetic algorithm is proposed. The algorithm includes solving feeder assignment and placement sequencing iteratively; the results are combined in order to find the optimal placement time.

Kumar and Li [25] study placement sequencing and feeder arrangement subproblems for sequential pick and place machines. They use a combined model of TSP and minimum weight matching problem (MWMP). Lower and upper bounds are calculated to compare the results found. Upper bound for the sequencing is found by solving the TSP for a given feeder assignment and upper bound for the feeder assignment is found by solving MWMP for a given component sequence. Lower bounds are calculated by means of linear programming and Lagrangian relaxation techniques. MWMP is solved using a software program, and initial component sequencing is found by nearest neighbor, nearest insertion, furthest insertion and random generation heuristics. 2-opt and 3-opt are used for the improvement of initial solution. The results found are also compared with the solutions obtained by the S shape and greedy algorithms.

De Souza and Lijun [26], for a rotary turret style machine, form a knowledge based system for component sequencing and feeder assignment subproblems. The components are grouped according to their types, quantity and sizes. Arrangement of the components is done considering the groups and the components' location on PCB. For the component sequence subproblem, a heuristic is formed as a combination of minimum spanning tree and knowledge based heuristic which guarantees a reasonable worst-case performance. With this heuristic, a placement sequence for each group of components is obtained. The solutions for the groups are combined in the end.

Gupta and Moyer [12] study rotary turret machine and use nearest neighbor algorithm for obtaining an initial placement sequence. The sequence is improved by means of a pairwise exchange algorithm until all the pairs are tried or a prespecified number of solutions that do not improve the sequence are observed. Each improved solution is recorded to be used in feeder assignment subproblem. The initial feeder assignment is based on the component ID numbers. Given the initial placement sequence, the feeder arrangement is tried to be improved according to the pairwise exchange algorithm with the same limitation stated above. After finishing the work with the initial sequence, other sequences in the record list are tried in the same manner. The solution is the one that gives the minimum placement time. The work done is tested against algorithms presented by Leu et al. [24] and De Souza and Lijun [26]. The main shortcoming of the study is the assumed traveling time (between two placement events), the authors do not construct the distance as the maximum time of three concurrent events (PCB table movement, feeder carriage movement, turret movement).

Ng [27] develops heuristics to solve feeder assignment and placement sequence problems for a rotary turret machine. The research is based on the following assumptions: (i) the feeder movement starts with a certain feeder position and the feeder movement is defined beforehand such that the feeder locations are filled with component types using that movement. (ii) The feeder movement by one feeder position is more significant than any placement movement. (iii) Movements are linear. The main idea is partitioning the locations into component types and modeling a connectivity and distance relation between the types using the functions and heuristics. Assignment of feeders is modeled as a group TSP each of which is solved by a nearest insertion algorithm. With the initial feeder assignment total placement time is tried to be minimized using the nearest insertion algorithm. With an exchange technique, where the exchange is limited with two feeder rack positions at a time, total placement time is tried to be improved. Ng [27] also tries to solve the problem of adding another feeder reel for a component type, using a weighted sum of the factors that affect the placement time, the component type and the place of the feeder is calculated. The heuristics' results are compared with the existing solution and a lower bound found by experience.

Ahmadi and Mamer [28] work on a sequential pick and place machine, and propose heuristics for feeder assignment and placement sequencing problem. The heuristics constructed solve TSP for each component type independently and then patch the tours together. Since cost of patching the tours depend on the assignment of end-start point for each part type, for a group of locations formed between different part types, a TSP is formulated to find the shortest path where in the end, the end-start points for each part type is calculated. For both TSP models, different heuristics are tried (nearest neighbour, greedy algorithm, S-shape methods, space-filling curves) and the results are tried to be improved by 2-opt heuristics. Different types and numbers of PCBs (with different shapes and number of components) are solved.

Altinkemer et al. [29], for a multi-head machine, try to solve the subproblems in an integrated approach. They model the overall problem as a vehicle routing problem. Two cases are investigated: the feeder moves and does not move. Since non-moving feeder case is a more complex environment, certain assumptions had to be made for the modeling. The integrated algorithm proposed consists of solving a vehicle routing problem for each component type at every possible feeder location, and an assignment problem (where the costs are obtained in the previous step) is solved for the placement sequencing problem. They show that the proposed algorithm provides feasible solutions with an error gap less than or equal to the maximum error gap of the vehicle routing problem.

Ellis [30] solves the subproblems in a number of iterations for rotary turret head SMT machine. The component types according to the allowable machine’s table and turret rotation speed are grouped. An initial placement sequence for each group is formed using nearest neighbor algorithm considering the table movement time. The placement sequence is transformed into a component type flow matrix for each group where the relation between two component types are tried to be shown. QAP greedy algorithm is used for feeder sequencing considering the flows between component types. Using arbitrary insertion point or nearest neighbor algorithm and considering the cost of placement as the maximum of turret rotation, placement and feeder movement time; a placement sequence and a feeder arrangement are found. This solution is tried to be improved with 2-opt iteratively where the placement sequence is tried to be improved as the feeder arrangement is kept fixed and the feeder arrangement is improved as the placement sequence is kept fixed. If the feeder arrangement is improved, the placement sequence is again tried to be improved. Results are compared with the lower bound generated in Moyer and Gupta [17]. It is claimed that the results are close to the lower bounds and better than the commercial software results. Although the study aims to decrease the computation time, for large PCB sizes the algorithm requires 12 hours. Time difference between the initial solution and the improved one is also another aspect that points out room for improvement.

## 2.5 The link between the current study and the literature

Our study aims to improve the component placement sequence (SP6 of Crama et al. [2] hierarchy) for a specific SMT machine, which is of turret style. Some assumptions made in the studies in the literature contradict with the specific SMT machine in concern.

Table 2.1 contains the studies carried out on the machine types mostly related to our study. These machine types are turret style, sequential pick & place and dual delivery SMT machines. For turret style machines, there are three basic movement

Table 2.1. Classification of the studies according to the problem type and machine type considered

| **Authors** | **Reference** | **Problem** | **Machine Type** | **Turret** | **Table** | **Feeder** | **Assumptions/Remarks** |
|---|---|---|---|---|---|---|---|
| Chan & Mercier | [16] | SP6 | Sequential Pick & Place | – | √ | √ | Table moves slower than the feeder |
| Moyer & Gupta | [17] | SP6 | Turret Style | √ | √ | √ | Turret movement speed is constant |
| | | | | | | | Table moves slower than the feeder |
| | | | | | | | Table movement: Euclidean Distances |
| Duman & Or | [18] | SP6 | Special Type | – | √ | – | Position of the component types around the sequencer is not critical |
| Ahmadi | [19] | SP5 | Dual Delivery | √ | √ | √ | Head movement & Table movement: Constant |
| Crama | [23] | SP5 | Turret Style | √ | √ | √ | A component type's all locations will be placed to continue with another component type’s locations |
| | | | | | | | The feeder moves from left to right by one |
| Ball & Magazine | [15] | SP5 & SP6 | Sequential Pick & Place | – | √ | √ | Distance between locations to feeder locations used |
| Leipala & Nevalainen | [20] | SP5 & SP6 | Turret Style | √ | √ | √ | Turret has 2 heads |
| | | | | | | | Turret movement is constant |
| Leu et al. | [24] | SP5 & SP6 | Turret Style | √ | √ | √ | Turret has 2 heads. Turret movement is constant |
| De Souza&Lijun | [26] | SP5 & SP6 | Turret Style | √ | √ | √ | No figure for movements in the paper |
| Moyer & Gupta | [12] | SP5 & SP6 | Turret Style | √ | √ | √ | Cycle duration is not the maximum of three movements |
| Ng | [27] | SP5 & SP6 | Turret Style | √ | √ | √ | A component type's all locations will be placed to continue other locations with different component type |
| | | | | | | | Feeder movement is slower compared to other movements |
| | | | | | | | Table movements are Euclidean |
| | | | | | | | Turret movement is constant |
| Ellis | [30] | SP5 & SP6 | Turret Style | √ | √ | √ | Durations reflect the machine characteristics. |
| | | | | | | | Statistical analysis is done to estimate the duration figures. |
| | | | | | | | Due to the M/C type examined, table speed is taken as a function of the component type according to the table speed index |

mechanisms (which will be presented in detail in Chapter 3) that duration of a PCB placement depends on: turret rotation, PCB table movement and feeder movement.

Some critical assumptions regarding these movements in the literature are:

1. The feeder carrier moves slower than PCB table.
2. Table or turret movement speeds are constant.
3. The table distance between locations is taken as Euclidean instead of Tchebycheff distance.

Based on one or more of the assumptions stated above, a solution approach from the literature cannot be applied directly to real life cases. In Table 2.1, it is seen that Ellis et al.'s [30] approach is the most realistic on with respect to the machine aspects.

In our study, based on the specifications of the machine under study, we include the followings with the modeling of the situation:

- We explicitly include turret rotation time, feeder carriage time and table movement time. We do not make any assumption on the dominance of one type of concurrent event's time to the others.
- There are 12 heads on the turret, and the turret rotation time depends on the types of the components on the turret.
- The actual feeder times are based on the feeder arrangement currently applied for each family of PCBs included.
- Table movement times are calculated based on Tchebycheff distance in between the locations. We assume constant velocity for the motors.

# CHAPTER 3

# PROBLEM DEFINITION AND THE SOLUTION APPROACH PROPOSED

SMT electronics production is composed of three phases considering process requirements as it is briefly discussed in Chapter 1.

a) SMT production line: An automated assembly line consists of machines with different functional abilities that produce semi-finished goods. Figure 1.4 shows a general layout for an assembly line.

b) Visual inspection and assembly: Semi-finished goods (PCBs) are optically inspected to test whether they conform to the international standards, such as IPC-610, or not. If the PCB involves components that could not be mounted on the automated assembly line, those components are also placed at this stage. Wave soldering is used to solder the through hole components at this phase.

c) Test: In-circuit tests are performed to judge whether there exist any problems regarding the solder joints and the circuit paths or not. Functional tests are performed to check if the finished goods conform to the functional requirements or not. Non-conforming products are sent to the rework station. Software is also uploaded on the PCBs (if any required) at this step.

Some electronics companies widen their operations by adding cleaning, coating, mechanical assembly, packaging processes as well as monitoring defects.

The company where the study is carried out is one of the well-known companies in Turkish electronics industry. It was founded as a telecommunication company and therefore focuses on telecommunication sector. In recent years, it has widened its product mix to cover electronic cards for households products. The company has three main processes:

- automated assembly where the components are placed onto PCBs by using SMT, axial and radial insertion techniques.
- telecommunication PCB production that deals with the PCBs used in telecommunication products.
- electronic card production where electronic cards functioning as display, power or control units used in appliances are produced.

Automated assembly feeds both telecommunication PCB and electronics card production lines. It should be noted that production departments have their own control of visual inspection and testing processes.

## 3.1. Problem Environment

### 3.1.1. Assembly Line

Automated assembly line has the ability to produce electronic cards using axial and SMT placement techniques. There are three axial (electronic components are inserted into the PCB through the holes vertically) and one radial (electronic components are inserted into the PCB through the holes horizontally) placement assembly lines. There are three SMT assembly lines (see Figure 1.4) in the production facility. Two of these have chip shooter machines as multi-station placement machines having multiple heads working in a concurrent manner. Placement occurs as the PCB moves along a conveyor [3]. These machines are state-of-the-art technology, more traceable in functioning and faster than the other chip shooter SMT machine. Their operations are believed to be fine tuned already.

The other SMT line contains Panasonic Mv2F which is a chip shooter machine placed after screen printing where the solder paste is printed over the PCB in order to mount electronic components over it. A fine pitch machine – Panasonic MPA G1 SMT Machine – is positioned after the chip shooter machine for more sensitive electronic components. The assembly line ends with the re-flow oven that heats and cools down the PCB and the solder paste in order to mount the components onto the PCB (see Figure 1.4). The chip shooter machine is a turret style SMT machine that is produced in 1998. Its performance is inferior when compared to the multi-station placement machines. The machine can place 40,000 components/hour theoretically if one component type is placed; it produces 10,000 components/hour in the case of multiple component types. One of the reasons for low output level in the case of multiple component types is the setups for PCB manufacturing. The placement sequence is determined with the help of the "optimizer" software that Panasonic has provided and it is known that the output usually requires expert modifications to have satisfactory performance. The higher the number of PCBs of a certain type produced, the more expertise is added. The main aim of this study is to come up with a methodology yielding better results than the expert retouch.

### 3.1.2 Turret Style Placement Machine

Figure 3.1 depicts the main parts of the machine listed below:

a) Feeder carriage: The component reels are placed on feeders and these feeders are put into the carriage. The carriage has slot numbers that describes where component reels are placed. Feeder carriage moves horizontally for the component to be picked at the grip station.

b) Turret: The machine has a rotary turret on which the picking and placement events occur. The turret has 12 revolving heads. Each head has 5 types of nozzles that can grab different types of components. After the turret turns 30 degrees, the head that meets grip station picks the component to be placed 6 steps ahead and the placement event occurs at the placement station. Grip and placement stations are fixed.

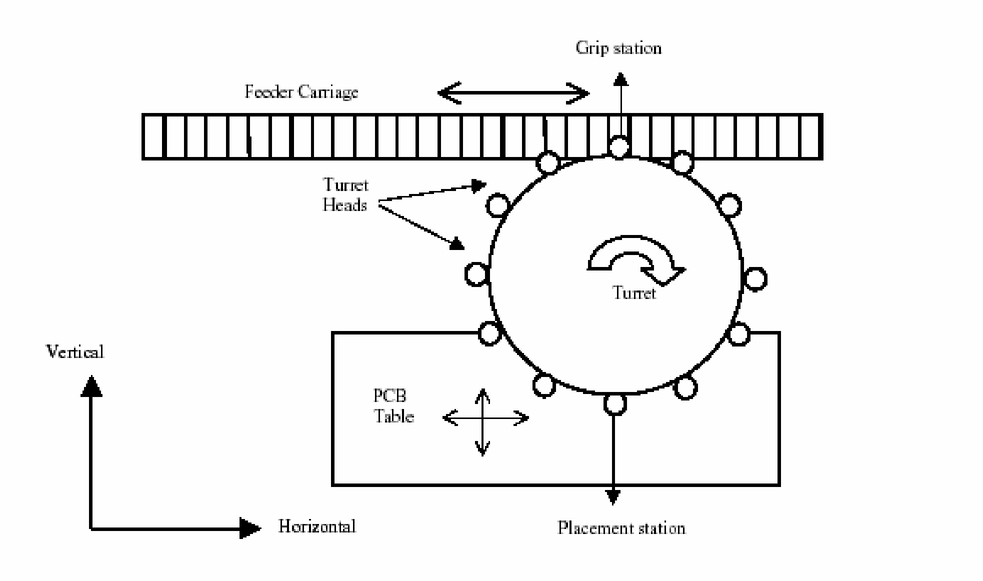


Figure 3.1 – Schematic representation of the turret style machine

c) PCB table: PCB is placed over this table. It moves in X and Y directions to bring the PCB under the placement station. Each axis uses its own motor to generate the movement, therefore the time it takes the table to meet the placement station is the maximum of X and Y movements' time, the $l_\infty$ norm known as Tchebycheff's distance metrics.

These three elements' movements depend on each other, in other words they move concurrently. Table A.1 in Appendix A, is an updated version of the one that Ng [27] undertakes. As it can be seen in Table A.1, there are three concurrent events that affect the time it takes to complete a step:

**Turret_rotation:** Each component type used in a PCB has different turret rotation index, meaning that every component cannot rotate at the same speed, since nozzle that grabs the component changes and the precision of the placement may be affected during the rotations. Turret's one rotation depends on the turret rotation indices of the components picked by the heads; component having the slowest turret rotation index (hence the smallest speed) determines the turret rotation time. The rotation index is divided into 8 groups where 1 means 0.10 sec/component, 2 means 0.16 sec/component, 3 means 0.21 sec/component, 4 means 0.27 sec/component, 5 means

0.33 sec/component, 6 means 0.38 sec/component, 7 means 0.44 sec/component and 8 means 0.50 sec/component.

**Feeder:** Feeder movement occurs if the previous component picked is not the same type with the component to be picked. If so, the feeder carriage has to move to the right component reel, defining the feeder cost. The feeder moves with a constant velocity.

**Table_movement:** Movement time between previous placement position to the new position is the minimum time traveled in X and Y axis. If position 2 $(x_2,y_2)$ is placed after position 1 $(x_1,y_1)$, then the table_movement for this step is defined as

$$\text{table\_movement}_{1,2} = \max\{|x_1-x_2| / v(X), |y_1-y_2| / v(Y)\},$$

where v(X) and v(Y) are velocity figures for x and y motors, respectively.

Each placement step cannot be completed until all the movements are finished, since three events occur concurrently. The placement event can occur after the table comes to the right position and new position’s component reel (6 steps ahead of the placed position) parks to be picked by the empty head as long as the turret head reaches the placement station and the empty one reaches the picking station. Therefore, the maximum time of these three concurrent events defines the duration for each placement step.

Once positioning is completed, placing requires a fixed time, as nozzle selection happens while the turret head is empty. An empty PCB is loaded from a conveyor and an assembled PCB is unloaded back to the conveyor in a fixed time.

### 3.1.3. The setup process

Working principles of the turret style SMT machine is explained in the previous section. PCB data should be transformed into the machine; the component types to be placed on the PCB form the component array data, where the turret rotation index for each type should be entered. Placement sequencing and feeder arrangement are currently obtained by the Panasonic optimizer which yields solutions that need to be improved.

The company has two families of PCBs as stated previously:

- Electronic cards produced for the household products
- Electronic cards to be used in the telecommunication industry

This grouping is formed considering the below factors:

1. The ease in production planning: Demand figures and the customers of the two families differ. Household products' PCB production resembles mass production with low diversity and higher amounts, while telecommunication PCBs are highly diversified and manufactured in small quantities.

2. Organizational differentiation is considered, since semi-finished products of each family are produced by different departments. Moreover, each family requires different processes on the assembly line.

3. Component types, PCB design parameters and functional purposes of these families are different. Household products' PCBs are sparser, while a larger number of components are placed per unit area of a PCB of the telecommunication family.

Having two families, there are two different setup figures on the production line; one setup holds the low volume/high variety PCBs' component types, while the other holds high volume/low variety PCBs' component types. There are 150 feeders assigned to each product family. From these 150 feeders, none of them carry the same component type because the PCB families are formed in such a way that all feeders have to be appointed to different component types. A mechanical engineer makes improvements over the solution provided by the optimizer software using his expert opinion without any analytical tools.

Since the components are already placed to the feeders at specific locations, the setup process does not include the feeder exchange time. It only involves changing the conveyor width used for loading and unloading operations and entering the placement sequence in the assembly language format by means of the machine's user- interface.

### 3.2. Problem Statement

We have already presented a detailed analysis of the turret style SMT machine, as well as the managerial decisions regarding the production planning. The opportunity to increase the productivity of turret style machines comes from the inferiority of the performance of the Panasonic software. Secondly, the literature reports that even application of the most well-known heuristics (e.g. nearest neighbor, greedy algorithms) may provide better results compared to machine specific optimizer software [27], [30]. Thirdly, the rough competition in the electronics sector demands to increase the throughputs, as we observe that the amount of backlogs in the firm under study is considerable.

The feeder assignment subproblem, (SP5) as stated by Crama et al. [2], is not the concern of this study while our main focus is to develop a modelling framework for the placement sequencing (SP6) problem for all PCB instances in each product family.

### 3.3. Model Formulation

We follow an evaluationary approach for the mathematical programming formulation of the problem: conceptual→quadratic→ integer.

### 3.3.1 Combinatorial Formulation

Let us start from a conceptual formulation after introducing the notation given below. For simplicity, we assume that the turret rotational velocity (hence time) is constant and each component is already assigned to a single feeder.

$L$ : the set of locations on the PCB $L = \{1, 2, ..., n\}$

$K$ : the set of components on the PCB $K = \{1, 2, ..., m\}$

$mt_{l_1 l_2}$ : Table movement time from location $l_1$ to location $l_2$

Let $(x_1, y_1)$ and $(x_2, y_2)$ be the $x$ and $y$ coordinates of locations $l_1$ and $l_2$, respectively. Depending on the type of the machine, one of the following (Manhattan, Euclidian, Tchebycheff’s) distances are used:

$$mt_{l_1, l_2} = \begin{cases} \dfrac{|x_1 - x_2|}{v_x} + \dfrac{|y_1 - y_2|}{v_y} & \text{, if machine has a single motor to move} \\ & \text{the table, (Manhattan distance)} \\ \sqrt{\left(\dfrac{|x_1 - x_2|}{v_x}\right)^2 + \left(\dfrac{|y_1 - y_2|}{v_y}\right)^2} & \text{, if table moves in any direction, (Euclidian distance)} \\ Max\left\{\dfrac{|x_1 - x_2|}{v}, \dfrac{|y_1 - y_2|}{v}\right\} & \text{, if there are two independent motors in} \\ & \text{x and y directions, (Tchebycheff's distance)} \end{cases}$$

where $v_x$, $v_y$, $v$ are constant velocities. We assume that acceleration and deceleration effects are negligible.

Let

$c_l$ : Component type at location $l$

$ft_{i,j}$ : feeder movement time from the feeder $i$ containing components $c_{l_1}$ to the feeder $j$ containing $c_{l_2}$ .

Since feeders are located in tandom along the horizontal axis, the movement time from feeder $i$ to feeder $j$ is simply $\frac{|x_i - x_j|}{v_f}$, where $v_f$ is the constant velocity of the feeder.

$r$ : constant time for turret rotation

Our problem is similar to a symmetric TSP, combinatorially speaking, with the following distances $d_{l_1 l_2} = \max\{mt_{l_1 l_2}, ft_{ij}, r\}$. However, while the turret in the system is making a placement on a PCB, it is being fed from the feeder. For instance, consider the turret given in Figure 3.2. While the second head holds the component for insertion on the PCB, the fifth head is being fed. In general, let $\tau$ be the turret capacity (even) in terms of the number of heads. While the $h^{th}$ head is used in placement, head $h + \tau/2$ (mod $\tau$ ) is being fed.

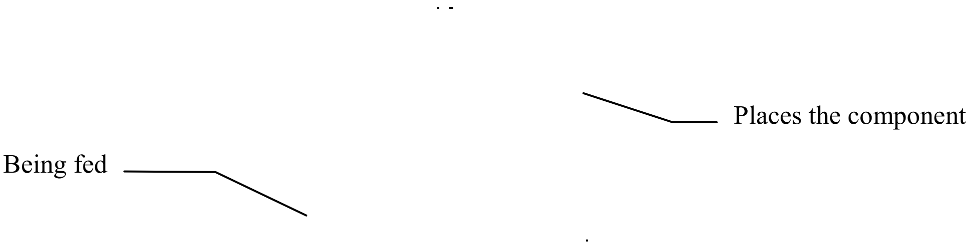


Figure 3.2 – A turret with 6 heads

Thus, assuming that we choose to move from location $l_s$ to $l_{s+1}$ at step $s$ , the correct feeder movement is for the placement at $\tau/2$ ahead, that is the feeder movement time from the feeder $i$ containing components $c_{l_s+\tau/2-1}$ to the feeder $j$ containing $c_{l_s+\tau/2}$ should be used. Therefore, distances are sequence dependent.

This conceptual formulation ignores the fact that one component type could be assigned to more than one feeder. This requires a further association of a location and a feeder, even feeder assignment is given.

### 3.3.2 Quadratic Formulation

Let the feeder assignment for components be given. Let us keep constant turret rotation time assumption for $\frac{360}{\tau}$ degrees. Let us introduce our further notation:

Sets

$F$ : set of feeders, $F = \{1, 2, ..., f\}$

$F_c$ : Set of feeders containing component $c$, $F = \bigcup_{c \in K} F_c$

Indices

$l, k$ : locations on PCB

$i, j$ : feeders

$s$: sequence counter for a pick & place operation. This index shows the step information of a picking and placement activity since a picking activity happening at step $s$ will imply a placement at step $s + \tau/2$ where $\tau$ is the number of heads on a turret. As there are $n$ locations on a PCB there are also $n$ sequences for a PCB to be produced.

Decision variables:

$T_s$: Duration of step $s$

$$x_{sl} = \begin{cases} 1, & \text{if } l^{\text{th}} \text{ location is selected in the sequence at step } s \\ 0, & \text{otherwise} \end{cases}$$

$$y_{si} = \begin{cases} 1, \text{ if feeder } i \text{ is used in the sequence at step } s \\ 0, \text{ otherwise} \end{cases}$$

Formulation:

$$\text{Min} \sum_{s} T_s \qquad (1)$$

s.t.

$$\sum_{l} \sum_{k \neq l} mt_{lk} \cdot x_{sl} \cdot x_{(s+1(\text{mod}\, n))k} \leq T_s \qquad \forall s \qquad (2)$$

$$\sum_{i} \sum_{j} ft_{ij} \cdot y_{si} \cdot y_{(s+1(\text{mod}\, n))j} \leq T_s \qquad \forall s \qquad (3)$$

$$r \leq T_s \qquad \forall s \qquad (4)$$

$$\sum_{s} x_{sl} = 1 \qquad \forall l \qquad (5)$$

$$\sum_{l} x_{sl} = 1 \qquad \forall s \qquad (6)$$

$$\sum_{i \in F_{c_l}} y_{si} \leq x_{s+\tau/2(\text{mod}\, n),l} \qquad \forall s, \forall l \qquad (7)$$

$$\sum_{i \in F} y_{si} = 1 \qquad \forall s \qquad (8)$$

$$x_{sl} = 0 \text{ or } 1 \qquad \forall l, \forall s \qquad (9)$$

$$y_{si} = 0 \text{ or } 1 \qquad \forall s, \forall i \qquad (10)$$

Objective function (1) is the sum of the costs of all the steps occurring in the PCB assembly. The objective function is aimed to be minimized. The cost of each step is determined in constraints (2), (3) and (4). As the maximum duration of three concurrent events is related to the cost associated for each step, the constraints are to incorporate the cost of placement, feeder movement and turret, respectively. If the step becomes the last location to be placed ($s = n$), the next step will be the first step of the placement sequence and this is implied by the modular notation "$\text{mod}\, n$".

Constraint (2) implies that moving from location $l$ at a step $s$, to a location $k$ in the next step $(s+1)$ generates movement time of the table between two locations ($mt_{lk}$). Constraints (3) are related to the feeder carrier movement. While the feeder carrier is at location $i$ (feeder number of the component in the carrier) at a step $s$ for picking operation, it has to move to location $j$ for the next step's ($s+1^{\text{st}}$) picking operation. Picking operation is the preparation of placement operation, i.e. the component to be picked at a step $s$ will be placed at step ($s+\tau/2$). Therefore, picking and placement mechanisms depend on each other, and this dependence is represented by constraints (7). Since a component type is assigned to more than one feeder in the model, a feeder should be selected among the possible ones only if the corresponding component is going to be placed $\tau/2$ steps ahead, as seen in (7). Constraint set (8) points out to the fact that only one feeder can be used at each step. Constraint set (4) is related to the turret movement costs.

Constraint set (5) specifies that a location on a PCB must be visited exactly once during the operations. Constraint (6) implies that at any step, only one location will be visited during the operations. These two sets of constraints eliminate subtours since we only sequence all locations and get the associated tour length using this sequence with the help of the quadratic behavior of the constraints (2) and (3).

This model involves quadratic constraints: constraint (2) and the constraint (3), where the costs related to the placement and picking sequencing are given respectively. One could linearize the model by introducing new decision variables,

$$\alpha_{slk} = x_{sl} \cdot x_{(s+1)k} = \begin{cases} 1, & \text{if the table moves from location } l \text{ to location } k \text{ at step } s \\ 0, & \text{otherwise} \end{cases}$$

$$\beta_{sij} = y_{si} \cdot y_{(s+1)j} = \begin{cases} 1, & \text{if there is a movement from feeder } i \text{ to feeder } j \text{ at step } s \\ 0, & \text{otherwise} \end{cases}$$

where,

$$\begin{aligned} &\alpha_{slk} \le x_{sl} & & \beta_{sij} \le y_{si} \\ &\alpha_{slk} \le x_{(s+1)k} & \text{and} \quad & \beta_{sij} \le y_{(s+1)j} \\ &x_{sl} + x_{(s+1)k} \le 1 + \alpha_{slk} & & y_{si} + y_{(s+1)j} \le 1 + \beta_{sij} \end{aligned}$$

### 3.3.3 Mixed Integer Formulation

We can reformulate the problem using $\alpha$ and $\beta$ decision variables.

$$\text{Min } \sum_s T_s \tag{1}$$

s.t.

$$\sum_l \sum_{k \ne l} mt_{lk} \cdot \alpha_{slk} \le T_s \qquad \forall s \tag{2'}$$

$$\sum_i \sum_j ft_{ij} \cdot \beta_{sij} \le T_s \qquad \forall s \tag{3'}$$

$$r \le T_s \qquad \forall s \tag{4}$$

$$\sum_k \sum_s \alpha_{slk} = 1 \qquad \forall l \tag{5'}$$

$$\sum_k \sum_l \alpha_{slk} = 1 \qquad \forall s \tag{6'}$$

$$\sum_l \sum_s \alpha_{slk} = 1 \qquad \forall k \tag{11}$$

$$\sum_{i \in F_{c_l}} \sum_{j \in F_{c_k}} \beta_{sij} \le \alpha_{s+\tau/2(\mathrm{mod}\, n),l,k} \qquad \forall s, \forall l, \forall k \tag{7'}$$

$$\sum_{i \in F_{c_l}} \sum_{j \in F_{c_k}} \beta_{sij} = 1 \qquad \forall s \tag{8'}$$

$$\sum_{k=1}^{n} \alpha_{s+1(\mathrm{mod}\, n),l,k} = \sum_{k=1}^{n} \alpha_{s,k,l} \qquad \forall s,\ \forall l \tag{12}$$

$$\alpha_{slk} = 0 \text{ or } 1 \qquad \forall l, \forall k, \forall s \tag{9'}$$

$$\beta_{sij} = 0 \text{ or } 1 \qquad \forall s, \forall i, \forall j \tag{10'}$$

After linearization, constraint set (5’) implies that there is only one location visited from any location. Since there is only one previous location after which any location is visited, constraint set (11) is added to reflect this property. Another constraint set (12) is also added to reflect continuity of the tour. This set implies that, if we visit location $l$ at a step $s$, in the next step ($s+1(\bmod n)$) we should move from location $l$. Constraint sets (5’), (6’), (11) and (12) will eliminate subtours, since (5’), (6’) and (11) enable both sides of the equation to 1, i.e., there is one movement at each step; while (12) ensures the continuity of the tour as it forces the assignment of ending point of one step to the starting point of the next step.

Even the formulation with linear constraints is not realistic, since the turret rotation time depends on the components carried on the turret. Each component type has a range of rotation speed of the turret. At any step in the placement process, the turret velocity should be set to the minimum (maximum) of the maximum (minimum) speed (time) of the different component types carried. To reflect this property on the formulation, parameter $r_{c_l}$ is used as turret rotation time for a component type $c_l$ to be placed on location $l$, and define $z_{c_l s}$ as a decision variable, whether a component $c_l$ is active (on the turret) at step $s$. We modified the constraint set as below.

$$r_{c_l} \cdot z_{c_l s} \le T_s \qquad \forall s, \forall l \qquad (4')$$

$$\sum_{l} \sum_{t=s}^{s+\tau/2-1(\bmod n)} \alpha_{tlk} \le (\tau/2) \cdot z_{c_k s} \qquad \forall s, \forall k \qquad (13)$$

$$z_{c_l s} = 0 \text{ or } 1 \qquad \forall l, \forall s \qquad (14)$$

Relaxing the constant turret movement assumption made in the quadratic model, $z_{c_l s}$ is introduced as a nonnegative decision variable as explained in constraint (14), and the turret movement time at a step $s$ enters the model ($r_{c_l} \cdot z_{c_l s}$) by constraint (4’). Selection of the slowest turret rotation at step $s$ is by means of the placement sequence, where of $\tau/2$ turret heads after the picking event are filled with components and the maximum rotation time of the component in this set

($s, s+1, ..., s+\tau/2-1$) determines the turret rotation time. Therefore, constraint set (13) indicates the dependence of turret rotation and the component placement sequencing. The active component that effects the turret rotation time is one of the components to be placed in this set (that would be placed in $\tau/2$ steps after step $s$). Since all of the following $\tau/2$ steps after a step $s$ might be selected from the locations that have same component type, right hand side of the constraint set is written to reflect this possibility by putting $\tau/2 \cdot z_{c_l s}$.

The quadratic model has $n$ continuous and $n(f+n)$ binary variables with $n(6+n)$ constraints. In the mixed integer formulation with linear constraints, there are $n$ continuous and $n(f^2+n^2)$ binary variables with $n(7+n+n^2)$ constraints. After removing the assumption that turret moves with a constant velocity, the model has $n$ continuous and $n(n^2+f^2+n)$ binary variables with $n(6+m+2n+n^2)$ constraints.

### 3.4. Model Validation

The formulation in the previous section is adapted to a number of PCBs instances in order to test whether the model generates optimal solution with no subtours, check if the optimal is found regarding the given constraints. Starting from small PCBs and by reducing the number of heads in the turret; four real-life PCB instances are run by CPLEX. In addition to these real PCBs, randomly generated 120 PCBs are solved optimally. Detailed analyses on the results are further discussed in Chapter 4.

In order to show the verification our model formulation, a small PCB produced in the company, will be examined in detail. The PCB has 12 locations and 3 component types, i.e., $n=12$ and $m=3$. For simplicity, each component type is assigned to one feeder and heads in the turret are taken as four[1], i.e. $f=m=3$ and $\tau=4$.

---

[1] The problem formulation for turret size of 6 is intractible, CPLEX continues to run after two days of computation.

Coordinates of the locations, component types of the locations and duration of turret rotation for the component types are given in Table B.1; distances between feeders that the components are carried are given in Table B.2, table movement distances are given in Table B.3 in Appendix B.

The results (see Figure B.1 in Appendix B for the CPLEX output) are summarized in Table 3.1.

Table 3.1. Costs realized in the optimal solution

| Step ($s$) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Table | $mt_{8,10}$ | $mt_{10,9}$ | $mt_{9,7}$ | $mt_{7,2}$ | $mt_{2,11}$ | $mt_{11,5}$ | $mt_{5,3}$ | $mt_{3,4}$ | $mt_{4,6}$ | $mt_{6,1}$ | $mt_{1,12}$ | $mt_{12,8}$ |
|  | 136 | 68 | 134 | 362 | 322 | 463 | 136 | 68 | 136 | 361 | 322 | 463 |
| Feeder | $ft_{2,2}$ | $ft_{2,1}$ | $ft_{1,3}$ | $ft_{3,2}$ | $ft_{2,2}$ | $ft_{2,2}$ | $ft_{2,2}$ | $ft_{2,1}$ | $ft_{1,3}$ | $ft_{3,2}$ | $ft_{2,2}$ | $ft_{2,2}$ |
|  | 0 | 250 | 150 | 100 | 0 | 0 | 0 | 250 | 150 | 100 | 0 | 0 |
| Turret | $r_2$ | $r_1$, $r_2$, $r_3$ | $r_1$, $r_2$, $r_3$ | $r_1$, $r_3$ | $r_1$, $r_2$, $r_3$ | $r_1$, $r_2$, $r_3$ | $r_2$ | $r_2$ | $r_1$, $r_2$, $r_3$ | $r_1$, $r_3$ | $r_1$, $r_2$, $r_3$ | $r_2$ |
|  | 500 | 271, 500, 157 | 271, 500, 157 | 271, 157 | 271, 500, 157 | 271, 500, 157 | 500 | 500 | 271, 500, 157 | 271, 157 | 271, 500, 157 | 500 |
| Cost ($T_s$) | 500 | 500 | 500 | 362 | 500 | 500 | 500 | 500 | 500 | 361 | 500 | 500 |

Durations for each concurrent event are shown in Table 3.1. For example at step 1: $\alpha_{1,8,10}$=1, $\beta_{1,2,2}$=1, $z_{2,1}$=1. Feeder movement duration occurring at a step $s$ is formed by the table movement realized at step ($s+2$). For step 1, the feeder carrier should stay at component type 2. The table movement at step 3 also shows that the table will move from location 9 to location 7 whose component types are the same, component type 2. Decision variable regarding the turret movement shows that the present step's component type and the next $(\tau/2-1)$ steps' component types that are determined by the table movement variables is taken into account. For step 1, since table movements at steps 1 and 2 indicate that the turret will place a component of type 2, the turret rotation duration for component type 2 is the associated cost.

The optimal solution indicates that the number of active decision variables related to the turret can be more than the number of loaded heads ($\tau / 2$). This is realized as a result of the model formulation since the model tries to limit the turret rotation cost occurring at a step (see constraint set (4’)) as the dependency of the turret movement and the table movement has to be considered (see constraint set (13)). Constraint set (13) does not limit the number of active components on the turret; i.e. if the table movement between two locations is realized, the model will assign the decision variable related to the turret to the component type of the ending location. The reverse is not true for the model, i.e., even in the case that the table movements are not related to a component type the turret decision variable related to the component type can be assigned to 1. Addition of another constraint that would limit the number of active components on the turret ($\sum_c z_{cs} \leq \tau / 2$) does not guarantee that all active decision variables related to the turret will reflect the active table movement decision variables in the following ($\tau / 2$-1) steps. As the model will minimize the turret related costs, active decision variables related to the turret movement will be selected from the ones that are lower than the overall cost associated at a step. Hence, this does not affect the optimum solution.

Duration of a step $(T_s)$ is the maximum of the concurrent events realized. For step 1, duration is maximum of 136, 0 and 500 which is 500. Tour obtained as the optimal solution does not contain any subtours and the optimal solution is 5723 msec, which is solved in 3.5 hours on a PC with P4 1.5 GHz processor with 256 MB RAM.

The model is a mixed integer programming and the solution is found by using CPLEX’s standard branch and bound (B&B) technique. B&B tree sizes as well as the solution times are also related to the concurrent events; if one dominates the others (i.e., for example if the table movement durations are more than the others), smaller solution times are observed in 120 randomly generated problems. The details will be explained in Chapter 4.

## 3.5. Solution Approach

After finding out that the optimal solutions to PCBs that have 10 components and 4 locations may be found in 19 hours (69646 seconds), a PCB that has 19 locations and 5 components is investigated. Having 12 heads on the turret, the model is run on the CPLEX, the software could solve 37% of the B&B tree in 14 days and it is estimated that finding the optimal solution would be realized at most in 40 days.

Having discovered that an optimal solution to our formulation is difficult to be found in a reasonable time for a real instance, a new approach has to be developed to find a component placement sequence for the rotary turret machine. The concurrency in the machine complicates the problem as well as it becomes intractable even in small instances. Therefore, two upper-bounding algorithms ignoring some of the concurrent events and using the classical TSP heuristics are developed in this study.

### 3.5.1. Heuristic 1

The main characteristics are listed below:

1. The overall PCB component insertion problem is divided into smaller sized problems such that a symmetric TSP considering only the table movement times is solved for each component type and solutions are conquered. This strategy is used by some of the researchers (such as Ng [25], Ellis et al. [30]). When only one component type is to be placed, the feeder carriage does not move while the turret continues to rotate. The feeder duration is zero and the total time of each step is the maximum of table movement time and turret rotation time, which is constant for a component group.

2. An optimal TSP solution is found by using CONCORDE [31] for each component type.

3. In order to construct a complete tour, we combine the subtours obtained for component types one by one at each iteration. Maximum of the movement times between pairs of locations from two subtours and the associated feeder movement times are taken for all pairs of locations of different component types. Among all possible ways of combining subtours, the one with the minimum duration is selected and the corresponding pairs of subtours are combined at the associated location pairs. This step continues until we construct a solution, one tour covering all the locations. The length of the constructed tour differs according to the direction, as the placement sequence of the locations (on which the turret movement costs and feeder carrier costs depend) also changes when the direction of the constructed tour changes. The tour is evaluated for selecting the best direction, clockwise or counter-clockwise. Another reason for tour length calculations for both directions is the usage of clockwise and counter-clockwise costs (regarding the turret rotation and feeder carrier movement costs) for a specific location at improvement phase calculations.

4. As an improvement heuristics, we modified the classical 2-opt. By breaking two arcs between locations and connecting the locations in the reverse direction, possible gains are calculated and the one with the maximum gain is applied. In traditional 2-opt, at any iteration, gain can be calculated by finding the difference between two connections that are cut and two connections that are newly formed. Because of the machine characteristics, the direction change means a change in the placement sequence, which also implies that tour length will change, since the turret rotation costs and the feeder carriage costs are sequence dependent. The modified 2-opt, therefore, considers the changes in the distance matrix regarding the directional changes in the tour as the cost at a step is taken as the maximum duration of three movements performed by the SMT machine. This greedy step continues until no improvement is obtained.

The details of the first algorithm are given in Appendix C. An example PCB is used to illustrate the developed algorithm. This PCB has 32 locations with 7 types of components. The PCB layout is depicted in scale in Figure 3.3. Table 3.2 contains the component types used in the PCB and machine specifications, which are feeder positions of each component type on the SMT machine and the turret rotation index for component types respectively.

Table 3.2. Component Data provided from the SMT machine

| Component No | Component Name | Feeder Position Assigned | Turret Rotation Index | |
|---|---|---|---|---|
| | | | Minimum Speed | Maximum Speed |
| 1 | HDRN00160-449 | 174 | 8 | 2 |
| 2 | HDRN00162 | 10 | 8 | 4 |
| 3 | HDRN00169 | 9 | 8 | 4 |
| 4 | HDRN00180 | 74 | 8 | 4 |
| 5 | HDRN00291 | 212 | 8 | 4 |
| 6 | HKPS00089 | 6 | 8 | 4 |
| 7 | HYRL00146 | 2 | 8 | 6 |

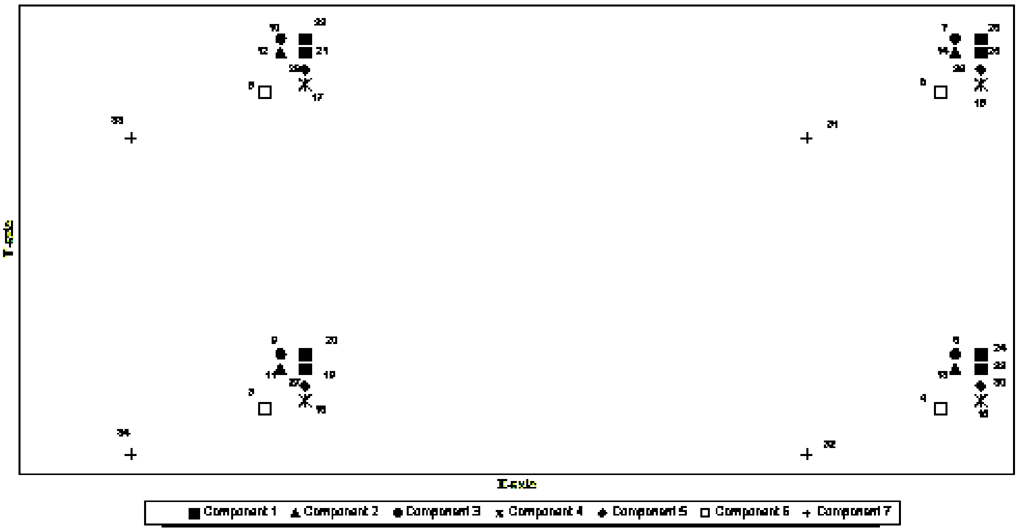


Figure 3.3 – PCB Layout of the example problem

The table moves in 6.7 mm/sec. If a feeder is located at position $i$ in a feeder carrier, it will take 0.67 seconds to reach to the next feeder located at position $i+1$. Turret heads' maximum movement velocity on the components are listed in Table 3.2. For instance, a component that has turret rotation index of 8 can rotate one turn in 0.5 seconds, which means that this component will rotate 2 turns in one second.

Table 3.3. Movement characteristics of the turret style SMT machine

| Turret rotation Index | Max turret rotation speed (turn/sec) allowed |
|---|---|
| 1 | 10.000 |
| 2 | 6.365 |
| 3 | 4.668 |
| 4 | 3.686 |
| 5 | 3.045 |
| 6 | 2.594 |
| 7 | 2.259 |
| 8 | 2.000 |

STEP 1: SUBTOUR GENERATION FOR EACH COMPONENT TYPE

For each component type, CONCORDE is run to obtain the associated subtour. In Figure 3.4, these subtours are depicted for all component types. Table 3.4 shows the subtour information for component types.

Table 3.4. Subtour Information for component types

| Component ID | Subtour Strucuture | Length (msec) |
|---|---|---|
| 1 | 19→20→21→22→26→25→24→23→19 | 24316 |
| 2 | 11→13→14→12→11 | 23978 |
| 3 | 7→8→9→10→7 | 23978 |
| 4 | 15→16→17→18→15 | 23978 |
| 5 | 27→28→29→30→27 | 23978 |
| 6 | 3→5→6→4→3 | 23976 |
| 7 | 31→32→34→33→31 | 23976 |

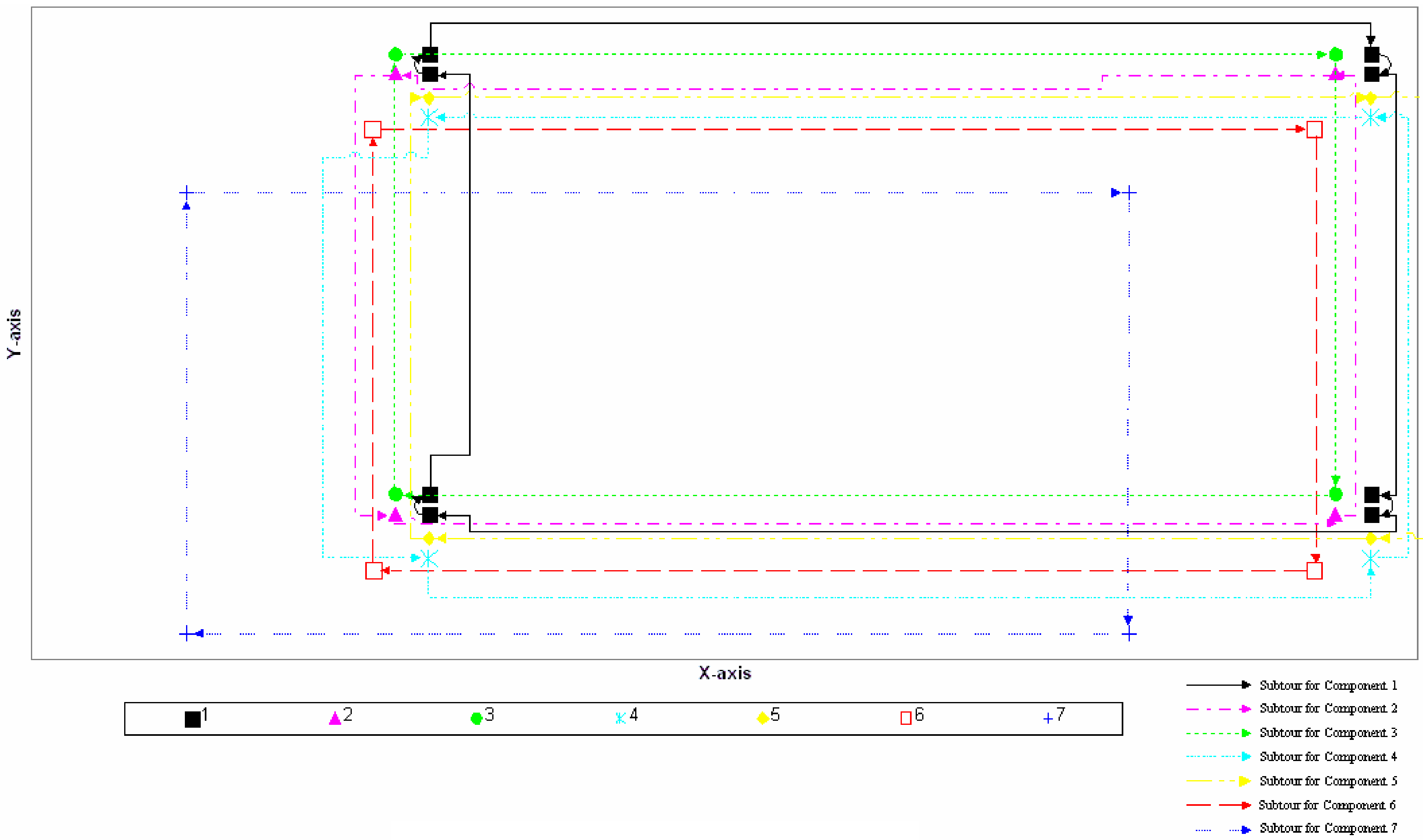


Figure 3.4 – Subtours over PCB layout

STEP 2: COMBINING SUBTOURS TO GET A COMPLETE TOUR

Subtours obtained for each component should be joined to form the assembly sequence for a PCB. In order to join two components' subtours, arcs from both components' subtours should be cut and new tours have to be formed by joining the locations whose links are cut. The best cut is found after enumerating all such pairs and selecting minimum time (maximum of the gains) is found to build the new tour cycle.

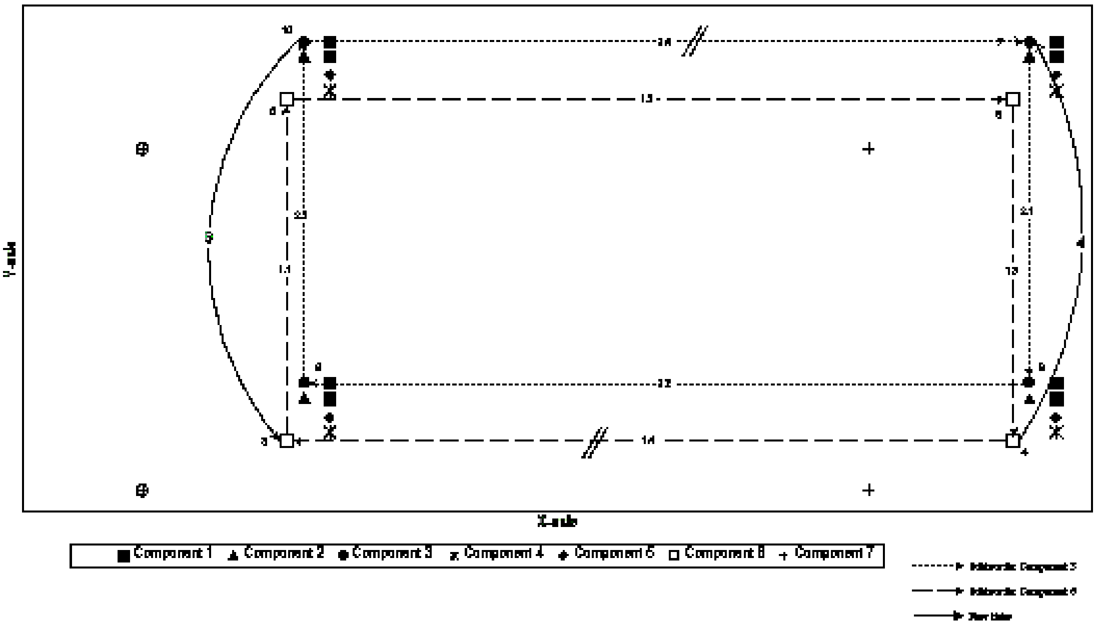


Figure 3.5 – Combining associated subtours of two component types

Two components, HKPS00089 and HDRN00169, are taken as an example. The links are numbered with the component index, as 1.1 represents the first movement occuring in component 1. Assume that we have selected 2.4 and 1.4, the new cycle will be formed with the addition of connections 4 and 8 as illustrated in Figure 3.5; the gain value for a pair of links is obtained as follows:

$$Move\_Gain(HKPS00089, HDRN00169)$$

$$= \text{Length of the new arcs} - \text{Length of the old arcs}$$

$$= Length\ (4,7)+Length(3,10)-Length(4,3)-Length(10,7)$$

Let CONN1(i, j) be the function used to represent the first location to be cut, CONN2(i, j) shows the second one. CONN1(1, 3) = 4 while CONN2(1, 3) = 7. Table 3.5 shows the results of iterations to find the maximum gain between these two components.

Table 3.5. PCB Table movement gains by combining components (msec)

| **Component 1** | **Component 3 Table Movement** | | | | **Component 3 Feeder movement** | | | |
|---|---|---|---|---|---|---|---|---|
| | **7** | **8** | **9** | **10** | **7** | **8** | **9** | **10** |
| **3** | -1687 | -10357 | -374 | -374 | 4500 | 4500 | 4500 | 4500 |
| **4** | 6976 | -1694 | -1312 | -1313 | 4500 | 4500 | 4500 | 4500 |
| **5** | -1314 | -382 | 9601 | -1 | 4500 | 4500 | 4500 | 4500 |
| **6** | -1313 | -381 | 0 | -9601 | 4500 | 4500 | 4500 | 4500 |

In calculating the best gain values between locations, it should be noted that combining the tours also implies a change in feeder location, i.e. feeder carrier will move to feed the next placement. This movement should also be taken into account as two components' tours are combined. Feeder movement between component 1 and component 3 is calculated as:

$$Feeder(i, j) = \frac{|fp(i) - fp(j)|}{v_f} \times 1000,$$

where $fp(i)$ : feeder position of component $i$

$v_f$ : feeder velocity (racks/sec).

Therefore,

$$Feeder(component1, component3) = \frac{|6-9|}{0.67} \times 1000 = 4500\ m\sec$$

It can be seen in Table 3.5 that the maximum gain is obtained by joining location 5 and location 9 with a gain of 9601 msec, since the feeder movement will imply 4500 msec of cost joining the components from these locations will end up with a better subtour. Therefore, 1.3 and 2.2 are cut while locations 8-4 and 5-9 are joined, in the end the cycle is formed as: 5→9→10→7→8→4→3→6→5.

The feeder time should also be included in the gain calculations, this will be done by taking the feeder time as a cost factor into the function such that:

$$Net\_Gain(i,j)=\begin{cases} move\_gain(i,j) & \text{, if } move\_gain(i,j) < Feeder(i,j), \\ -Feeder(i,j) & \text{, otherwise.} \end{cases}$$

Net_gain is obtained after calculation of locational gains between all pairs of component types, i.e. comparison of gain results with the feeder costs for each pair will give the combined gain figure for each pairs of component types. Table 3.6 shows the gain values between each component types, the highest gain value will be selected as the components having the highest gain will be joined.

Table 3.6. Net_Gain calculations

| | **2** | **3** | **4** | **5** | **6** | **7** |
|---|---|---|---|---|---|---|
| **1** | 9601 | 9262 | -102000 | -252000 | -309000 | 6545 |
| **2** | | 9008 | -97500 | -247500 | -304500 | -10500 |
| **3** | | | -96000 | -246000 | -303000 | -12000 |
| **4** | | | | -150000 | -207000 | -108000 |
| **5** | | | | | -57000 | -258000 |
| **6** | | | | | | -315000 |

Continuing in this manner, we combine subtour 1 & 2, then 3 is added to them, afterwards 3 and 7 are combined. In the next iteration, 5 and 6 form a new subtour, where 4 is connected to this subtour. Subtours formed by components 1-2-3-7 and 4-5-6 are joined by the locations from 3 and 4. The full constructed tour given in Figure 3.6 is obtained finally.

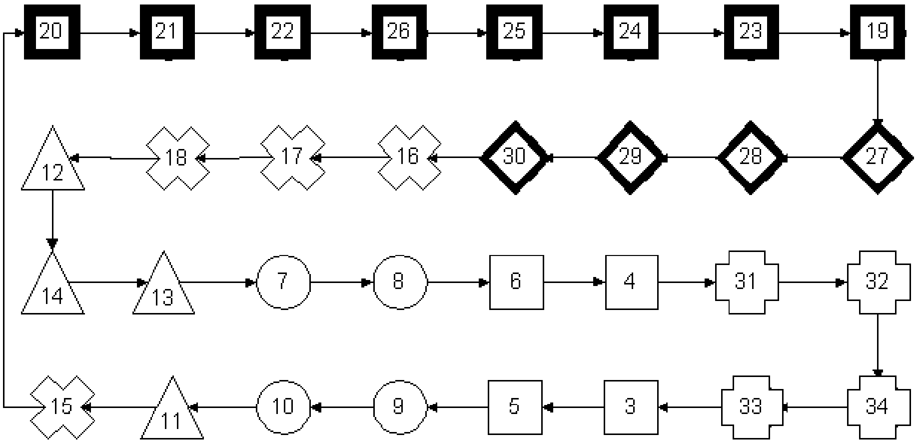


Figure 3.6 – Constructed tour

The concurrent events taking place under the tour given in Figure 3.5 are listed in Table D.1 in Appendix D.

STEP 3: IMPROVEMENT WITH 2-OPT

The modified 2-opt application is shown in Figure 3.7 where two connections are cut and a new tour structure is formed.

After cutting the connection between locations 13 – 7 and 11 – 15, the new tour is formed by combining locations 13 – 11 and 7 – 15 as the tour is revised by changing the directions of the connections between locations 11 to 7. In this example, since the tour is clockwise directed, the direction of the tour between locations 11 and 7 becomes naturally counter-clockwise.

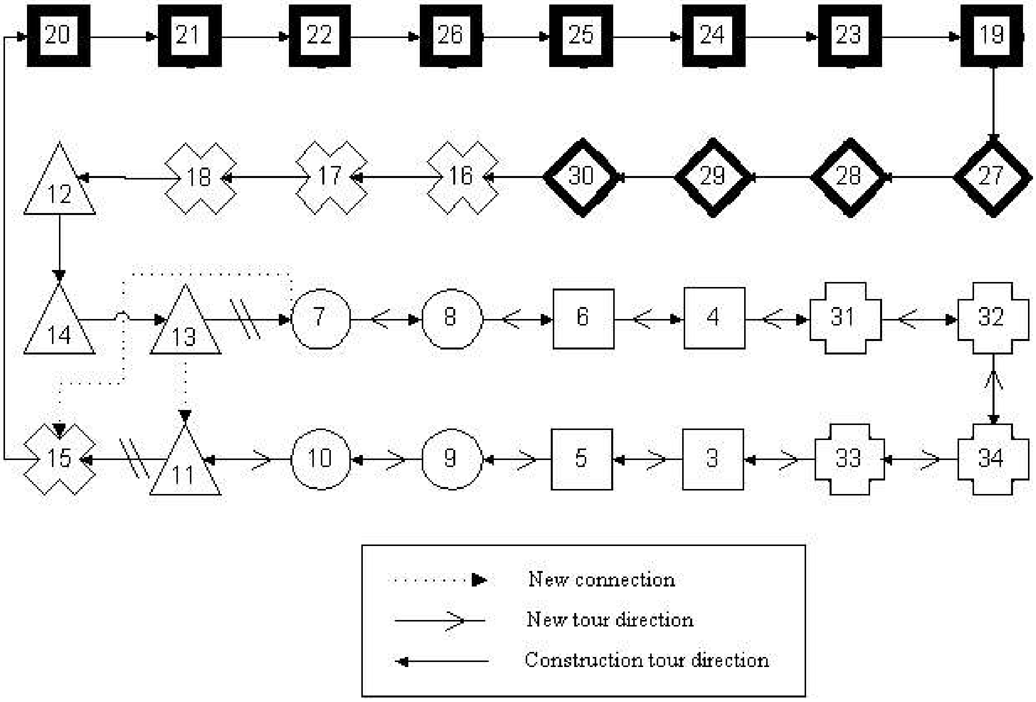


Figure 3.7 – Application of 2-opt to the constructed tour

Improvement, using 2-opt requires GAIN calculations. As the tour lengths for clockwise and counter-clockwise directions are calculated prior to 2-opt, the tour was broken into stepwise costs where the tour length is found out after adding the cost figure affecting each step.

The difference between the new and the old tour obtained by exchanging $i \& j$ can be divided into three parts:

1. Although the tour direction does not change from the location $right(j)$ to $i$, the change in tour structure after location $i$ points out that, calculations regarding the locations that are 6 steps prior to location $i$ should be made since turret movement and feeder movement costs are different than the old tour.

2. Tour structure between locations $left(j)$ and $right(i)$ becomes counter-clockwise direction where one can use counter-clockwise calculations performed previously with a difference that the new tour structure after location $right(i)$ requires new calculations for the locations $right(i)$ and the locations 6 steps prior to $right(i)$. This requirement is again caused by the changes in feeder and turret movements.

3. Calculations to be done for right(i) and 6 steps prior to this location, is similar to table movement time changes for part 2.

The details of GAIN calculations are provided in Appendix E.

After GAIN calculations for all possible pairs (except the neighbors) are made, the arc with the maximum GAIN is selected if it improves the tour. In our algorithm that uses 2-opt for improvement phase, starting from a given point and cutting all connections except neighbor connections next to the given point, all pairs are enumerated as GAIN results are recorded to find the best tour. The best GAIN figure is used to change the tour structure, after changing the tour structure 2-opt iterations are applied to the new tour again until no better improvement is achieved. For the example shown in Figure 3.2, Table 3.7 shows the summary of results of our 2-opt improvement step.

Table 3.7 Results obtained after 2-opt iterations for Heuristic 1

| Iteration | Tour | Tour length (msec) |
|---|---|---|
| 0 | 20→21→22→26→25→24→23→19→27→28→29→30→16→17→18→12→14→13→7→8→6→4→31→32→→34→33→3→5→9→10→11→15→20 | 742074 |
| 1 | 20→19→27→28→29→30→16→17→18→12→14→13→7→8→6→4→31→32→34→33→3→5→9→10→11→15→23→24→25→26→22→21→20 | 718976 |
| 2 | 20→21→22→26→25→24→16→17→18→12→14→13→7→8→6→4→31→32→34→33→3→5→9→10→11→15→23→30→29→28→27→19→20 | 708679 |
| 3 | 20→21→22→26→25→24→16→17→18→12→14→7→13→8→6→4→31→32→34→33→3→5→9→10→11→15→23→30→29→28→27→19→20 | 701797 |
| 4 | 20→21→22→26→25→24→16→17→18→12→14→7→13→8→6→4→31→32→34→33→3→5→10→9→11→15→23→30→29→28→27→19→20 | 695557 |
| 5 | 20→21→22→26→25→24→16→17→18→12→14→7→6→8→13→4→31→32→34→33→3→5→10→9→11→15→23→30→29→28→27→19→20 | 694277 |
| 6 | 20→21→22→26→25→24→16→17→18→12→14→7→6→13→8→4→31→32→34→33→3→5→10→9→11→15→23→30→29→28→27→19→20 | 693122 |

### 3.5.2. Heuristic 2

The second algorithm, instead of decomposing the locations according to the component types, treats the placement sequencing as a whole. The distance matrix for the PCBs is developed such that the asymmetry in rotation times is resolved by taking the maximum:

$$d_{l,k} = \max\{mt_{lk}, \max\{r_{c_l}, r_{c_k}\}\}$$

The distance matrix obtained is a symmetric TSP and it is run by CONCORDE, the solution serves as the constructed solution. We apply the same improvement procedure that of the first heuristic once we set a direction (clockwise/counter-clockwise).

## STEP 1: OBTAINING A CONSTRUCTION TOUR

For the example PCB used in the first heuristic, distance matrix is formed which is sent to the CONCORDE software. Duration of the tour obtained is calculated in clockwise and counter-clockwise direction. Minimum of the durations is selected as the constructed solution. If the tour selected is directed counter-clockwise modification of the tour is done.

After tour is obtained from CONCORDE output, cycle duration with clockwise and counter-clockwise is found as 2821703 miliseconds and 2826832 miliseconds, respectively. Since duration of clockwise direction is lower construction tour has clockwise direction.

## STEP 2: IMPROVEMENT BY MODIFIED 2-OPT

The constructed solution has to be improved since feeder distances are not involved in the distance matrix and concurrency is not fully covered yet. The constructed solution takes the movements between two locations into account, and it neglects the "$\tau/2$ - 2" turret movements which should be the maximum duration of the component types to be placed within $\tau/2$ steps. The feeder movement should also be evaluated in the solution procedure as a duration that would affect the duration of a step by taking the component change which will be the table movement happen after $\tau/2$ steps. The output and tour information for the constructed solution provided by the software is seen in Table 3.8 as the "iteration 0" row. During the 2-opt iterations, the directions of the tour also change as described in "Step 3" of the heuristic 1.

Table 3.8 Results obtained after 2-opt iterations for Heuristic 2

| Iteration | Tour | Tour length (msec) |
|---|---|---|
| 0 | 1→2→15→9→25→32→31→27→10→16→4→3→22→19→29→28→12→13→8→7→23→18→6→14→11→5→24→17→26→30→20→21→1 | 2821703 |
| 1 | 1→2→15→9→25→32→31→27→10→16→4→3→22→19→29→28→12→13→11→14→6→18→23→7→8→5→24→17→26→30→20→21→1 | 2324585 |
| 2 | 1→2→15→9→25→32→31→27→10→16→14→11→13→12→28→29→19→22→3→4→6→18→23→7→8→5→24→17→26→30→20→21→1 | 1820082 |
| 3 | 1→2→15→9→25→32→31→27→10→16→14→11→13→12→28→29→19→20→30→26→17→24→5→8→7→23→18→6→4→3→22→21→1 | 1421460 |
| 4 | 1→2→15→9→25→32→31→27→10→16→14→11→13→12→28→29→19→20→30→26→17→18→23→7→8→5→24→6→4→3→22→21→1 | 1110055 |
| 5 | 1→21→22→3→4→6→24→5→8→7→23→18→17→26→30→15→9→25→32→31→27→10→16→14→11→13→12→28→29→19→20→2→1 | 912271 |
| 6 | 1→21→22→3→4→6→24→23→7→8→5→18→17→26→30→15→9→25→32→31→27→10→16→14→11→13→12→28→29→19→20→2→1 | 806483 |
| 7 | 1→6→4→3→22→21→24→23→7→8→5→18→17→26→30→15→9→25→32→31→27→10→16→14→11→13→12→28→29→19→20→2→1 | 721168 |
| 8 | 1→6→4→3→22→21→24→23→7→8→5→18→17→26→30→29→28→12→13→11→14→16→10→27→31→32→25→9→15→19→20→2→1 | 707539 |
| 9 | 1→6→4→3→22→21→24→23→7→8→5→18→17→11→13→12→28→29→30→26→14→16→10→27→31→32→25→9→15→19→20→2→1 | 699056 |
| 10 | 1→6→4→3→22→21→24→23→7→8→5→18→17→11→13→12→28→29→30→26→14→15→9→25→32→31→27→10→16→19→20→2→1 | 692507 |
| 11 | 1→6→4→3→22→21→24→23→7→8→5→18→17→11→12→13→28→29→30→26→14→15→9→25→32→31→27→10→16→19→20→2→1 | 689778 |
| 12 | 1→6→4→3→22→21→24→23→7→8→5→18→17→11→12→13→28→29→30→26→14→15→9→25→32→31→27→16→10→19→20→2→1 | 688279 |
| 13 | 1→6→4→3→22→21→24→23→7→8→5→18→17→11→12→13→28→29→30→32→25→9→15→14→26→31→27→16→10→19→20→2→1 | 685753 |
| 14 | 1→6→4→3→22→21→24→23→7→8→5→18→17→11→12→13→28→29→30→32→25→15→9→14→26→31→27→16→10→19→20→2→1 | 684253 |
| 15 | 1→2→6→4→3→22→21→24→23→7→8→5→18→17→11→12→13→28→29→30→32→25→15→9→14→26→31→27→16→10→19→20→1 | 683915 |
| 16 | 1→2→6→4→3→22→24→21→23→7→8→5→18→17→11→12→13→28→29→30→32→25→15→9→14→26→31→27→16→10→19→20→1 | 671292 |

## 3.6. Lower Bounds

In order to measure the performance of our heuristics over real life problems in the absence of the optimal solutions, we need to find a sound basis for comparison purposes. In this section, we report the development two lower bounding schemes, which are used in the comparative study given in Chapter 4.

### 3.6.1. Lower Bound 1

The first lower bounding approach considers table movement times and rotation times first. In this sense, it is similar to the second heuristic. The optimum solution for the case in which we ignore one or two concurrent events yields clearly a lower bound to the overall problem. In the mixed integer formulation (see Section 3.3.3), we use constraint sets (2’) and a simpler version of (4’). Constraint set (4’) has to be modified, since a sequence that will define the turret rotation cost will also mean finding the optimal solution which is difficult to be obtained. We would like to make use of a symmetric TSP solver to find the optimum solutions. In order to use such a software, the distance matrix should be symmetric. To have a lower bound, we modify the distance matrix of the first heuristic as

$$d_{l,k} = \max\{mt_{lk}, \min\{r_{c_l}, r_{c_k}\}\}$$

by taking the minimum of the rotation times corresponding to the components to be mounted at location *l* and location *k*. Assume that $z_{TSP}^{1}$ be the optimum tour using this distance matrix. Let $z*$ be the optimum solution of our problem and let the maximum edge length on this TSP tour be $mt*$. Since we are taking the maximum of the table movement, turret rotation and feeder movement times, we have $z_{TSP}^{1} \le z^{*}$.

The extreme case in which the effect of the feeder movements is kept at the minimum is when we have a TSP tour on $n$ locations such that

1- all the locations of the same component type are consecutive (in a connected path) in the optimal tour. Thus, one feeder is used to supply all the components and it is visited only once.

2- the component type paths are concatinated in such a way that the feeder sequence is optimized with respect to feeder movements.

Therefore, we have another tour of length on $m$ feeder locations in this extreme case. Let the maximum edge length on this TSP tour be $ft*$.

Since the two TSPs are concurrent, we may safely increase the lower bound as $LB_1 \doteq z_{TSP}{}^1 + [ft^* - mt^*]^+ \le z^*$, where $[ft^* - mt^*]^+ = \max\{0, ft^* - mt^*\}$. Considering the mixed integer formulation, constraint set (3') is taken as an independent problem through which we obtain an optimal solution that minimizes the feeder carriage movements for a PCB's component types. The highest cost of a step in the optimal solution, is merged into the first problem's optimal solution.

Consider the situation given in Figure 3.8 where we have four locations {a,b,c,d} and three component types (a and b are of the same type). Let the turret rotation time is 1 for all components. Then, the edge values in the figure are the table movement times determining the distances in the first TSP. One of the alternative optimal tours in the PCB is "a → b → d → c → a" yielding a length of 2+4+5+3=14=$z_{TSP}{}^1$. The maximum edge length here is $mt* = 5$. There is only one feeder movement TSP tour "I → II → III → I" with length 2+5+7. The maximum edge length here is $ft* = 7$. Then, the lower bound is $LB_1 \doteq z_{TSP}{}^1 + [ft^* - mt^*]^+ = 14 + (7-5) = 16 \le z^*$.

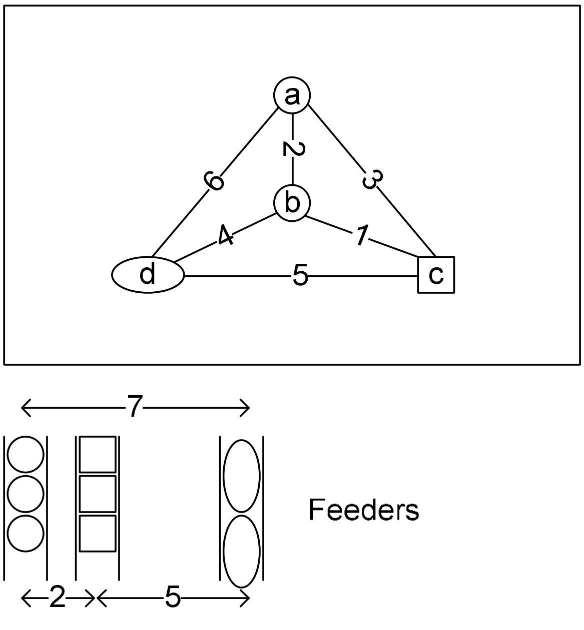


Figure 3.8 – Lower Bound 1

The real cost of the first tour "a → b → d → c → a" is max{2,0,1}[2] + max{4,7,1} + max{5,5,1} + max{3,2,1} = 2+7+5+3 = 17. Consider the tour "a → b → c → d → a" with length max{2,0,1}+max{1,2,1}+max{5,5,1}+max{6,7,1}=2+2+5+7=16. Thus, $z^* \leq 16$. Hence, $LB_1 = z^* = 16$ for this instance.

### 3.6.2. Lower Bound 2

The second lower bound focuses on feeder movements first. The smallest feeder movement tour is resulted from the same extreme case explained in the previous section. Let the optimal TSP tour length for $m$ feeder locations be $z_{TSP}{}^2$. The overall tour specifying a solution to our problem should contain $n$ edges. In order to have a sound lower bound, we can extend the $z_{TSP}{}^2$ value if we can find the minimum length path with $n-m$ edges using the distance function ($d_{l,k} = \max\{mt_{lk}, \min\{r_{c_l}, r_{c_k}\}\}$) used in $z_{TSP}{}^1$, independent of the optimal feeder TSP tour. If the minimum length of a path of size $n-m$ is $z_{path}{}^*$, then $LB_2 \doteq z_{TSP}{}^2 + z_{path}{}^* \leq z^*$.

Feeder movements' minimum tour is found by constraint set (3') in the mixed integer formulation. As the tour found has $m$ locations and $n-m$ locations are required to complete the tour, a modified approach to the problem is applied using the distance function used in Lower Bound 1.

In our example instance given in Figure 3.7, $z_{TSP}{}^2 = 2+5+7 = 14$ and since $n-m = 4-3 = 1$ we pick the one--edge path with the minimum length as "b–c" with value $z_{path}{}^* = 1$. Therefore, the value of the second lower bound in this instance is $LB_2 \doteq z_{TSP}{}^2 + z_{path}{}^* = 14 + 1 = 15 \leq z^* = 16$.

We modified Prim's algorithm [37] developed for finding the minimum spanning tree, for the purpose of finding the shortest path of size $n-m$ in a given graph $G = (V, E)$. Our procedure for a given vertex $v \in V$ starts with the path of size one

---

[2] max{table,feeder,turret}

after we locate the closest vertex $w = \arg\min\{d_{u,w} : (u,w) \in E\}$ to $v$. At iteration $k$, we have a path of size $k-1$, and augment this path by finding a vertex outside the current path with the minimum distance to one of the two endpoints of the path. This augmenting algorithm for a vertex $v \in V$ terminates at the end of iteration $n-m$ with a shortest path containing $v$ and having $n-m$ edges. If we enumerate this augmenting algorithm initiated for all such vertices, and pick the one with the shortest length, we determine $z_{path}^*$.

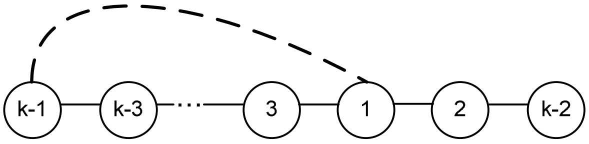


Figure 3.9 – Iteration k in the augmenting algorithm

Consider iteration $k \geq 3$ and the situation given in Figure 3.9. It is not possible to find a better path of the same size? The answer is no! The edge in the path incident to any node is either the closest edge or the second closest edge. For instance, the edge in between node 1 and 2 is the closest edge to node 1 whereas the edge between nodes 1 and 3 is the second closest edge to node 1, whose distance value is less than that of the one in between 2 and k-2. We know that node 3 is closer to node 1 than node k-1; otherwise, node k-1 is selected in the second or third iteration. Thus, $d_{31} \leq d_{(k-1)1}$. Then, $d_{31} + \cdots + d_{(k-3)(k-1)} \leq d_{(k-1)1} + \cdots + d_{(k-3)(k-1)}$ implying that the current path is better than that of cutting edge 13 and inserting edge (k-1)1. Similarly, the new path formed by cutting one of the edges in the subpath from node 3 to node k-1 and adding the edge (k-1)1 does not yield a better solution, as this subpath was formed by picking the best possible edges. Thus, it is enough to consider only the nodes outside the current path at each augmenting iteration.

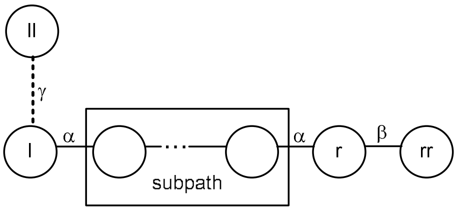


Figure 3.10 – Ties in the augmenting algorithm

What if we have a tie broken arbitrarily. Consider the situation given in Figure 3.10. Assume that a tie occurs in deciding whether we add node *r* or node *l* and we choose to add node *r*. If $\beta \leq \alpha$, we add node *rr* to the path in the next iteration; otherwise, the path is augmented through node *l*. If $\beta \geq \gamma$ and we terminate our procedure without adding node *ll*, we give up a better path ("*ll*—*l*—subpath—*r*" over "*l*—subpath—*r*—*rr*" or "*ll*—*l*—subpath" over "subpath—*r*—*rr*"). However, there is another search initiated from node 1. We may obtain these ("*ll*—*l*—subpath—*r*" or "*ll*—*l*—subpath"), if a better path does not exist, at the end of that search. Thus, we can safely break ties arbitrarily.

The detailed procedure is given in Appendix F. This procedure terminates in $n[2n(n-m)]$ operations, thus it is pretty fast.

Our algorithms given in this chapter are coded in Visual Basic scripts and as a user interface MS Excel is used. The main reason of selecting MS Office tools is to allow the users working for the company to run the program without installing any other licensed software. Recall that we use CONCORDE in the process of obtaining constructed solutions for the heuristics, which is a shareware program. Another reason is that most of the industrial people are more familiar with Excel spreadsheets.

# CHAPTER 4

# COMPUTATIONAL STUDY

A computational study is carried out to test the performance of the proposed heuristics and lower bounds. In the study, the PCBs that are currently in the product mix of the company, as well as randomly generated problem instances are considered.

The proposed algorithms are coded in Visual Basic. The code is run on a computer with Pentium M processor 2 GHz. The mathematical model provided in Chapter 3 is implemented using CPLEX. For most of the PCBs that the company produces, the optimal solutions could not been obtained within 10 days. Therefore, for these PCBs the performance of the heuristics is reported as relative to the lower bounds proposed. In order to investigate the quality of both heuristics and lower bounds against optimal values, fictitious PCBs are generated randomly. In addition, for some of the PCBs that the company produces we could obtain the optimal solutions for a simplified case where there are four (instead of twelve) heads. The results of the computational study are discussed in Sections 4.1 and 4.2 for the PCBs that are in the product mix of the company, and the randomly generated problem instances, respectively.

## 4.1. Real PCBs currently produced

The design parameters of PCBs studied are given in Appendix G and a summary on number of locations, number of different components used and annual demand is provided in Table 4.1. Note that PCBs for which there is no annual demand figure are the ones that are not produced in regular basis, but were produced for some project work in the past.

Table 4.1. Summary information on PCBs studied

| PCB Name | Number of Locations | Number of Component types | Annual Demand |
|---|---|---|---|
| FT10-2 | 38 | 15 | 1200 |
| PS03 | 32 | 7 | 5000 |
| FT20-1GA | 38 | 15 | 2000 |
| CPUMOD | 39 | 6 | 360 |
| FIRIN | 38 | 15 | -- |
| EVM224SL | 137 | 28 | 100 |
| DSS16-2 | 131 | 27 | 500 |
| CCSLAVE2 | 173 | 21 | 40 |
| 200CPU | 118 | 28 | 600 |
| DS200SBP | 231 | 5 | 150 |
| 48EX26 | 455 | 52 | 6000 |
| 128EX1 | 963 | 51 | 3500 |
| DS10BP | 384 | 50 | 5 |
| 224CPUOPT | 344 | 66 | 150 |
| MS48ABA | 1156 | 83 | 10000 |
| POWERSTAR_PNP_N | 60 | 31 | 21600 |
| ZARC00076-AAA | 66 | 22 | 85000 |
| ZGDA33AAA | 53 | 20 | 4200 |
| A0V01 | 72 | 31 | 600000 |
| HBDK00437PNP | 109 | 24 | 45000 |
| VECTOR-V03 | 132 | 32 | 50 |
| HBDK00414 | 154 | 16 | 25000 |
| HBDK00437AAC2 | 141 | 26 | 48000 |
| HBDK00446 | 172 | 34 | 850000 |
| HBDK00386-AAA-P | 186 | 41 | 15000 |
| 7241-0101-SS1 | 240 | 67 | 4 |
| PIO2-1 | 465 | 26 | 200 |
| SLIC | 326 | 47 | 2800 |
| 7550-0078-SS | 329 | 41 | 4 |
| PIO68 | 432 | 21 | 100 |

Tour length (in milliseconds) is our primary performance measure. Table 4.2 includes the lower bounds on the tour lengths, and the tour lengths found by the proposed two heuristics. The percentage deviation of the solution generated by each heuristic from the best lower bound (the greater lower bound) is also provided in this table. Out of 30 PCBs studied, lower bound 1 provides a larger lower bound value for 17 PCBs. We could not identify the conditions in terms of number of locations and number of component types on the PCB under which each lower bound calculation scheme provides better results based on the problems that we consider (See Figure 4.1 and 4.2).

Table 4.2. Lower bounds and heuristic solutions

| **PCB Name** | **Lower** | | **Heuristic 1** | | | **Heuristic 2** | | |
|---|---|---|---|---|---|---|---|---|
| | **Tour length (msec.)** | | **Tour length (msec.)** | | **% Gap from best LB** | **Tour length (msec.)** | | **% Gap from best LB** |
| | **LB1** | **LB2** | **Constructed** | **Improvement** | | **Constructed** | **Improvement** | |
| POWERSTAR_PNP_N | 16054 | 17538 | 34801 | 24698 | 40.83 | 62291 | 26295 | 49.93 |
| ZARC00076-AAA | 31613 | 19749 | 75645 | 39986 | 26.49 | 176832 | 39510 | 24.98 |
| ZGDA33AAA | 9075 | 14855 | 28995 | 19980 | 34.5 | 21003 | 17625 | 18.65 |
| A0V01 | 16810 | 17434 | 43233 | 29782 | 70.83 | 72101 | 29329 | 68.23 |
| HBDK00437PNP | 19259 | 22912 | 42008 | 35653 | 55.61 | 37343 | 31619 | 38 |
| VECTOR-V03 | 46503 | 41025 | 95578 | 72176 | 55.21 | 267826 | 69474 | 49.4 |
| HBDK00414 | 53065 | 43834 | 103151 | 72362 | 36.36 | 127733 | 72639 | 36.89 |
| HBDK00437AAC2 | 27411 | 29668 | 68937 | 43547 | 46.78 | 62498 | 40829 | 37.62 |
| HBDK00446 | 31456 | 35492 | 70652 | 59220 | 66.85 | 76192 | 55240 | 55.64 |
| HBDK00386-AAA-P | 43627 | 37763 | 138429 | 84508 | 93.71 | 522691 | 97566 | 123.64 |
| 7241-0101-SS1 | 50011 | 52451 | 298513 | 134035 | 155.54 | 370809 | 121473 | 131.59 |
| PIO2-1 | 89534 | 73876 | 193935 | 160529 | 79.29 | 745314 | 179358 | 100.32 |
| 100900022NEC | 80911 | 73780 | 283422 | 168308 | 108.02 | 423525 | 150843 | 86.43 |
| 7550-0078-SS | 65379 | 62514 | 146618 | 111096 | 69.93 | 195422 | 102348 | 56.55 |
| PIO68 | 133664 | 145206 | 218001 | 165700 | 14.11 | 514202 | 184153 | 26.82 |
| PS03 | 29109 | 17783 | 44194 | 32486 | 11.6 | 149712 | 33175 | 13.97 |
| FT10-2 | 26579 | 18226 | 49847 | 33374 | 25.57 | 91331 | 36975 | 39.11 |
| CMD4B1 | 12802 | 12627 | 24362 | 22343 | 74.53 | 35624 | 21603 | 68.75 |
| CPUMOD | 25330 | 17817 | 30962 | 26634 | 5.15 | 58274 | 26986 | 6.54 |
| FIRIN | 26406 | 16390 | 63645 | 33495 | 26.85 | 74676 | 34590 | 30.99 |
| EVM224SL | 46987 | 46888 | 120006 | 77246 | 64.4 | 195850 | 71350 | 51.85 |
| DSS16-2 | 40140 | 33857 | 120314 | 68134 | 69.74 | 198045 | 62156 | 54.85 |
| CCSLAVE2 | 52300 | 49287 | 144532 | 68384 | 30.75 | 268235 | 77784 | 48.73 |
| 200CPU | 45003 | 44339 | 125346 | 69119 | 53.59 | 267557 | 66931 | 48.73 |
| DS200SBP | 61443 | 52250 | 84574 | 70074 | 14.05 | 144217 | 82608 | 34.45 |
| 48EX26 | 111950 | 128260 | 333414 | 217658 | 69.7 | 485888 | 190850 | 48.8 |
| 128EX1 | 235717 | 265060 | 494641 | 368984 | 39.21 | 1E+06 | 695641 | 162.45 |
| DS10BP | 112192 | 117030 | 330542 | 174426 | 49.04 | 1E+06 | 174639 | 49.23 |
| 224CPUOPT | 95424 | 102218 | 313682 | 172124 | 68.39 | 641861 | 187006 | 82.95 |
| MS48ABA | 274069 | 319197 | 818363 | 526005 | 64.79 | 529695 | 400656 | 25.52 |

From Table 4.2 it can be observed that heuristic 1 provides better results than the heuristic 2 for 14 PCBs. At the construction phase, heuristic 1 always generates better schedules than heuristic 2. Heuristic 1 constructs solutions by merging subtours (of component types) as it considers feeder locations of the component types; whereas, in heuristic 2 component locations are not involved in the construction phase. Therefore, if a PCB's component types are placed close in the feeder carrier (a feeder arrangement where the locations of component types are designed specifically for a PCB type) heuristic 2 is expected to perform better than heuristic 1 regarding the constructed solution. Similar to the lower bound calculations schemes, we could not identify the conditions on number of component types and locations under which each heuristic performs better.

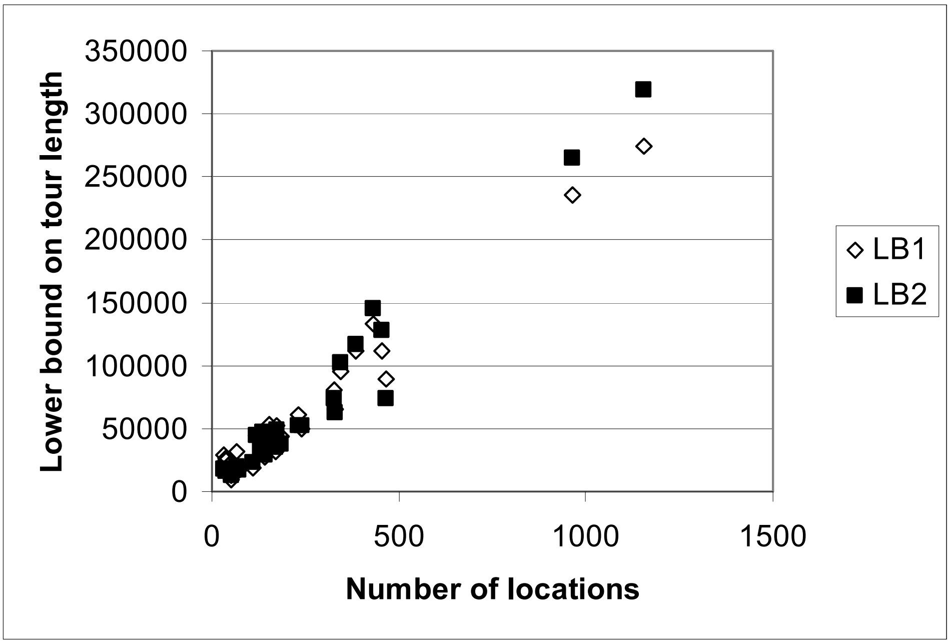


Figure 4.1 – Lower bound on tour length calculated according to two schemes proposed versus number of locations

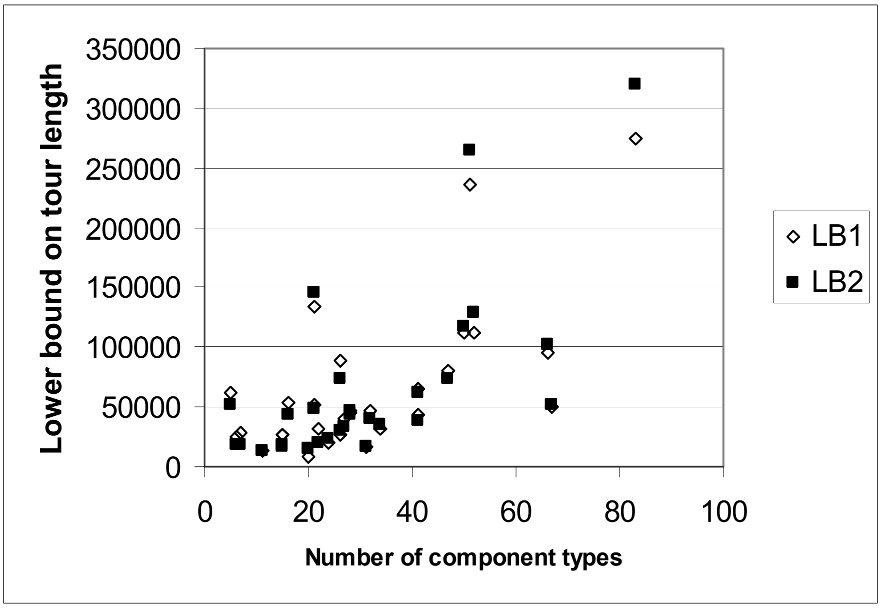


Figure 4.2 – Lower bound on tour length calculated according to two schemes proposed versus number of component types

In Table 4.3, we summarize the percentage deviation from the best lower bound statistics based on 30 PCBs studied. These summary statistics indicate that heuristic 1 provides better results in terms of average, best and worst case performance.

Table 4.3. Summary statistics for percentage deviation from the best lower bound

| | Heuristic 1 | Heuristic 2 |
|---|---|---|
| Average | 54.04 | 55.72 |
| Maximum | 155.54 | 162.44 |
| Minimum | 5.15 | 6.54 |
| Standard Deviation | 31.55 | 35.52 |

Heuristic 1 provides more robust results, in other words the standard deviation of the deviations from the best lower bound is smaller.

Another performance criterion in the comparison of the procedures proposed is the run time of the heuristics. The run times of lower bounding schemes and heuristics are given on a computer with Pentium M 2 GHz processor, are provided in Table 4.4. Run time requirements are higher for heuristic 1 than heuristic 2 for small PCBs. Heuristic 1 requires longer time since constructed solution is found after obtaining the subtours and merging processes. Heuristic 2 constructs a solution in less time since it does not require any iterative processes. Although constructed solutions of heuristic 1 is better as stated previously, improvement phase for small PCBs does not require much time. Heuristic 1 is better in run time performance for larger PCBs, as

Table 4.4. Run time of Heuristics and Lower Bounds

| **PCB Name** | **Heuristic 1 (sec)** | **Heuristic 2 (sec)** | **LB1 (sec)** | **LB2 (sec)** |
|---|---|---|---|---|
| FT10-2 | 43 | 20 | 7 | 7 |
| PS03 | 35 | 12 | 5 | 6 |
| CMD4B1 | 45 | 25 | 5 | 5 |
| CPUMOD | 65 | 18 | 6 | 6 |
| FIRIN | 40 | 15 | 6 | 6 |
| EVM224SL | 368 | 332 | 6 | 9 |
| DSS16-2 | 357 | 108 | 9 | 9 |
| CCSLAVE2 | 497 | 369 | 6 | 10 |
| 200CPU | 436 | 150 | 6 | 8 |
| DS200SBP | 410 | 1060 | 7 | 7 |
| 48EX26 | 9400 | 13678 | 21 | 124 |
| 128EX1 | 108426 | 193457 | 149 | 354 |
| DS10BP | 5394 | 17055 | 14 | 20 |
| 224CPUOPT | 3359 | 6396 | 13 | 36 |
| MS48ABA | 231400 | 506800 | 252 | 662 |
| POWERSTAR_PNP_N | 84 | 55 | 7 | 7 |
| ZARC00076-AAA | 390 | 155 | 7 | 9 |
| ZGDA33AAA | 46 | 35 | 7 | 8 |
| A0V01 | 65 | 45 | 7 | 8 |
| HBDK00437PNP | 153 | 62 | 8 | 12 |
| VECTOR-V03 | 354 | 308 | 8 | 21 |
| HBDK00414 | 331 | 285 | 7 | 16 |
| HBDK00437AAC2 | 285 | 177 | 8 | 13 |
| HBDK00446 | 463 | 349 | 8 | 17 |
| HBDK00386-AAA-P | 728 | 1028 | 9 | 10 |
| 7241-0101-SS1 | 1545 | 2370 | 9 | 15 |
| PIO2-1 | 4911 | 20917 | 24 | 19 |
| 100900022NEC | 2746 | 6284 | 14 | 25 |
| 7550-0078-SS | 2272 | 3657 | 13 | 133 |
| PIO68 | 2569 | 15262 | 20 | 17 |

it obtains better constructed solution, it takes less time for the heuristic to reach a solution in improvement phase. Besides as number of locations increase number of possible pairs to be cut and connected increase in the modified 2-opt phase. It is observed that for 12 out of 30 PCBs, heuristic 2 requires more time to reach a solution. In general, run times are expected to increase as both number of locations and number of component types on the PCB, we could not observe a numeric relation.

In order to investigate the benefits that can be obtained by the application of the heuristics, we compare the best heuristic's results with the company's solutions. The results are tabulated in Table 4.5. The minimum, maximum and average percent improvements are 7.88%, 70.96% and 26.49%, respectively. In order to capture the overall impact on the company's side, we consider demand weighted percent improvement. This figure is around 24.40%, it is found out that there is no relationship between the amount of PCB demand with the improvements obtained with the heuristics.

Table 4.5. Percent improvements by the best heuristic over the company's solutions

| **PCB Name** | **Tour length best heuristic (msec.)** | **Tour length company's solution (msec.)** | **% improvement** |
|---|---|---|---|
| POWERSTAR_PNP_N | 24698 | 42481 | 41.86 |
| ZARC00076-AAA | 39510 | 136058 | 70.96 |
| ZGDA33AAA | 17625 | 21108 | 16.50 |
| A0V01 | 29329 | 37410 | 21.60 |
| HBDK00437PNP | 31619 | 35835 | 11.77 |
| VECTOR-V03 | 69474 | 156155 | 55.51 |
| HBDK00414 | 72362 | 89962 | 19.56 |
| HBDK00437AAC2 | 40829 | 55428 | 26.34 |
| HBDK00446 | 55240 | 65912 | 16.19 |
| HBDK00386-AAA-P | 84508 | 99452 | 15.03 |
| 7241-0101-SS1 | 121473 | 133256 | 8.84 |
| PIO2-1 | 160529 | 195194 | 17.76 |

Table 4.5. Percent improvements by the best heuristic over the company's solutions (continued)

| **PCB Name** | **Tour length best heuristic (msec.)** | **Tour length company's solution (msec.)** | **% improvement** |
|---|---|---|---|
| 100900022NEC | 150843 | 229014 | 34.13 |
| 7550-0078-SS | 102348 | 111106 | 7.88 |
| PIO68 | 165700 | 204547 | 18.99 |
| PS03 | 32486 | 49141 | 33.89 |
| FT10-2 | 33374 | 41803 | 20.16 |
| CMD4B1 | 21603 | 36219 | 40.35 |
| CPUMOD | 26634 | 32343 | 17.65 |
| FIRIN | 33495 | 45492 | 26.37 |
| EVM224SL | 71350 | 84024 | 15.08 |
| DSS16-2 | 62156 | 94969 | 34.55 |
| CCSLAVE2 | 68384 | 84748 | 19.31 |
| 200CPU | 66931 | 122099 | 45.18 |
| DS200SBP | 70074 | 84914 | 17.48 |
| 48EX26 | 190850 | 215625 | 11.49 |
| 128EX1 | 368984 | 420760 | 12.31 |
| DS10BP | 174426 | 404173 | 56.84 |
| 224CPUOPT | 172124 | 249554 | 31.03 |
| MS48ABA | 400656 | 573620 | 30.15 |

The main reason behind the observation that there is no consistent behavior of lower bounds, individual heuristic performances and the percent improvement over the company's solution with respect to changing number of locations and component types on the PCBs, is the fact that these two characteristics carry very restricted information about the complexity of the problem instances. The performance of the

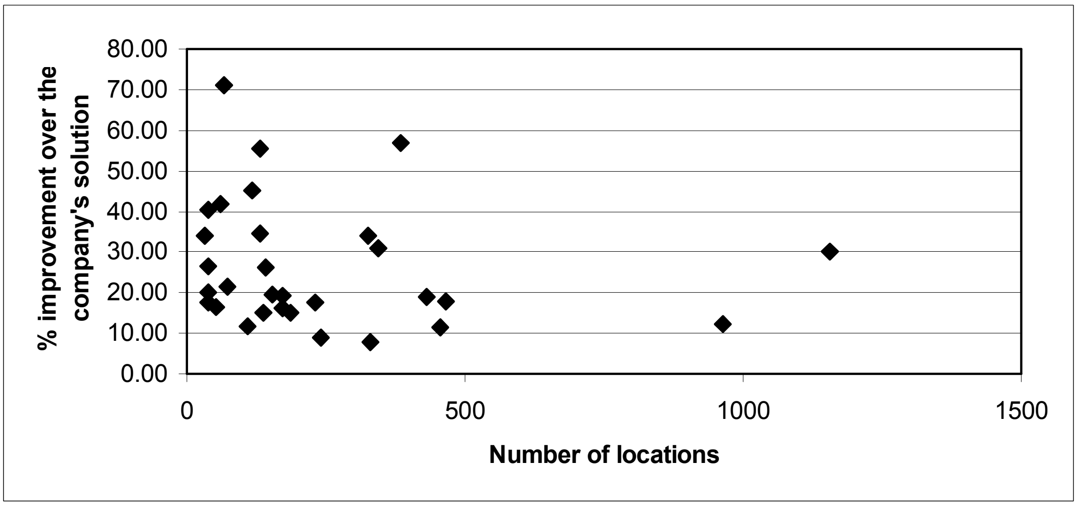


Figure 4.3 – Percent improvement over the company's solution versus number of locations on the PCB

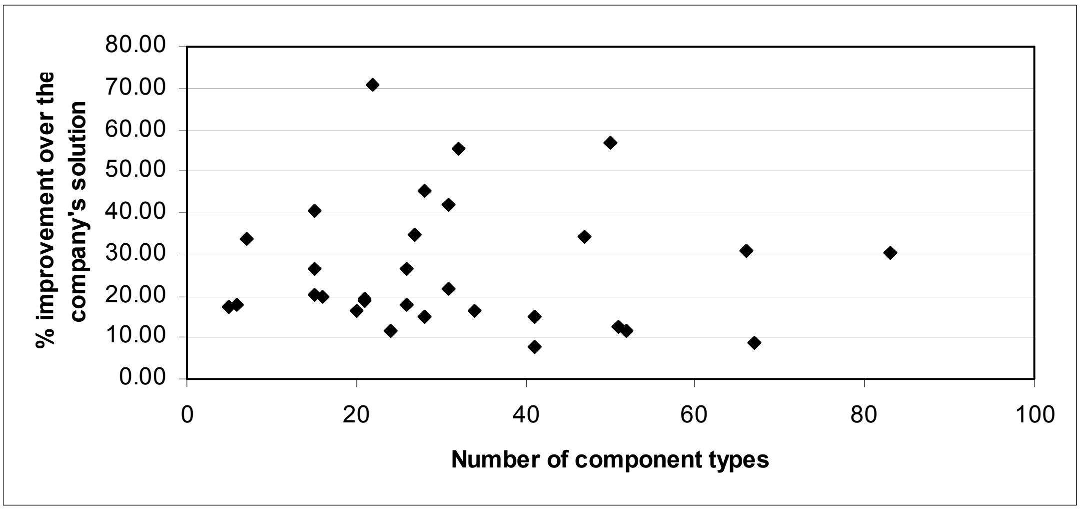


Figure 4.4. Percent improvement over the company's solution versus number of component types on the PCB

both heuristics and the lower bounds proposed heavily depend on other factors like the arrangement of component feeders, the speed requirements of different types of components, and so on. Especially the dominance relation between table movement times in between the locations, the turret rotation time and the feeder time plays the most important role on the performance of the procedures that we propose. In order to assess the performance of the heuristics and lower bounding schemes against these factors, we set up an experiment. The results of the study are discussed in Section 4.2.

Note that we could not obtain the optimal solutions for none of these 30 PCBs. In order to assess the performance of the lower bounding schemes and heuristic procedures on real PCBs that the company produces, we consider 4 PCBs for which both number of locations and component types are very small. We also restrict the problem complexity by reducing the number of heads on the turret to four from twelve. Therefore, for these problem instances, it is not meaningful to compare the generated results with company's solutions since the latter ones are proposed for the machine with twelve heads. In Table 4.6 we provide optimal tour lengths, the lower bounds under two schemes proposed and the two heuristic's results. The columns "% Gap" provide figures that are calculated against the optimal tour lengths. As it can be seen, there is no dominance of one lower bounding method. It can be seen that heuristic 2 performs better than heuristic 1. The performances of the heuristics are quite good; the maximum percentage gap from the optimal figure is almost 2.3%. On the other hand, our lower bounds are not as good as the heuristics, even the minimum percent deviation from the optimal is around 7.5%.

Table 4.6. The performance of the lower bounding schemes and heuristics with respect to optimal solutions.

| **PCB #** | **Number of Locations** | **Number of component types** | **Optimal tour length (msec.s)** | **LB1** | | **LB2** | | **Heuristic 1** | | **Heuristic 2** | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | **Tour length (msec.s)** | **%Gap** | **Tour length (msec.s)** | **%Gap** | **Tour length (msec.s)** | **% Gap** | **Tour length (msec.s)** | **% Gap** |
| 1 | 10 | 2 | 7235 | 6485 | 10.37 | 4463 | 38.31 | 7302 | 0.93 | 7235 | 0.00 |
| 2 | 12 | 3 | 5723 | 3730 | 34.82 | 5293 | 7.51 | 5758 | 0.61 | 5723 | 0.00 |
| 3 | 13 | 6 | 14276 | 12855 | 9.95 | 7932 | 44.44 | 14508 | 1.63 | 14527 | 1.76 |
| 4 | 10 | 5 | 16452 | 15417 | 6.29 | 12003 | 27.04 | 16452 | 0.00 | 16452 | 0.00 |

## 4.2. Experimental Study

In order to investigate the parameter sensitivity of the proposed heuristics and lower bounds' performances, we set up an experimental study. We first randomly generate *x* and *y* coordinates of 10 locations on a 10*25 cm PCB; i.e., *x* and *y* coordinates are generated from uniform distribution over the range 0 and 10, and over the range 0 and 25, respectively. 10 different PCB instances are generated in that way. For number of different components to be mounted on these locations, we consider two different cases: there are either 2 or 4 different component types. Each location's component type is randomly generated from discrete uniform distribution over the range 1 and 2, or 1 and 4, depending on the number of different component types case. For the problem instances, where the generated number of different components is less than the desired number, a new set of random variates are generated until we have exactly 2 or 4 different component types on the PCB.

In order to assess the performance of the proposed heuristics on different scenarios that differ in the dominance of table movement time or the feeder time, we consider the following scenarios:

- Component feeder locations: The component feeders are sorted in the index of component type, i.e., component feeders are arranged in the order of 1, 2, or 1, 2, 3, 4. For the distance between the racks, the actual distances in the company considered. Two cases on the racks including different components are considered.

    a. Case A: Component type *c*'s feeder is next to component type *c*+1's feeder, i.e., racks including different components are adjacent to each other.

    b. Case B: There are four racks in between component type *c*'s feeder and component type *c*+1.

- PCB dimension: In order to assess the performance against different table movement times, we change the dimension of PCBs. In addition the original 10*25 cm PCBs, we rescale the locations generated to 1*2.5 cm, and 2*5 cm PCBs. Note that, since the table movement times are the results of Tchebycheff movement in between the locations, reducing the size from

10*25 to 1*2.5 is equivalent to speeding up the table movement 10 times, and to 2*5 is equivalent to speeding the table movement type 5 times. For the movement times, actual speed of the machine is considered.

Instead of twelve heads on the machine, we consider four heads to reduce the computational complexity. For a PCB having two different component types, as one of them is given ID as 1, the other's ID is given as 2. Feeder racks and turret rotation indices are assigned according to the component type ID, for example component type ID 1 has the turret rotation index of 1. Turret rotation times for differently indexed components are taken as in the real life environment. Table 4.7 summarizes the factors and levels of these factors. PCB data used in the experiment is available in Appendix G.

Table 4.7: Factors and the corresponding levels in the experiment

| **Factor** | **Levels** |
|---|---|
| PCB size (table movement speed) | **1.** 1*2.5 (Table movement speed is 10 times of the actual speed)<br>**2.** 2*5 (Table movement speed is 5 times of the actual speed)<br>**3.** 10*25 (Table movement speed is the actual speed) |
| Number of different component types | **1.** 2<br>**2.** 4 |
| Feeder placement | **1.** **Case A:** Feeders are next to each other<br>**2.** **Case B:** There are four racks in between the feeder carriages. |

There are 3*2*2=12 combinations of these factors, and 10 PCBs for each combination are considered for each combination. Therefore, in a total of 120 problem instances are solved in the study. The lower bounds and heuristic solutions are found on a computer with Pentium M processor (2 GHz). On the other hand, the optimal solutions are found on another computer with Pentium 4 processor (1.5 GHz).

For each combination of the factors of the experiment, we report the average, maximum, minimum and standard deviation of the tour lengths generated by upper bounding methods, and heuristics from the optimal tour lengths. These basic statistics are provided in Table 4.8.

The following observations are made based on Table 4.8:

**The relative performance of two lower bounding schemes:** The second lower bounding method outperforms the first one for every problem instance where

- the feeders of different components placed next to each other, i.e., case A
- when the size of the PCB is large in terms of dimension, i.e., 2*5 cm, and there are 4 racks in between two feeders, i.e., case B.

The first lower bounding method outperforms the second one for every problem instance where

- the size of the PCB is small in terms of its dimensions, i.e., 1*2.5 cm and feeders are placed next to each other, i.e., case A.

Notice that the factors levels where the second lower bounding scheme outperforms the first are the cases where table movement times dominate the feeder times: either feeders are located next to each other or feeders are farther from each other but the dimension of the card is bigger. Similarly, the parameter settings where the first lower bounding method outperforms the second one are the ones under which the feeder times dominate the table movement times, the feeders are located relatively far from each other and the dimension of PCB is smaller.

Table 4.8. Basic Statistics on the percentage deviation from the optimal tour length under different levels of the factors

| Feeder Placement | Number of different component types | | 2 | | | | | | 4 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PCB Size | % deviation from The optimal | LB1 | LB2 | Best LB | Heuristic 1 | Heuristic 2 | Best Heuristic | LB1 | LB2 | Best LB | Heuristic 1 | Heuristic 2 | Best Heuristic |
| Case A | 1*2.5 | Minimum | 26.14 | 9.37 | 9.37 | 0 | 0 | 0 | 42.65 | 24.2 | 24.2 | 0 | 0 | 0 |
| | | Maximum | 34.74 | 20.37 | 20.37 | 0.53 | 1.27 | 0 | 54.06 | 35.57 | 35.57 | 2.37 | 0 | 0 |
| | | Average | 28.99 | 13.93 | 13.93 | 0.05 | 0.16 | 0 | 48.95 | 28.9 | 28.9 | 0.72 | 0 | 0 |
| | | Std. Dev. | 2.52 | 3.65 | 3.65 | 0.17 | 0.4 | 0 | 4.07 | 3.13 | 3.13 | 0.85 | 0 | 0 |
| | 2*5 | Minimum | 26.33 | 0 | 0 | 0 | 0 | 0 | 40.91 | 5.4 | 5.4 | 0 | 0 | 0 |
| | | Maximum | 36.41 | 7.89 | 7.89 | 7.99 | 2.16 | 0 | 52.11 | 28.83 | 28.83 | 4.27 | 2.52 | 2.52 |
| | | Average | 32.32 | 4.35 | 4.35 | 0.87 | 0.22 | 0 | 46.02 | 18.1 | 18.1 | 1.49 | 0.31 | 0.31 |
| | | Std. Dev. | 3.38 | 3.02 | 3.02 | 2.51 | 0.68 | 0 | 3.81 | 7.32 | 7.32 | 1.77 | 0.8 | 0.8 |
| | 10*25 | Minimum | 32.28 | 0 | 0 | 0 | 0 | 0 | 44.05 | 0 | 0 | 0 | 0 | 0 |
| | | Maximum | 47.53 | 0.68 | 0.68 | 3.08 | 0 | 0 | 63.04 | 2.68 | 2.68 | 8.25 | 1.02 | 1.02 |
| | | Average | 39.61 | 0.07 | 0.07 | 0.46 | 0 | 0 | 55.31 | 0.48 | 0.48 | 1.39 | 0.1 | 0.1 |
| | | Std. Dev. | 4.24 | 0.22 | 0.22 | 1.04 | 0 | 0 | 6.19 | 0.89 | 0.89 | 2.66 | 0.32 | 0.32 |
| Case B | 1*2.5 | Minimum | 12.16 | 16.15 | 12.16 | 0 | 0 | 0 | 0.91 | 20.54 | 0.91 | 0 | 0 | 0 |
| | | Maximum | 19.49 | 24.97 | 19.49 | 1.67 | 6.14 | 0 | 21.15 | 35.43 | 21.15 | 1.87 | 0 | 0 |
| | | Average | 15.45 | 19.48 | 15.45 | 0.38 | 0.61 | 0 | 13.04 | 27.06 | 13.04 | 0.26 | 0 | 0 |
| | | Std. Dev. | 2.1 | 2.71 | 2.1 | 0.65 | 1.94 | 0 | 6.01 | 4.95 | 6.01 | 0.61 | 0 | 0 |
| | 2*5 | Minimum | 16.76 | 6.63 | 6.63 | 0 | 0 | 0 | 9.39 | 18.36 | 9.39 | 0 | 0 | 0 |
| | | Maximum | 27.44 | 17.05 | 16.76 | 4.09 | 2.1 | 1.38 | 25.03 | 31.34 | 25.03 | 3.78 | 4.97 | 0.68 |
| | | Average | 22.25 | 12.29 | 12.27 | 0.97 | 0.35 | 0.14 | 18.83 | 23.93 | 18.65 | 0.87 | 0.67 | 0.07 |
| | | Std. Dev. | 3.81 | 3.59 | 3.55 | 1.61 | 0.75 | 0.44 | 4.39 | 4.21 | 4.36 | 1.47 | 1.54 | 0.21 |
| | 10*25 | Minimum | 27.97 | 0 | 0 | 0 | 0 | 0 | 35.05 | 0 | 0 | 0 | 0 | 0 |
| | | Maximum | 44.23 | 1.19 | 1.19 | 3.08 | 0 | 0 | 52.76 | 3.04 | 3.04 | 8.2 | 0 | 0 |
| | | Average | 35.76 | 0.18 | 0.18 | 0.46 | 0 | 0 | 43.99 | 1.26 | 1.26 | 1.33 | 0 | 0 |
| | | Std. Dev. | 4.48 | 0.4 | 0.4 | 1.04 | 0 | 0 | 6.08 | 0.93 | 0.93 | 2.82 | 0 | 0 |

In between these two extreme cases, i.e., under case B, for 2*5 PCBs, we do not have such a consistent relative performance relation over all instances. When number of component types is 2, the second lower bound generates better results in terms of worst, best and average performances. On the other hand when there are 4 types of components, the first lower bound is better in terms of these three.

In general it is observed that the first lower bounding method performance gets better as the feeder time dominates the table movement time; i.e., as the size of PCB increases, or the speed of table decreases, or the feeders are far from each other. On the other hand, the performance of the second lower bounding method gets worse under these conditions.

**The sensitivity of lower bounding schemes with respect to number of component types:** In order to assess this we calculate the percentage increase in gap from the optimal tour when the number of component types increases from 2 to 4 in the worst and the average performance. The figures are provided in Table 4.9.

Table 4.9: Change in % gap from optimal as the number of component types is increased from 2 to 4.

| Feeder Placement | PCB size | | LB1 | LB2 |
|---|---|---|---|---|
| Case A | 1*2.5 | Maximum | 55.59 | 74.62 |
| | | Average | 68.84 | 107.40 |
| | 2*5 | Maximum | 43.12 | 265.21 |
| | | Average | 42.38 | 316.13 |
| | 10*25 | Maximum | 32.64 | 293.59 |
| | | Average | 39.64 | 606.09 |
| Case B | 1*2.5 | Maximum | 8.51 | 41.90 |
| | | Average | -15.57 | 38.92 |
| | 2*5 | Maximum | -8.78 | 83.82 |
| | | Average | -15.40 | 94.68 |
| | 10*25 | Maximum | 19.27 | 155.66 |
| | | Average | 23.00 | 612.67 |

It can be observed that lower bound 2 is more sensitive to an increase in the number of components, its both average and worst case performances get worse with an increase in number of component types. On the other hand, the second lower bound is less sensitive to this parameter. It provides better results when the feeder time dominates the table movement time.

**The overall performance of the lower bounding schemes:** When we examine the overall problem instances, the percentage deviation of the best lower bound from the optimal tour is maximum 35.57%, and the average is 10.56%. This shows a room for improvement in lower bounding schemes.

**The relative performance of two heuristics proposed:** The second heuristic outperforms the first one for problem instances except the following parameter sets

- PCB size is 1*2.5, there are 2 component types under case A: Average and worst case performances of heuristic 1 is better.

- PCB size is 2*5, there are 2 component types under case B: Although there is no consistent behavior, heuristic 2 performs better on the average and based on the worst performance.

In general we expect that heuristic 1 under which the construction is based on the subtours of different component types are combined provide good results when

- the number of different component types is large,

- the number of locations requiring a particular component type is large.

In this study, due to time pressure we could not find any chance to examine such problem instances due to high computational requirements for the optimal solutions. Nevertheless, the maximum percentage deviation from optimal is 8.25% for heuristic 1 over all problem instances considered.

**The overall performance of the heuristics:** When we examine the overall problem instances, except 4 out of 120 problem instances, we always find the optimal tour with heuristic 1 or heuristic 2. The maximum error made by the best heuristic 2.52% over all instances.

**Runtime performance of the heuristics:** Table 4.10 is a summary table for the run time figures of the random generated PCBs. Although construction and improvement phase of the heuristics are examined together, it can be concluded that much of the run time comes from the construction part of the heuristics. Improvement phase does not take more than 1 second, therefore as heuristic 1 decomposes locations according to the component types, we have subtours to be solved as much as the number of components. It is also seen that as heuristic 2 needs one tour to be solved, it requires approximately half the time of heuristic 1 for the PCB card groups with component type size 2. When number of component types increase run time of Heuristic 1 is higher. Lower bound run time is close to each other for each group investigated. Optimal solutions are found in different times, since different groups of PCBs have different characteristics; for example in PCBs with 10*25 cm dimensions, table movement distances are more with respect to feeder movement and turret durations which means the optimal will treat the problem as a TSP and find the solution in less time compared to the other problems. If concurrency is forced by the entrance of other distances; i.e., turret durations, feeder and table movement distances get closer, the run time to acquire optimal solution is higher.

Table 4.10. Run time of heuristics and lower bounds for the random generated PCBs.

| Feeder Placement | PCB size | | Component Size = 2 | | | | | Component Size = 4 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | **LB1** | **LB2** | **Optimal** | **Heuristic 1** | **Heuristic 2** | **LB1** | **LB2** | **Optimal** | **Heuristic 1** | **Heuristic 2** |
| **Case A** | **1*2.5** | **Minimum** | 4 | 4 | 195 | 10 | 5 | 5 | 5 | 2064 | 20 | 6 |
| | | **Maximum** | 7 | 7 | 56728 | 15 | 9 | 7 | 7 | 71806 | 28 | 9 |
| | | **Average** | 5.6 | 6 | 10631.1 | 12.4 | 6.8 | 6.1 | 6.4 | 33831.4 | 25.7 | 7 |
| | | **Std.Dev** | 0.97 | 1.15 | 17246.42 | 1.96 | 1.14 | 0.74 | 0.84 | 26808.31 | 2.79 | 1.05 |
| | **2*5** | **Minimum** | 6 | 6 | 87 | 13 | 6 | 4 | 4 | 1087 | 20 | 5 |
| | | **Maximum** | 7 | 7 | 1534 | 16 | 8 | 5 | 5 | 21058 | 21 | 7 |
| | | **Average** | 6.5 | 6.4 | 404.3 | 14.5 | 6.8 | 4.8 | 4.9 | 4568.2 | 20.2 | 5.9 |
| | | **Std.Dev** | 0.53 | 0.52 | 429.99 | 0.85 | 0.63 | 0.42 | 0.32 | 5905.93 | 0.42 | 0.57 |
| | **10*25** | **Minimum** | 4 | 5 | 13 | 10 | 5 | 5 | 5 | 15 | 18 | 5 |
| | | **Maximum** | 7 | 7 | 102 | 15 | 8 | 7 | 8 | 673 | 38 | 8 |
| | | **Average** | 5.6 | 6 | 46.4 | 11.8 | 5.7 | 6.3 | 6.5 | 156.2 | 26.7 | 6.7 |
| | | **Std.Dev** | 1.17 | 1.05 | 30.54 | 1.99 | 0.95 | 0.82 | 0.97 | 219.45 | 5.44 | 1.16 |
| **Case B** | **1*2.5** | **Minimum** | 4 | 4 | 501 | 10 | 5 | 4 | 5 | 1237 | 19 | 5 |
| | | **Maximum** | 5 | 5 | 44009 | 11 | 6 | 5 | 5 | 33004 | 21 | 6 |
| | | **Average** | 4.4 | 4.9 | 10013 | 10.5 | 5.3 | 4.8 | 5 | 10117.1 | 19.7 | 5.5 |
| | | **Std.Dev** | 0.52 | 0.32 | 12642.55 | 0.53 | 0.48 | 0.42 | 0 | 10844.97 | 0.67 | 0.53 |
| | **2*5** | **Minimum** | 4 | 4 | 187 | 10 | 5 | 4 | 5 | 1689 | 19 | 4 |
| | | **Maximum** | 5 | 5 | 5649 | 20 | 6 | 5 | 5 | 7554 | 22 | 7 |
| | | **Average** | 4.8 | 4.8 | 1480.8 | 12.75 | 5.25 | 4.9 | 5 | 4349.1 | 19.7 | 5.3 |
| | | **Std.Dev** | 0.42 | 0.42 | 1680.76 | 4.2 | 0.46 | 0.32 | 0 | 1967.25 | 0.95 | 0.82 |
| | **10*25** | **Minimum** | 4 | 5 | 10 | 10 | 5 | 4 | 4 | 18 | 20 | 5 |
| | | **Maximum** | 5 | 5 | 142 | 12 | 6 | 5 | 5 | 256 | 20 | 5 |
| | | **Average** | 4.4 | 5 | 58.7 | 11 | 5.1 | 4.8 | 4.9 | 113.9 | 20 | 5 |
| | | **Std.Dev** | 0.52 | 0 | 41.14 | 0.67 | 0.32 | 0.42 | 0.32 | 79.18 | 0 | 0 |

# CHAPTER 5

# CONCLUSION

The pressure of global competition in the electronic devices sector demands the productivity and efficiency increase in the manufacturing processes through the technological development of the state of the art machinery and equipment as well as improvements in the decision process to use these devices. Our study focuses on the latter for a certain equipment setting in a real manufacturing environment. We have designed a decision methodology for some of the manufacturing processes at that equipment in this thesis work. The application of our methodology on two families of real products yielded a considerable increase (24.4 %) in productivity.

The production environment investigated is the SMT manufacturing process on a turret style SMT machine. The aim is to increase the productivity of the machine by improving the component placement sequence in a prespecified family setups, i.e. the feeder arrangement. The problem of determining a component placement sequence under three concurrent events (table movement, turret rotation and feeder movement) is modelled and formulated. We followed an evaluational approach in modelling; a combinatorial optimization formulation as a conceptual model, a non-linear mixed integer programming formulation taking some part of the real environment into consideration, and finally a linear-mixed integer formulation for the upmost degree of representation for the placement sequence determination problem as far as we could. The final formulation is NP-Hard in the strong sense, since we may detect special cases as symmetric/assymmetric TSP problems if we oversimplfy the problem. Thus, we resort our efforts on the development of solution methodology to heuristics.

We have designed two heuristics. In the first algorithm, PCB locations are decomposed according to the component types, near optimal tours for each subtours are found using a software, CONCORDE, by considering table movement and turret rotation times only. The subtours are then merged to form a constructed tour by using a myopic local search. As an improvement step over the constructed tour, we apply a modified 2-opt. In the second algorithm, we treat all locations (irrespective of their component types) together and form a distance matrix as a function of table movement and turret rotation durations only. We use CONCORDE to get a constructed tour; that is improved further by using the same modified 2-opt. It is observed that the second heuristic provides better tour durations for PCBs whose feeder positions of the component types are located close, and the first algorithm gives better figures when the feeder arrangement of the component types of the PCB are far from each other. Another observation is that the quality of the constructed tour determines the amount of improvement that can be achieved. The run times for the heuristics indicate that solutions can be obtained in reasonable times.

A comparison between the optimal values and the heuristics' results for the placement sequencing problem instances of the real PCBs could not be carried out, since we are not able to find the optimal values. Instead, two lower bounds are developed to test the performance of the heuristics. The first lower bounding approach considers table movement times and rotation times to compile a symmetric distance matrix in such a way that the distance values are lower bounds to the durations of the three concurrent events. Then, the corresponding TSP problem is solved, whose optimal tour length yields a lower bound. Consequently, this lower bound is improved with the use of the optimal TSP tour based on only feeder movement times. The second lower bound is calculated by adding the minimum length of a path, of size equal to the difference between number of locations and the number of different components, in the complete graph generated by the distance matrix compiled to the length of the optimal TSP tour considering only the feeder movements.

We set two experiments to evaluate the performance of our methodology. We considered four real but small PCBs and 120 randomly generated problem instances to compare our heuristics and lower bounds with the optimum solutions. The machine characteristic is modified such that the turret has 4 heads instead of 12 to find the optimum solutions in a reasonable time by means of CPLEX. Our heuristics are able to find almost all of the optimum solutions. Although there is a room for improvement for the lower bounds, they seem to be performing satisfactorily. In the second computational study, we consider 30 PCBs that are currently in the product mix of the company. In these 30 problem instances, we compare the heuristics' results with the lower bounds and the company's solution generated by an expert touch over the solutions provided by the machine specific optimizer software. We could not identify the conditions in terms of number of locations and number of component types on the PCB under which each lower/upper bound calculation scheme provides better results based on the problems that we consider. However, when we apply demand weighted percent improvement as the sole measure to capture the overall impact on the company's side, we get 24.4% improvement.

We address the following further research directions:

- Throughout the study, we assumed that the acceleration/deceleration effect is negligible. The solution technique we developed could easily be modified to incorporate these effects.

- We also assumed that feeder assignment is given for all family of products. Better results for our comparative study could be obtained if the feeder assignment is done for the two families under study. We may investigate the effect of feeder placement change or equivalently robustness of the solution quality with respect to feeder setup.

- One may improve the current lower/upper bounds and can develop new bounds. All these could be incorporated into a branch and cut/bound scheme to find the optimum solutions for larger problems.

- Another possible research direction is to study feeder assignment and component placement problems simultaneously. A starting point for such an integrated approach could be an iterative one: First solve the feeder assignment problem for a product family (which imposes constraints for the component placement sequence). Then, using the placement solutions of all PCBs in that family, provide a feedback to the feeder assignment problem and continue in this way.

- We studied a turret style SMT machine. Similar approaches could be devised for other technologies such as for sequential pick & place, dual delivery, etc.

- Balancing the overall SMT assembly line is another topic for future research. After the solution methodologies for all such technologies are developed, a production line composed of different types of automated machinery laid out in a tandom configuration could be studied. Once, a bottleneck station is found and optimized, the next bottleneck could be identified for a family. These family specific optimizations could be integrated for all families produced in that manufacturing line.

# REFERENCES

[1] Wikipedia, the free encyclopedia, *Printed Circuit Board*, Available at http://en.wikipedia.org/wiki/Printed_circuit_board#History, (2007), accessed at 12.01.2007.

[2] Yves Cramaa; Joris van de Klundert, Frits C.R. Spieksma, *Production planning problems in printed circuit board assembly*, Discrete Applied Mathematics, 123 (2002), pp.339-361.

[3] Masri Ayob, Peter Cowling, Graham Kendall, *Optimisation for Surface Mount Placement Machines*, Available at http://www.cs.nott.ac.uk/~mxa/OptimisationForSurfaceMountPlacementMachines.pdf#search=%22optimization%20for%20surface%20mount%20placement%22, accessed at 12.01.2007.

[4] Moyer, L.K. and Gupta, S. M., *SMT feeder slot assignment for predetermined component placement paths*, Journal of Electronics Manufacturing (1996), 6, 173-192.

[5] L.F. McGinnis, J.C. Ammons, M. Carlyle, L. Cranmer, G.W. DePuy, K.P. Ellis, C.A. Tovey, H. Xu, *Automated process planning for printed circuit card assembly*, IIE Trans. 24 (1992) 18–30.

[6] Wang, C., Ho, L. -S. and Cannon, D. J., *Heuristics for assembly sequencing and relative magazine assignment for robotic assembly*, Computers and Industrial Engineering (1998) 34, 423-431.

[7] Magyar, G., Johnsson, M. And Nevalainen, O, *On solving single machine optimization problems in electronics assembly*, Journal of Electronics Manufacturing (1999), 9(4), 249-267.

[8] Bentzen, B, *SMD placement*, SMT in FOCUS.. (27.11.2000), Available at: http://www.smtinfocus.com/PDF/SMD_placement.pdf, accessed at 12.01.2007.

[9] Jeevan, K., Parthiban, A., Seetharamu, K. N., Azid, I. A. and Quadir, G. A., *Optimization of PCB Component Placement using Genetic Algorithms*, Journal of Electronics Manufacturing (2002), 11(1), 69-79.

[10] P.J.M. van Laarhoven, W.H.M. Zijm, *Production preparation and numerical control in PCB assembly*, Int. J. Flexible Manuf. Systems 5 (1993), pp. 187-207.

[11] J. C. Ammons, M. Carlyle, L. Cranmer, G. DePuy, K. Ellis, L. F. McGinnis, C. A. Tovey, H. Xu, *Component allocation to balance workload in printed circuit card assembly systems*, IIE Transactions 29 (4): 265-75, 1997

[12] Moyer, L.K. and Gupta, S. M, *Simultaneous component sequencing and feeder assignment for high speed chip shooter machines*, Journal of Electronics Manufacturing, 6 (1996), 271-305.

[13] Nelson, K. M. and Wille, L.T., *Comparative study of heuristics for optimal printed circuit board assembly*, Proceedings of Southcon'95. Fort Lauderdale (1995), Florida, USA, 322-327.

[14] Sun, D. -S., Lee, T. -E and Kim, K. -H., *Component allocation and feeder arrangement for a dual-gantry multi-head surface mounting placement tool*, International Journal of Production Economics, 2005, vol. 95, issue 2, pages 245-264.

[15] M. O. Ball and M. J. Magazine, *Sequencing of insertions in printed circuit board assembly*, Operations Research, 36(2):192–201, 1988.

[16] Chan, D. and Mercier, D., *IC Insertion: an Application of the Travelling Salesmen Problem*, International Journal of Production research, Vol. 27, No. 10, pp. 1837-1841, 1989.

[17] Moyer, L. K. and Gupta, S, M., *An Efficient Assembly Sequencing Heuristic for Printed Circuit Boards Configurations*, Journal of Electronics Manufacturing, Vol. 7, No. 2, pp. 143-160, 1997.

[18] Duman, E and Or, I., *Precedence constrained TSP arising in printed circuit board assembly*, International Journal of Production Research, 42(1), (2004), 67-78.

[19] Ahmadi, J., Ahmadi, R., Matsuo, H., Tirupati, D.. *Component Fixture Positioning/Sequencing for Printed Circuit Board Assembly with Concurrent Operations*, Operations Research, Vol. 43, No. 3, pp. 444-457, 1995.

[20] T. Leipala and O. Nevalainen, *Optimization of the movements of a component placement machine*, European Journal of Operational Research, 38:167–77, 1989.

[21] Francis, R. L., McGinnis, L. F. and White, J. A., *Facility Layout and Location: An Analytical Approach*, (1992), NJ:Prentice-Hall.

[22] Foulds, L. R. and Hamacher, H. W., *Optimal bin location and sequencing in printed circuit board assembly*, European Journal of Operational Research, 66, 279-290, 1993.

[23] Crama, Y., Flippo, O.E., Van de Klundert, J., Spieksma, F., *The Assembly of Printed Circuit Boards: A Case with Multiple Machines and Multiple Board Types*, European Journal of Operational Research, Vol. 98, pp. 457-472, 1997.

[24] Ji, Z. Leu, M. C. and Wong, H., *Application of linear assignment model for planning of robotic PC board assembly*, ASME Journal of Electronic Packaging, 114, 455-460, (1992).

[25] Kumar, R. and Li, H., *Assembly Time Optimization for PCB Assembly*, Proceedings of the American Control Conference, pp. 306, 1994.

[26] De Souza, R. and Lijun, W., *Intelligent Optimization of Component Insertion in Multi-Head Concurrent Operation PCBA Machines*, Journal of Intelligent Manufacturing, Vol. 6, pp. 235-243, 1995.

[27] Ng, M., *Heuristics approach to printed circuit board insertion problem*, Journal of the Operational Resarch Society, 49, 1051-1059, 1998.

[28] Ahmadi, R. H. and Mamer, J. W., *Routing heuristics for automated pick and place machines*, European Journal of Operational Research, 117, 533-552, (1999).

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

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

[31] Concorde TSP Solver, January 2005, Available at http://www.tsp.gatech.edu/concorde.html, accessed at 12.01.2007.

[32] Thorburn, LJ and Houghton, J., *The Victorian Electronics Industry Cluster*, Australian Electrical and Electronic Manufacturers Association, (2004), Available at www.cfses.com/documents/CSES_Victorian_Electronics_Cluster_Study.pdf, accessed at 12.01.2007.

[33] A. Schrijver, *On the history of combinatorial optimization (till 1960)*, in: ``Handbook of Discrete Optimization" (K. Aardal, G.L. Nemhauser, R. Weismantel, eds.), Elsevier, Amsterdam, 2005, Available at http://homepages.cwi.nl/~lex/files/histco.pdf

[34] Karla Hoffmann, Manfred Padberg, *Traveling Salesman Problem*, Available at http://iris.gmu.edu/~khoffman/papers/trav_salesman.html, accessed at 12.01.2007.

[35] E.L. Lawler, J.K. Lenstra, A.H. Rinooy Kan, D.B. Shmoys, *The Travelling Salesmen Problem*, John Wiley and Sons, 1983.

[36] David Applegate, Robert Bixby, Vašek Chvátal, William Cook, *Finding Tours in the TSP*, Forschungsinstitut für Diskrete Mathematik, Report No. 99885, 1998, Universitaet Bonn, available at http://www.tsp.gatech.edu/methods/papers/lk_report.pdf, accessed at 12.01.2007.

[37] R. C. Prim. *Shortest connection networks and some generalizations*. Bell System Technical Journal, 36:1389-1401, 1957.

# APPENDIX A

## Table A.1. Concurrent Events on a Turret Style SMT Machine

| Step (i) | PCB Table | Turret | Feeder Carriage | Distance (travelling time for step i) |
|---|---|---|---|---|
| 1 | | | Move from the initial (0) position to 1st pick position | Feeder (0,1) |
| | | Pick 1st component | | Picking event |
| 2 | | Rotate | Move from 1st position to 2nd pick position | max{turret_rotation(1), feeder(1,2)} |
| | | Pick 2nd component | | Picking event |
| 3 | | Rotate | Move from 2nd position to 3rd pick position | max{turret_rotation(1,2), feeder(2,3)} |
| | | Pick 3rd component | | Picking event |
| 4 | | Rotate | Move from 3rd position to 4th pick position | max{turret_rotation(1,2,3), feeder(3,4)} |
| | | Pick 4th component | | Picking event |
| 5 | | Rotate | Move from 4th position to 5th pick position | max{turret_rotation(1,2,3,4), feeder(4,5)} |
| | | Pick 5th component | | Picking event |
| 6 | | Rotate | Move from 5th position to 6th pick position | max{turret_rotation(1,2,3,4,5), feeder(5,6)} |
| | | Pick 6th component | | Picking event |
| 7 | Move from initial table position to 1st placement location | Rotate | Move from 6th position to 7th pick position | max{table_movement(0,1), turret_rotation(1,2,3,4,5,6), feeder(6,7)} |
| | | Pick 7th and place 1st component | | Pick&Place event |
| 8 | Move from 1st to 2nd placement location | Rotate | Move from 7th position to 8th pick position | **max{table_movement(1,2), turret_rotation(2,3,4,5,6,7), feeder(7,8)}** |
| | | Pick 8th and place 2nd component | | **Pick&Place event** |
| 9 | Move from 2nd to 3rd placement location | Rotate | Move from 8th position to 9th pick position | **max{table_movement(2,3), turret_rotation(3,4,5,6,7,8), feeder(8,9)}** |
| | | Pick 9th and place 3rd component | | **Pick&Place event** |
| : | : | : | : | **:** |
| : | : | : | : | **:** |
| : | : | : | : | **:** |
| n | Move from (n-7)th to (n-6)th placement location | Rotate | Move from (n-1)th position to (n)th pick position | **max{table_movement(n-7,n-6), turret_rotation(n-6,n-5,n-4,n-3,n-2,n-1), feeder(n-1,n)}** |
| | | Pick (n)th and place (n-6)th component | | **Pick&Place event** |
| n+1 | Move from (n-6)th to (n-5)th placement location | Rotate | Move from (n)th position to 1st pick position | **max{table_movement(n-6,n-5), turret_rotation(n-5,n-4,n-3,n-2,n-1,n), feeder(n,1)}** |
| | | Pick 1st and place (n-5)th component | | **Pick&Place event** |
| n+2 | Move from (n-5)th to (n-4)th placement location | Rotate | Move from 1st position to 2nd pick position | **max{table_movement(n-5,n-4), turret_rotation(n-4,n-3,n-2,n-1,n,1), feeder(1,2)}** |
| | | Pick 2nd and place (n-4)th component | | **Pick&Place event** |
| n+3 | Move from (n-4)th to (n-3)rd placement location | Rotate | Move from 2nd position to 3rd pick position | **max{table_movement(n-4,n-3), turret_rotation(n-3,n-2,n-1,n,1,2), feeder(2,3)}** |
| | | Pick 3rd and place (n-3)rd component | | **Pick&Place event** |
| n+4 | Move from (n-3)rd to (n-2)nd placement location | Rotate | Move from 3th position to 4th pick position | **max{table_movement(n-3,n-2), turret_rotation(n-2,n-1,n,1,2,3), feeder(3,4)}** |
| | | Pick 4th and place (n-2)nd component | | **Pick&Place event** |
| n+5 | Move from (n-2)nd to (n-1)st placement location | Rotate | Move from 4th position to 5th pick position | **max{table_movement(n-2,n-1), turret_rotation(n-1,n,1,2,3,4), feeder(4,5)}** |
| | | Pick 5th and place (n-1)st component | | **Pick&Place event** |
| n+6 | Move from (n-1)st to (n)th placement location | Rotate | Move from 5th position to 6th pick position | **max{table_movement(n-1,n), turret_rotation(n,1,2,3,4,5), feeder(5,6)}** |
| | | Pick 6th and place (n)th component | | **Pick&Place event** |
| ASSEMBLED PCB IS UNLOADED NEW PCB IS LOADED FROM THE CONVEYOR AND PLACED OVER THE PCB TABLE | | | | |
| 1 | Move from (n)th table position to 1st placement location | Rotate | Move from 6th position to 7th pick position | **max{table_movement(n,1), turret_rotation(1,2,3,4,5,6), feeder(6,7)}** |
| | | Pick 7th and place 1st component | | **Pick&Place event** |
| 2 | Move from 1st to 2nd placement location | Rotate | Move from 7th position to 8th pick position | max{table_movement(1,2), turret_rotation(2,3,4,5,6,7), feeder(7,8)} |
| | | Pick 8th and place 2nd component | | Pick&Place event |
| 3 | Move from 2nd to 3rd placement location | Rotate | Move from 8th position to 9th pick position | max{table_movement(2,3), turret_rotation(3,4,5,6,7,8), feeder(8,9)} |
| | | Pick 9th and place 3rd component | | Pick&Place event |
| : | : | : | : | : |
| : | : | : | : | : |
| : | : | : | : | : |
| LOOP UNTIL PRODUCTION OF PCB TYPE STOPS | | | | |

# APPENDIX B

# MATHEMATICAL MODEL AND RESULTS OF A REAL PCB

The PCB locations, coordinates of these locations, component types of the locations and turret rotation time for each location regarding its component type is given in Table B.1. Table movement distance matrix between the locations is given in Table B.2. Feeder distances are given in Table B.3.

**Table B.1. Location – component type data of the example PCB**

| $l$ | $X_l$ | $Y_l$ | $c_l$ | $r_{c_l}$ (msec) |
|---|---|---|---|---|
| 1 | 134880 | -435780 | 1 | 271 |
| 2 | 115860 | -450570 | 1 | 271 |
| 3 | 173660 | -439350 | 2 | 500 |
| 4 | 168570 | -439350 | 2 | 500 |
| 5 | 163470 | -439350 | 2 | 500 |
| 6 | 158380 | -439350 | 2 | 500 |
| 7 | 92240 | -447060 | 2 | 500 |
| 8 | 87270 | -447060 | 2 | 500 |
| 9 | 82180 | -447060 | 2 | 500 |
| 10 | 77100 | -447060 | 2 | 500 |
| 11 | 132630 | -443220 | 3 | 157 |
| 12 | 118130 | -443210 | 3 | 157 |

**Table B.2. Feeder distances between component types (msec)**

| $i$ \ $j$ | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 250 | 150 |
| 2 | 250 | 0 | 100 |
| 3 | 150 | 100 | 0 |

**Table B.3. Table movement time between locations (msec)**

| $k$<br>$l$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 451 | 565 | 497 | 429 | 361 | 719 | 785 | 853 | 921 | 129 | 322 |
| 2 | 451 | 0 | 920 | 852 | 784 | 717 | 362 | 428 | 496 | 564 | 322 | 128 |
| 3 | 565 | 920 | 0 | 68 | 136 | 204 | 1188 | 1255 | 1323 | 1390 | 599 | 792 |
| 4 | 497 | 852 | 68 | 0 | 68 | 136 | 1121 | 1187 | 1255 | 1322 | 531 | 724 |
| 5 | 429 | 784 | 136 | 68 | 0 | 68 | 1053 | 1119 | 1187 | 1254 | 463 | 656 |
| 6 | 361 | 717 | 204 | 136 | 68 | 0 | 985 | 1051 | 1119 | 1187 | 395 | 588 |
| 7 | 719 | 362 | 1188 | 1121 | 1053 | 985 | 0 | 66 | 134 | 202 | 590 | 397 |
| 8 | 785 | 428 | 1255 | 1187 | 1119 | 1051 | 66 | 0 | 68 | 136 | 656 | 463 |
| 9 | 853 | 496 | 1323 | 1255 | 1187 | 1119 | 134 | 68 | 0 | 68 | 724 | 531 |
| 10 | 921 | 564 | 1390 | 1322 | 1254 | 1187 | 202 | 136 | 68 | 0 | 792 | 598 |
| 11 | 129 | 322 | 599 | 531 | 463 | 395 | 590 | 656 | 724 | 792 | 0 | 193 |
| 12 | 322 | 128 | 792 | 724 | 656 | 588 | 397 | 463 | 531 | 598 | 193 | 0 |

Given data in the tables above, results of the linear model as an output of CPLEX, is shown in Figure B.1. Notations used in the CPLEX model are: “XsXlXk” is $\alpha_{slk}$; “YsYiYj” is $\beta_{sij}$; “ZiZs” is $z_{is}$ (since number of feeders equal to number of components $z_{is}$ can be used instead of $z_{c_l s}$).

Figure B.1. Output of the linear model

```
Integer optimal solution:  Objective =   5.7230000000e+003
Solution time = 20623.82 sec.  Iterations = 21666232  Nodes = 225215


Variable Name           Solution Value
T1                          500.000000
T2                          500.000000
T3                          500.000000
T4                          362.000000
T5                          500.000000
T6                          500.000000€
T7                          500.000000
T8                          500.000000
T9                          500.000000
T10                         361.000000
T11                         500.000000
T12                         500.000000
X1X3X5                       -0.000000
X1X5X3                        0.000000
X1X8X9                        0.000000
X1X8X10                       1.000000
X2X5X3                       -0.000000
X2X9X3                        0.000000
X2X10X9                       1.000000
```

```
X3X9X7                        1.000000
X4X7X2                        1.000000
X5X2X11                       1.000000
X6X11X5                       1.000000
X7X5X3                        1.000000
X8X1X10                       0.000000
X8X3X4                        1.000000
X8X3X9                       -0.000000
X9X4X6                        1.000000
X9X9X7                       -0.000000
X9X10X7                       0.000000
X10X6X1                       1.000000
X11X1X12                      1.000000
X12X11X3                     -0.000000
X12X11X5                      0.000000
X12X12X8                      1.000000
Y1Y2Y2                        1.000000
Y2Y2Y1                        1.000000
Y3Y1Y3                        1.000000
Y4Y3Y2                        1.000000
Y5Y2Y2                        1.000000
Y6Y1Y1                       -0.000000
Y6Y1Y2                        0.000000
Y6Y2Y2                        1.000000
Y7Y2Y2                        1.000000
Y8Y2Y1                        1.000000
Y9Y1Y3                        1.000000
Y10Y3Y2                       1.000000
Y11Y2Y2                       1.000000
Y12Y2Y2                       1.000000
Z2Z1                          1.000000
Z1Z2                          1.000000
Z2Z2                          1.000000
Z3Z2                          1.000000
Z1Z3                          1.000000
Z2Z3                          1.000000
Z3Z3                          1.000000
Z1Z4                          1.000000
Z3Z4                          1.000000
Z1Z5                          1.000000
Z2Z5                          1.000000
Z3Z5                          1.000000
Z1Z6                          1.000000
Z2Z6                          1.000000
Z3Z6                          1.000000
Z2Z7                          1.000000
Z2Z8                          1.000000
Z1Z9                          1.000000
Z2Z9                          1.000000
Z3Z9                          1.000000
Z1Z10                         1.000000
Z3Z10                         1.000000
Z1Z11                         1.000000
Z2Z11                         1.000000
Z3Z11                         1.000000
Z2Z12                         1.000000
All other variables in the range 1-1740 are zero.
```

# APPENDIX C

# ALGORITHM USED IN HEURISTIC 1

The algorithm used have been divided into 4 groups:

1. PCB is partitioned into component vs. the positions associated.
    a. Naming the sets for the components. A component set involves positions that are of same type.
    b. Location matrix for each set is derived.
2. Sending partitions for TSP solutions:
    a. Each set with location matrix are sent to CONCORDE TSP Package to receive solution for each component type
    b. The result of each component is transformed to be used in tour structure.
3. Components are combined:
    a. For each component set using an element location in the set; distance function between another component set is calculated.
    b. The minimum distance result gives the components to be combined.
    c. The component sets to be combined is decreased by 1.
    d. Go to step (a) till no component set exists to be combined.
4. Tour length calculation and 2-opt
    a. Calculate tour length via clockwise and counter-clockwise
    b. Select the one which gives better result.
    c. Starting from the tourpointer, apply 2-opt. Break 2 arcs form old tour. Form new tour, calculate the gain between new and old tour.
    d. Find the minimum gain regarding the broken locations in the tour
    e. If gain <0 ; form the new tour as found in 2-opt.Go to a. If gain >0 then stop.

## COMPONENT TYPE vs. LOCATION

$i$ : location index for PCB

$j$ : location index for PCB (alias of $i$)

$t$ : component type index

$N$ : Location set (i.e. $i \in N$) $N = \{1, 2, ..., n\}$

$M$ : Component type set (i.e. $t \in M$) $M = \{1, 2, ..., m\}$

$C(i)$ : Component type of a location $i$ $C(i) : N \to M$

$G(t)$ : Set of component type $t$ where locations' component type is $t$ (i.e. $i \in G(t) \leftrightarrow C(i) = t$)

$len(i, j)$ : Travel time from location $i$ to location $j$

$LOC(N)$ : Location matrix for all locations for placement (i.e. $LOC(N)$ is an $n \times n$ matrix)

$LOC(G(t))$ : Location matrix for $G(t)$

$tpnt$ : Tour pointer array for component type.

$right$ : Array to describe the right of a location in a sequence found

$left$ : Array to describe the right of a location in a sequence found

## INITIALIZATION

```
$\forall t \in M$  set  $G(t) = \{\}$
$\forall i \in N$   $C(i) = t \Rightarrow G(t) = G(t) \bigcup \{i\}$
$\forall t \in M$
        $\forall i \in G(t)$
                $\forall j \in G(t)$
                        $LOC(G(t)) = LOC(G(t)) \bigcup len\{i, j\}$
```

```
For $t = 1$ to m
    Run TSP ( $LOC(G(t))$ )
    CALL PROCEDURE ARR_TSP( $G(t)$ )
Next $t$
```

Using output of TSP file, string functions (LEN, LEFT, RIGHT, etc) are used to find arrays

**PROCEDURE ARR_TSP( $G(t)$ )**

```
    Set $tpnt\,(t)$
    For each $i \in G(t)$
        Set $right\,(i)$
        Set $left\,(i)$
    Next $i$
```

ASSUMING THE TOUR IS **CLOCKWISE**

```
MAXGAIN $= -999999$
GAIN1CUT $\leftarrow$ tourpnt
GAIN2CUT $\leftarrow$ tourpnt
$i \leftarrow$ tourpnt
    while right $(i) \neq$ tourpnt
        $j \leftarrow$ right.right$(i)$
            while $j \neq$ left $(i)$
                NTC = 0
                FIRSTCAL $(i, j, NTC)$
```

```
                    SECONDCAL $(i, j, NTC)$
                    If MAXGAIN $\leq$ NTC then
                        MAXGAIN $\leftarrow$ NTC
                        GAIN1CUT $\leftarrow i$
                        GAIN2CUT $\leftarrow j$
                    End if
                $j \leftarrow$ right $(j)$
        $i \leftarrow$ right $(i)$
```

PROCEDURE FIRSTCAL $(i, j, NTC)$: Calculates the net gain for **<u>Counter-clockwise tour</u>**:

NTC : Net gain found as a result of the new tour

NT : Array for the following 6 locations in a tour beginning with a given location

NT.feeder: Array holding the feeder movement cost of a location in the new tour

NT.turret: Array holding the turret movement cost of a location in the new tour

$p$ : Array index

*row* : Integer to record calculation sequence

*opt* : Pointer used to fill the array

*opttourptr* : Variable used for assignment of the location in the 2-opt sequence

### PROCEDURE FIRSTCAL $(i, j, NTC)$

```
NTC = 0
$opt \leftarrow$ right $(j)$
For $p = 1$ to 6
    NT $(p) \leftarrow opt$
    $opt \leftarrow$ right $(opt)$
Next p
$opttourptr \leftarrow$ right $(i)$
$row = 1$
```

```
While $opttourptr \neq j$
        If $row \leq 6$
                NT.feeder$(opttourptr)$ = FEEDERDISTANCE ( component
                ( NT (6) ), component ( NT (5) )
                NT.turret$(opttourptr)$ = component $(opttourptr)$.turret_min
                For $p = 1$ to 5
                        If NT.turret$(opttourptr)$ < component ( NT $(p)$ ).turret_min
                        NT.turret$(opttourptr)$ = component ( NT $(p)$ ).turret_min
                Next p
                NT (6) = NT (5)
                NT (5) = NT (4)
                NT (4) = NT (3)
                NT (3) = NT (2)
                NT (2) = NT (1)
                NT (1) = $opttourptr$
        End If
        NTC = NTC + [max {NT.turret$(opttourptr)$; NT.feeder$(opttourptr)$;
        DISTANCE (location$(opttourptr)$, location(right$(opttourptr)$))}- max
        {CW.turret$(opttourptr)$; CW.feeder$(opttourptr)$;
        DISTANCE(location$(opttourptr)$, location (right$(opttourptr)$))}]
$opttourptr \leftarrow$ right$(opttourptr)$
```

PROCEDURE SECONDCAL $(i, j, NTC)$: Calculates the net gain for **clockwise tour where the cost changes before the first cut**:

NTC : Net gain found as a result of the new tour

NT : Array for the following 6 locations in a tour beginning with a given location

NT.feeder: Array holding the feeder movement cost of a location in the new tour

NT.turret: Array holding the turret movement cost of a location in the new tour

$p$ : Array index

*opt* : Pointer used to fill the array

*opttourptr* : Variable used for assignment of the location in the 2-opt sequence

```
$opt \leftarrow j$
For $p = 1$ to 6
    NT $(p) \leftarrow opt$
    $opt \leftarrow$ left $(opt)$
Next p

$opttourptr \leftarrow i$
$row = 1$
While $row < 6$

    NT.feeder$(opttourptr)$ = FEEDERDISTANCE ( component ( NT (6) ), component ( NT (5) )
    NT.turret$(opttourptr)$ = component $(opttourptr)$.turret_min
    For $p = 1$ to 5
        If NT.turret$(opttourptr)$ < component ( NT $(p)$ ).turret_min
            NT.turret$(opttourptr)$ = component ( NT $(p)$ ).turret_min
    Next p
    NT (6) = NT (5)
    NT (5) = NT (4)
    NT (4) = NT (3)
    NT (3) = NT (2)
    NT (2) = NT (1)
    NT (1) = $opttourptr$
NTC = NTC + [max {NT.turret$(opttourptr)$; NT.feeder$(opttourptr)$; DISTANCE (location$(opttourptr)$, location(right$(opttourptr)$))}- max {CW.turret$(opttourptr)$;
```

```
CW.feeder($(opttourptr)$); DISTANCE(location$(opttourptr)$, location(right$(opttourptr)$))}]
$opttourptr \leftarrow$ left $(opttourptr)$
$row \leftarrow (row+1)$
```

# APPENDIX D

# COMBINING SUBTOURS

Starting with the location 20 in the constructed tour, the steps of the algorithm are explained. Any location can be taken as a starting point since the tour cycle will not be effected. Initially the PCB table is at a park point, but in mass manufacturing, steps will follow each other without visiting this park point. The first six steps is devoted to feeding the turret heads with components. As the first placement occurs after table movement from the initial placement, the cycle length of the construction tour with clockwise direction will be calculated from placement of location 21 to the next placement of location 21.

Table location is the coordinate to which the table will move to place component. Feeder location shows the coordinate to which the component type will be picked. Since the feeder carrier should move six steps ahead the placement, feeder location is represented as $(i+6)^{\text{th}}$ step. Movement time ($mt(l,k)$) is determined from the distance from location $l$ to $k$, and feeder time ($ft(c(l),c(k))$) is determined from the distance between feeder rack where component of location $l$ is placed and feeder rack where component of location $k$ is placed. Turret movement time is minimum time of the turret heads rotating with components before placement occurs. Rotation of a head depends on the turret rotation index of the component that head carries, thus $tr(c(l))$ is the time required for component type of location $l$, to rotate 1 turn, i.e. 30 degrees.

## Table D.1. Concurrent operations for the PCB and cycle time calculations for clockwise direction

| | Motion | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Table loc (i) | Feeder Loc (i+6) | Table move (i) | | Turret Movement_Clockwise(i) | | Feeder carrier move | | Distance (travelling time for step i) max (table, turret, feeder) |
| | 20 | No movement | | No movement | | ft(0,174) | | |
| | 21 | No movement | | tr(HDRN00160-449) | | ft(174,174) | | |
| | 22 | No movement | | max[tr(HDRN00160-449), tr(HDRN00160-449)] | | ft(174,174) | | |
| | 26 | No movement | | max[tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449)] | | ft(174,174) | | |
| | 25 | No movement | | max[tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449)] | | ft(174,174) | | |
| | 24 | No movement | | max[tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449)] | | ft(174,174) | | |
| 20 | 23 | mt(0,20) | | max[tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449)] | | ft(174,174) | | |
| 21 | 19 | mt(20,21) | 3658 | max[tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449)] | 157 | ft(174,174) | 0 | 3658 |
| 22 | 27 | mt(21,22) | 169 | max[tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449)] | 157 | ft(174,212) | 57000 | 57000 |
| 26 | 28 | mt(22,26) | 8162 | max[tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00291)] | 271 | ft(212,212) | 0 | 8162 |
| 25 | 29 | mt(26,25) | 169 | max[tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00291), tr(HDRN00291)] | 271 | ft(212,212) | 0 | 271 |
| 24 | 30 | mt(25,24) | 3658 | max[tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00291), tr(HDRN00291), tr(HDRN00291)] | 271 | ft(212,212) | 0 | 3658 |
| 23 | 16 | mt(24,23) | 169 | max[tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00291), tr(HDRN00291), tr(HDRN00291), tr(HDRN00291)] | 271 | ft(212,74) | 207000 | 207000 |
| 19 | 17 | mt(23,19) | 8162 | max[tr(HDRN00160-449), tr(HDRN00291), tr(HDRN00291), tr(HDRN00291), tr(HDRN00291), tr(HDRN00180)] | 271 | ft(74,74) | 0 | 8162 |
| 27 | 18 | mt(19,27) | 211 | max[tr(HDRN00291), tr(HDRN00291), tr(HDRN00291), tr(HDRN00291), tr(HDRN00180), tr(HDRN00180)] | 271 | ft(74,74) | 0 | 271 |
| 28 | 12 | mt(27,28) | 3827 | max[tr(HDRN00291), tr(HDRN00291), tr(HDRN00291), tr(HDRN00180), tr(HDRN00180), tr(HDRN00180)] | 271 | ft(74,10) | 96000 | 96000 |
| 29 | 14 | mt(28,29) | 8162 | max[tr(HDRN00291), tr(HDRN00291), tr(HDRN00180), tr(HDRN00180), tr(HDRN00180), tr(HDRN00162)] | 271 | ft(10,10) | 0 | 8162 |
| 30 | 13 | mt(29,30) | 3827 | max[tr(HDRN00291), tr(HDRN00180), tr(HDRN00180), tr(HDRN00180), tr(HDRN00162), tr(HDRN00162)] | 271 | ft(10,10) | 0 | 3827 |
| 16 | 7 | mt(30,16) | 3657 | max[tr(HDRN00180), tr(HDRN00180), tr(HDRN00180), tr(HDRN00162), tr(HDRN00162), tr(HDRN00162)] | 271 | ft(10,9) | 1500 | 3657 |
| 17 | 8 | mt(16,17) | 8162 | max[tr(HDRN00180), tr(HDRN00180), tr(HDRN00162), tr(HDRN00162), tr(HDRN00162), tr(HDRN00169)] | 271 | ft(9,9) | 0 | 8162 |
| 18 | 6 | mt(17,18) | 3827 | max[tr(HDRN00180), tr(HDRN00162), tr(HDRN00162), tr(HDRN00162), tr(HDRN00169), tr(HDRN00169)] | 271 | ft(9,6) | 4500 | 4500 |
| 12 | 4 | mt(18,12) | 4505 | max[tr(HDRN00162), tr(HDRN00162), tr(HDRN00162), tr(HDRN00169), tr(HDRN00169), tr(HKPS00089)] | 271 | ft(6,6) | 0 | 4505 |
| 14 | 31 | mt(12,14) | 8162 | max[tr(HDRN00162), tr(HDRN00162), tr(HDRN00169), tr(HDRN00169), tr(HKPS00089), tr(HKPS00089)] | 271 | ft(6,2) | 6000 | 8162 |
| 13 | 32 | mt(14,13) | 3827 | max[tr(HDRN00162), tr(HDRN00169), tr(HDRN00169), tr(HKPS00089), tr(HKPS00089), tr(HYRL00146)] | 386 | ft(2,2) | 0 | 3827 |
| 7 | 34 | mt(13,7) | 3997 | max[tr(HDRN00169), tr(HDRN00169), tr(HKPS00089), tr(HKPS00089), tr(HYRL00146), tr(HYRL00146)] | 386 | ft(2,2) | 0 | 3997 |
| 8 | 33 | mt(7,8) | 3827 | max[tr(HDRN00169), tr(HKPS00089), tr(HKPS00089), tr(HYRL00146), tr(HYRL00146), tr(HYRL00146)] | 386 | ft(2,2) | 0 | 3827 |
| 6 | 3 | mt(8,6) | 3361 | max[tr(HKPS00089), tr(HKPS00089), tr(HYRL00146), tr(HYRL00146), tr(HYRL00146), tr(HYRL00146)] | 386 | ft(2,6) | 6000 | 6000 |
| 4 | 5 | mt(6,4) | 3827 | max[tr(HKPS00089), tr(HYRL00146), tr(HYRL00146), tr(HYRL00146), tr(HYRL00146), tr(HKPS00089)] | 386 | ft(6,6) | 0 | 3827 |
| 31 | 9 | mt(4,31) | 4885 | max[tr(HYRL00146), tr(HYRL00146), tr(HYRL00146), tr(HYRL00146), tr(HKPS00089), tr(HKPS00089)] | 386 | ft(6,9) | 4500 | 4885 |
| 32 | 10 | mt(31,32) | 3827 | max[tr(HYRL00146), tr(HYRL00146), tr(HYRL00146), tr(HKPS00089), tr(HKPS00089), tr(HDRN000169)] | 386 | ft(9,9) | 0 | 3827 |
| 34 | 11 | mt(32,34) | 8161 | max[tr(HYRL00146), tr(HYRL00146), tr(HKPS00089), tr(HKPS00089), tr(HDRN000169), tr(HYRL00169)] | 386 | ft(9,10) | 1500 | 8161 |
| 33 | 18 | mt(32,33) | 3827 | max[tr(HYRL00146), tr(HKPS00089), tr(HKPS00089), tr(HDRN000169), tr(HYRL00169), tr(HDRN00162)] | 271 | ft(10,74) | 96000 | 96000 |
| 3 | 20 | mt(33,3) | 4889 | max[tr(HKPS00089), tr(HKPS00089), tr(HDRN000169), tr(HYRL00169), tr(HDRN00162), tr(HDRN00180)] | 271 | ft(74,174) | 150000 | 150000 |
| 5 | 21 | mt(3,5) | 3830 | max[tr(HKPS00089), tr(HDRN000169), tr(HDRN00169), tr(HDRN00162), tr(HDRN00180), tr(HDRN000160-449)] | 271 | ft(174,174) | 0 | 3830 |
| 9 | 22 | mt(5,9) | 3361 | max[tr(HDRN000169), tr(HDRN00169), tr(HDRN00162), tr(HDRN00180), tr(HDRN000160-449), tr(HDRN000160-449)] | 271 | ft(174,174) | 0 | 3361 |
| 10 | 26 | mt(9,10) | 3827 | max[tr(HDRN000169), tr(HDRN00162), tr(HDRN00180), tr(HDRN000160-449), tr(HDRN000160-449), tr(HDRN000160-449)] | 271 | ft(174,174) | 0 | 3827 |
| 11 | 25 | mt(10,11) | 3997 | max[tr(HDRN00162), tr(HDRN00180), tr(HDRN000160-449), tr(HDRN000160-449), tr(HDRN000160-449), tr(HDRN000160-449)] | 271 | ft(174,174) | 0 | 3997 |
| 15 | 24 | mt(11,15) | 8839 | max[tr(HDRN00180), tr(HDRN000160-449),tr(HDRN000160-449),tr(HDRN000160-449),tr(HDRN000160-449), tr(HDRN000160-449)] | 157 | ft(174,174) | 0 | 8839 |
| 20 | 23 | mt(18,20) | 8712 | max[tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449)] | 157 | ft(174,174) | 0 | 8712 |
| 21 | 19 | mt(20,21) | | max[tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449)] | | ft(174,174) | | |
| | | | | | | **TOTAL** | | **742074** |

The costs incurring at step $i$ can be formulated as follows:

$\text{cost}(i) = \max\{$Table movement at step $i$, Turret movement at step $i$, Feeder movement at step $i\}$

Using this step info in terms of tour functions as left() and right(), which is explained previously, cost associated to a location may be defined as follows:

$$\text{cl\_cost(i)} = \max\{table\_move(i), turret\_move\_clockwise(i), feeder\_move(i)\} \quad \text{where}$$

$$cl\_table\_move(i) = mt(left(i), i) \tag{1}$$

$$cl\_turret\_move(i) = \max\begin{bmatrix} tr[c\langle right(right(right(right(right(i)))))\rangle], \\ tr[c\langle right(right(right(right(i))))\rangle], \\ tr[c\langle right(right(right(i)))\rangle], \\ tr[c\langle right(right(i))\rangle], \\ tr[c\langle right(i)\rangle], \\ tr[c\langle i\rangle] \end{bmatrix} \tag{2}$$

$$cl\_feeder\_move(i) = ft(fl[c\langle right(right(right(right(right(i))))))\rangle], fl[c\langle right(right(right(right(right(right(i))))))\rangle]) \tag{3}$$

In equation (1), the table's movement from the last placed location to the new location is defined. Equation (2) defines the turret movement as the rotation of the turret which depends on the minimum turret rotation speed of turret heads carrying components. As each component type has a turret rotation index (i.e. tr(c(i), where c(i) is the component type of location i) shows the turret rotation index of component type of location i), meaning that each component type can rotate at a maximum allowable speed, the turret's one turn cannot exceed this allowable speed, i.e. turret's one turn is minimum of the components' (that are already picked and carried by turret heads) maximum allowable speed. Over the turret, in the opposite direction of the placement station, there is a picking station. Turret head which is at this picking station, is fed by the component type that would be placed after "turret head amount"/2 steps. Feeding the turret head requires the movement of the feeder carrier unless component type of location that would be placed after "turret head amount"/2 steps is same with the component that will be placed "turret head amount"/2 – 1. If the component type of location that would be placed after "turret head amount"/2 steps is not same with the one to be placed "turret head amount"/2 – 1 steps after; a feeder carrier movement, i.e. ft(fl(c(i)),fl(c(j))) is the feeder time that occurs if the carrier moves from feeder location reserved for the component of location i to feeder location reserved for the component of location j, which is described by Equation 3 will be realized.

The tour presented in the Figure 3.6 and the time table (Table D.1) is compiled base on the clockwise direction, the counter clockwise direction will also be examined to check whether the cycle time is better than that of clockwise direction. It should also be noted that calculations regarding the turret movement will be used in the 2-opt improvement part of the solution approach. Figure D.1 and Table D.2 are showing the cycle and the calculations regarding the counter clockwise direction. For counter clockwise direction, instead of building the tour structure again, costs for each step will be calculated using the cycle information developed for clockwise direction such that:

$$\text{cc\_cost(i)} = \max\{cc\_table\_move(i), cc\_turret\_move(i), cc\_feeder\_move(i)\} \quad \text{where}$$

$$cc\_table\_move(i) = mt(right(i), i) \tag{4}$$

$$cc\_turret\_move(i) = \max\begin{bmatrix} tr[c\langle left(left(left(left(left(i)))))\rangle], \\ tr[c\langle left(left(left(left(i))))\rangle], \\ tr[c\langle left(left(left(i)))\rangle], \\ tr[c\langle left(left(i))\rangle], \\ tr[c\langle left(i)\rangle], \\ tr[c\langle i\rangle] \end{bmatrix} \tag{5}$$

$$cc\_feeder\_move(i) = ft(fl[c\langle left(left(left(left(left(i)))))\rangle], fl[c\langle left(left(left(left(left(left(i))))))\rangle]) \tag{6}$$

Equations (4), (5) and (6) are substitutes of (1), (2) and (3) respectively.

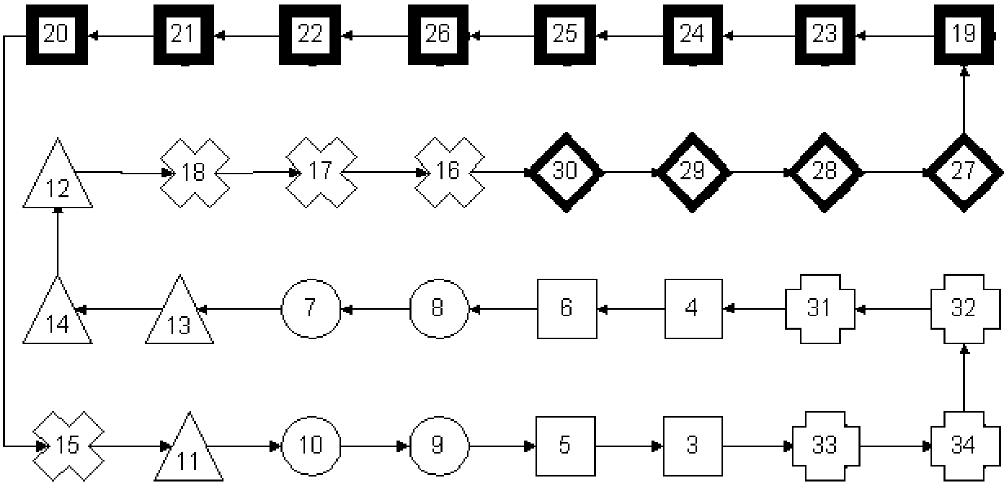


Figure D.1 Constructed tour with counter clockwise direction

## Table D.2. Concurrent operations for the example PCB and cycle time calculations for counter clockwise direction

Motion

| Table loc (i) | Feeder Lo (i+6) | Table move (i) | | Turret Movement_Clockwise(i) | | Feeder carrier move | | Distance (travelling time fo step i) max (table, turret, feeder) |
|---|---|---|---|---|---|---|---|---|
| | 20 | No movement | | | | ft(0, 174) | | |
| | 15 | No movement | | | | ft(174,74) | | |
| | 11 | No movement | | | | ft(74,10) | | |
| | 10 | No movement | | | | ft(10,9) | | |
| | 9 | No movement | | | | ft(9,9) | | |
| | 5 | No movement | | | | ft(9,6) | | |
| 20 | 3 | mt(0,20) | | | | ft(6,6) | | |
| 15 | 33 | mt(20,15) | 8712 | max [tr(HDRN00180), tr(HDRN00162), tr(HDRN00169), tr(HDRN00169), tr(HKPS00089), tr(HKPS00089)] | 271 | ft(6,2) | 6000 | 8712 |
| 11 | 34 | mt(15,11) | 8839 | max [tr(HDRN00162), tr(HDRN00169), tr(HDRN00169), tr(HKPS00089), tr(HKPS00089), tr(HYRL00146)] | 386 | ft(2,2) | 0 | 8839 |
| 10 | 32 | mt(11,10) | 3997 | max [tr(HDRN00169), tr(HDRN00169), tr(HKPS00089), tr(HKPS00089), tr(HYRL00146), tr(HYRL00146)] | 386 | ft(2,2) | 0 | 3997 |
| 9 | 31 | mt(10,9) | 3827 | max [tr(HDRN00169), tr(HKPS00089), tr(HKPS00089), tr(HYRL00146), tr(HYRL00146), tr(HYRL00146)] | 386 | ft(2,2) | 0 | 3827 |
| 5 | 4 | mt(9,5) | 3361 | max [tr(HKPS00089), tr(HKPS00089), tr(HYRL00146), tr(HYRL00146), tr(HYRL00146), tr(HYRL00146)] | 386 | ft(2,6) | 6000 | 6000 |
| 3 | 6 | mt(5,3) | 3830 | max [tr(HKPS00089), tr(HYRL00146), tr(HYRL00146), tr(HYRL00146), tr(HYRL00146), tr(HKPS00089)] | 386 | ft(6,6) | 0 | 3830 |
| 33 | 8 | mt(3,33) | 4889 | max [tr(HYRL00146), tr(HYRL00146), tr(HYRL00146), tr(HYRL00146), tr(HKPS00089), tr(HKPS00089)] | 386 | ft(6,9) | 4500 | 4889 |
| 34 | 7 | mt(33,34) | 3827 | max [tr(HYRL00146), tr(HYRL00146), tr(HYRL00146), tr(HKPS00089), tr(HKPS00089), tr(HDRN00169)] | 386 | ft(9,9) | 0 | 3827 |
| 32 | 13 | mt(34,32) | 8161 | max [tr(HYRL00146), tr(HYRL00146), tr(HKPS00089), tr(HKPS00089), tr(HDRN00169), tr(HDRN00169)] | 386 | ft(9,10) | 1500 | 8161 |
| 31 | 14 | mt(32,31) | 3827 | max [tr(HYRL00146), tr(HKPS00089), tr(HKPS00089), tr(HDRN00169), tr(HDRN00169), tr(HDRN00162)] | 386 | ft(10,10) | 0 | 3827 |
| 4 | 12 | mt(31,4) | 4885 | max [tr(HKPS00089), tr(HKPS00089), tr(HDRN00169), tr(HDRN00169), tr(HDRN00162), tr(HDRN00162)] | 271 | ft(10,10) | 0 | 4885 |
| 6 | 18 | mt(4,6) | 3827 | max [tr(HKPS00089), tr(HDRN00169), tr(HDRN00169), tr(HDRN00162), tr(HDRN00162), tr(HDRN00162)] | 271 | ft(10,74) | 96000 | 96000 |
| 8 | 17 | mt(6,8) | 3361 | max [tr(HDRN00169), tr(HDRN00169), tr(HDRN00162), tr(HDRN00162), tr(HDRN00162), tr(HDRN00180)] | 271 | ft(74,74) | 0 | 3361 |
| 7 | 16 | mt(8,7) | 3827 | max [tr(HDRN00169), tr(HDRN00162), tr(HDRN00162), tr(HDRN00162), tr(HDRN00180), tr(HDRN00180)] | 271 | ft(74,74) | 0 | 3827 |
| 13 | 30 | mt(7,13) | 3997 | max [tr(HDRN00162), tr(HDRN00162), tr(HDRN00162), tr(HDRN00180), tr(HDRN00180), tr(HDRN00180)] | 271 | ft(74,212) | 207000 | 207000 |
| 14 | 29 | mt(13,14) | 3827 | max [tr(HDRN00162), tr(HDRN00162), tr(HDRN00180), tr(HDRN00180), tr(HDRN00180), tr(HDRN00291)] | 271 | ft(212,212) | 0 | 3827 |
| 12 | 28 | mt(14,12) | 8162 | max [tr(HDRN00162), tr(HDRN00180), tr(HDRN00180), tr(HDRN00180), tr(HDRN00291), tr(HDRN00291)] | 271 | ft(212,212) | 0 | 8162 |
| 18 | 27 | mt(12,18) | 4505 | max [tr(HDRN00180), tr(HDRN00180), tr(HDRN00180), tr(HDRN00291), tr(HDRN00291), tr(HDRN00291)] | 271 | ft(212,212) | 0 | 4505 |
| 17 | 19 | mt(18,17) | 3827 | max [tr(HDRN00180), tr(HDRN00180), tr(HDRN00291), tr(HDRN00291), tr(HDRN00291), tr(HDRN00291)] | 271 | ft(212,174) | 57000 | 57000 |
| 16 | 23 | mt(17,16) | 8162 | max [tr(HDRN00180), tr(HDRN00291), tr(HDRN00291), tr(HDRN00291), tr(HDRN00291), tr(HDRN00160-449)] | 271 | ft(174,174) | 0 | 8162 |
| 30 | 24 | mt(16,30) | 3657 | max [tr(HDRN00291), tr(HDRN00291), tr(HDRN00291), tr(HDRN00291), tr(HDRN00160-449), tr(HDRN00160-449)] | 271 | ft(174,174) | 0 | 3657 |
| 29 | 25 | mt(30,29) | 3827 | max [tr(HDRN00291), tr(HDRN00291), tr(HDRN00291), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449)] | 271 | ft(174,174) | 0 | 3827 |
| 28 | 26 | mt(29,28) | 8162 | max [tr(HDRN00291), tr(HDRN00291), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449)] | 271 | ft(174,174) | 0 | 8162 |
| 27 | 22 | mt(28,27) | 3827 | max [tr(HDRN00291), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449)] | 271 | ft(174,174) | 0 | 3827 |
| 19 | 21 | mt(27,19) | 211 | max [tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449)] | 157 | ft(174,174) | 0 | 211 |
| 23 | 20 | mt(19,23) | 8162 | max [tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449)] | 157 | ft(174,174) | 0 | 8162 |
| 24 | 18 | mt(23,24) | 169 | max [tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449)] | 157 | ft(174,74) | 150000 | 150000 |
| 25 | 11 | mt(24,25) | 3658 | max [tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00180)] | 271 | ft(74,10) | 96000 | 96000 |
| 26 | 10 | mt(25,26) | 169 | max [tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00180), tr(HDRN00162)] | 271 | ft(10,9) | 1500 | 1500 |
| 22 | 9 | mt(26,22) | 8162 | max [tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00180), tr(HDRN00162), tr(HDRN00169)] | 271 | ft(9,9) | 0 | 8162 |
| 21 | 5 | mt(22,21) | 169 | max [tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00180), tr(HDRN00162), tr(HDRN00169), tr(HDRN00169)] | 271 | ft(9,6) | 4500 | 4500 |
| 20 | 3 | mt(21,20) | 3658 | max [tr(HDRN00160-449), tr(HDRN00180), tr(HDRN00162), tr(HDRN00169), tr(HDRN00169), tr(HKPS00089)] | 271 | ft(6,6) | 0 | 3658 |
| 15 | 33 | | | max [tr(HDRN00180), tr(HDRN00162), tr(HDRN00169), tr(HDRN00169), tr(HKPS00089), tr(HKPS00089)] | 271 | ft(6,6) | 0 | |
| | | | | | | TOTAL | | 744304 |

With the tour illustrated in Figure D.1 the SMT machine’s costs at each step is shown at Table D.2. Since clockwise directed tour gives better cycle time, construction tour will be directed clockwise. Modified 2-opt will be applied to this construction tour and improvement to the initial solution will be made.

# APPENDIX E

# GAIN CALCULATIONS

For the example given in Figure 3.6, calculations of part (1) is shown at Table F.1. Location 30 in the table is placed to show the costs are same, since the next six steps are same for both tours' structure. Calculations regarding the turret and feeder movements have to be conducted because of the change in the tour structure. The second part, as shown in Table E.2 shows that the 2-opt change in the tour has provided a subtour whose length can be calculated as a means of the counter clockwise costs. The third part requires one to calculate the costs of movements since the turret movement and feeder movement figures changed. At the end, it is found out that this sample 2-opt process improves the tour 100 seconds approximately.

It is important for one to define the calculation steps by using conditional statements, since there may not be 6 locations between locations right(j) and i, or the second part can not be applied in the case where the amount of locations between left(j) and i are less than 6.

## Table E.1. First part of the gain calculations

| Tabl e loc (i) | Feeder Loc (i+6) | New table move (i) | | Clockwise table move (i) | | New Turret Movement | | Clockwise Turret Movement | New Feeder carrier move | | Clockwise Feeder carrier move | | New distance (for step i) max (table, turret, feeder) | Old distance (for step i) max (table, turret, feeder) | Gain (difference between distances) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Motion | | | | | | | | | | | |
| : | : | : | : | : | : | : | : | : | : | : | : | : | : | : | : |
| : | : | : | : | : | : | : | : | : | : | : | : | : | : | : | : |
| 30 | 13 | mt(29,30) | 3827 | mt(29,30) | 3827 | max[tr(HDRN00291), tr(HDRN00180), tr(HDRN00180), tr(HDRN00180), tr(HDRN00162), tr(HDRN00162)] | 271 | 271 | ft(10,10) | 0 | ft(10,10) | 0 | 3827 | 3827 | 0 |
| 16 | 11 | mt(30,16) | 3657 | mt(30,16) | 3657 | max[tr(HDRN00180), tr(HDRN00180), tr(HDRN00180), tr(HDRN00162), tr(HDRN00162), tr(HDRN00162)] | 271 | 271 | ft(10,10) | 0 | ft(10,9) | 1500 | 3657 | 3657 | 0 |
| 17 | 10 | mt(16,17) | 8162 | mt(16,17) | 8162 | max[tr(HDRN00180), tr(HDRN00180), tr(HDRN00162), tr(HDRN00162), tr(HDRN00162), tr(HDRN00162)] | 271 | 271 | ft(10,9) | 1500 | ft(9,9) | 0 | 8162 | 8162 | 0 |
| 18 | 9 | mt(17,18) | 3827 | mt(17,18) | 3827 | max[tr(HDRN00180), tr(HDRN00162), tr(HDRN00162), tr(HDRN00162), tr(HDRN00162), tr(HDRN00169)] | 271 | 271 | ft(9,9) | 0 | ft(9,6) | 4500 | 3827 | 4500 | -673 |
| 12 | 5 | mt(18,12) | 4505 | mt(18,12) | 4505 | max[tr(HDRN00162), tr(HDRN00162), tr(HDRN00162), tr(HDRN00162), tr(HDRN00169), tr(HDRN00169)] | 271 | 271 | ft(9,6) | 4500 | ft(6,6) | 0 | 4505 | 4505 | 0 |
| 14 | 3 | mt(12,14) | 8162 | mt(12,14) | 8162 | max[tr(HDRN00162), tr(HDRN00162), tr(HDRN00162), tr(HDRN00169), tr(HDRN00169), tr(HKPS00089)] | 271 | 271 | ft(6,6) | 0 | ft(6,2) | 6000 | 8162 | 8162 | 0 |
| 13 | 33 | mt(14,13) | 3827 | mt(14,13) | 3827 | max[tr(HDRN00162), tr(HDRN00162), tr(HDRN00169), tr(HDRN00169), tr(HKPS00089), tr(HKPS00089)] | 271 | 386 | ft(6,2) | 6000 | ft(2,2) | 0 | 6000 | 3827 | 2173 |
| | | | | | | | | | | | | | | **total** | **1500** |

## Table E.2. Second  part of the gain

| Tabl e loc (i) | Feeder Loc (i+6) | New table move (i) | | Clockwise table move (i) | | New Turret Movement (i.e. counter clockwise turret movement) | | Clockwise Turret Movement | New Feeder carrier move | | Clockwise Feeder carrier move | | New distance (for step i) max (table, turret, feeder) | Old distance (for step i) max (table, turret, feeder) | Gain (difference between distances) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 32 | mt(11,10) | 3997 | mt(9,10) | 3827 | max[tr(HDRN00169), tr(HDRN00169), tr(HKPS00089), tr(HKPS00089), tr(HYRL00146), tr(HYRL00146)] | 386 | 271 | ft(2,2) | 0 | ft(174,174) | 0 | **3997** | 3827 | 170 |
| 9 | 31 | mt(10,9) | 3827 | mt(5,9) | 3361 | max[tr(HDRN00169), tr(HKPS00089), tr(HKPS00089), tr(HYRL00146), tr(HYRL00146), tr(HYRL00146)] | 386 | 271 | ft(2,2) | 0 | ft(174,174) | 0 | **3827** | 3361 | 466 |
| 5 | 4 | mt(9,5) | 3361 | mt(3,5) | 3830 | max[tr(HKPS00089), tr(HKPS00089), tr(HYRL00146), tr(HYRL00146), tr(HYRL00146), tr(HYRL00146)] | 386 | 271 | ft(2,6) | 6000 | ft(174,174) | 0 | **6000** | 3830 | 2170 |
| 3 | 6 | mt(5,3) | 3830 | mt(33,3) | 4889 | max[tr(HKPS00089), tr(HYRL00146), tr(HYRL00146), tr(HYRL00146), tr(HYRL00146), tr(HKPS00089)] | 386 | 271 | ft(6,6) | 0 | ft(74,174) | 150000 | **3830** | 150000 | -146170 |
| 33 | 8 | mt(3,33) | 4889 | mt(32,33) | 3827 | max[tr(HYRL00146), tr(HYRL00146), tr(HYRL00146), tr(HYRL00146), tr(HKPS00089), tr(HKPS00089)] | 386 | 271 | ft(6,9) | 4500 | ft(10,74) | 96000 | **4889** | 96000 | -91111 |
| 34 | 7 | mt(33,34) | 3827 | mt(32,34) | 8161 | max[tr(HYRL00146), tr(HYRL00146), tr(HYRL00146), tr(HKPS00089), tr(HKPS00089), tr(HDRN00169)] | 386 | 386 | ft(9,9) | 0 | ft(9,10) | 1500 | **3827** | 8161 | -4334 |
| | | | | | | | | | | | | | | **total** | **-238809** |

# Table E.3. Third part of the gain calculations

| | | Motion | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Table loc (i) | Feeder Loc (i+6) | New table move (i) | | New Turret Movement (i.e. counter clockwise turret movement) | | New Feeder carrier move | | New distance (for step i) max (table, turret, feeder) | Old distance (for step i) max (table, turret, feeder) | Gain (difference between distances) |
| 11 | 34 | mt(13,11) | 8162 | max[tr(HDRN00162), tr(HKPS00089), tr(HDRN00169), tr(HDRN00169), tr(HDRN00180), tr(HDRN00160-449)] | 271 | ft(2,2) | 0 | 8162 | 3997 | 4165 |
| 32 | 15 | mt(34,32) | 8161 | max[tr(HYRL00146), tr(HYRL00146), tr(HKPS00089), tr(HKPS00089), tr(HDRN00169), tr(HDRN00169)] | 386 | ft(9,74) | 97500 | 97500 | 3827 | 93673 |
| 31 | 20 | mt(32,31) | 3827 | max[tr(HYRL00146), tr(HKPS00089), tr(HKPS00089), tr(HDRN00169), tr(HDRN00169), tr(HDRN00180)] | 386 | ft(74,174) | 150000 | 150000 | 4885 | 145115 |
| 4 | 21 | mt(31,4) | 4885 | max[tr(HKPS00089), tr(HKPS00089), tr(HDRN00169), tr(HDRN00169), tr(HDRN00180), tr(HDRN00160-449)] | 271 | ft(174,174) | 0 | 4885 | 3827 | 1058 |
| 6 | 22 | mt(4,6) | 3827 | max[tr(HKPS00089), tr(HDRN00169), tr(HDRN00169), tr(HDRN00180), tr(HDRN00160-449), tr(HDRN00160-449)] | 271 | ft(174,174) | 0 | 3827 | 6000 | -2173 |
| 8 | 26 | mt(6,8) | 3361 | max[tr(HDRN00169), tr(HDRN00169), tr(HDRN00180), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449)] | 271 | ft(174,174) | 0 | 3361 | 3827 | -466 |
| 7 | 25 | mt(8,7) | 3827 | max[tr(HDRN00169), tr(HDRN00180), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449)] | 271 | ft(174,174) | 0 | 3827 | 3997 | -170 |
| 15 | 24 | mt(7,15) | 550 | max[tr(HDRN00180), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449), tr(HDRN00160-449)] | 271 | ft(174,174) | 0 | 550 | 8839 | -8289 |

# APPENDIX F

## PROCEDURE FOR LOWER BOUND 2

$n$ : # of locations

edges: length of the path

$D$: Distance matrix $\ni$ $l \neq k \Rightarrow d_{lk} = d_{kl} = \max\{mt_{lk}, r\}$

$l = k \Rightarrow d_{ll} = M =$ large number

```
PROCEDURE LowerBound ($D$, slength, $n$, edges)
// Initialization //
slength = $M$
// Process //
for $l = 1,...,n$
    getshortest($D, l, n,$ length, edges)
    if length < slength then slength = length
// Output //
Print slength
PROCEDURE Getshortest($D, l, n$, length, edges)
// Initialization //
last = 0
mdistance = $M$
for $k = 1,...,n$
    if mdistance > $D(l,k)$ then    last = $k$
                                    mdistance = $D(l,k)$
    inpath($k$) = False
first = $l$
```

```
length = mdistance
inpath (first) = True
inpath (last) = True
// process //
for $s = 2,..., $ edges
Check ( $D$ , first, $n$ , inpath, fdistance, nextf)
Check ( $D$ , first, $n$ , inpath, ldistance, nextl)
If ldistance < fdistance
        Then  length = length + ldistance
              last = nextl
              Inpath(nextl)= True
        Else  length = length + fdistance
              first = nextf
              inpath (nextf) = True
PROCEDURE Check( $D$ , key, $n$ , inpath, fdistance, next)
// Initialization //
distance = $M$
next = 0
for $k = 1,...,n$
        if (not.inpath( $k$ )) then if $D(k,$ key) < distance then  distance = $D(k,$ key)
                                                                    next = k
```

# APPENDIX G

## Table G.1. PCB DATA FOR CASE A – 1*2.5 – 2 COMPONENTS

| PCB 01 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $g_l$ | $r_g$ | Feeder |
| 1 | 23753 | 6154 | 1 | 100 | 1 |
| 2 | 5778 | 7919 | 1 | 100 | 1 |
| 3 | 15171 | 9218 | 2 | 157 | 2 |
| 4 | 12150 | 7382 | 1 | 100 | 1 |
| 5 | 22282 | 1763 | 1 | 100 | 1 |
| 6 | 19052 | 4057 | 1 | 100 | 1 |
| 7 | 11412 | 9355 | 1 | 100 | 1 |
| 8 | 463 | 9169 | 2 | 157 | 2 |
| 9 | 20535 | 4103 | 1 | 100 | 1 |
| 10 | 11118 | 8936 | 1 | 100 | 1 |

| PCB 02 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $g_l$ | $r_g$ | Feeder |
| 1 | 382 | 8381 | 1 | 100 | 1 |
| 2 | 18670 | 196 | 2 | 157 | 2 |
| 3 | 11127 | 6813 | 1 | 100 | 1 |
| 4 | 23295 | 3795 | 2 | 157 | 2 |
| 5 | 11650 | 8318 | 1 | 100 | 1 |
| 6 | 10466 | 5028 | 2 | 157 | 2 |
| 7 | 21156 | 7095 | 1 | 100 | 1 |
| 8 | 13129 | 4289 | 2 | 157 | 2 |
| 9 | 5066 | 3046 | 2 | 157 | 2 |
| 10 | 16803 | 1897 | 2 | 157 | 2 |

| PCB 03 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $g_l$ | $r_g$ | Feeder |
| 1 | 12414 | 7271 | 2 | 157 | 2 |
| 2 | 22494 | 3093 | 2 | 157 | 2 |
| 3 | 20541 | 8385 | 2 | 157 | 2 |
| 4 | 16123 | 5681 | 2 | 157 | 2 |
| 5 | 20449 | 3704 | 1 | 100 | 1 |
| 6 | 16506 | 7027 | 2 | 157 | 2 |
| 7 | 8549 | 5466 | 1 | 100 | 1 |
| 8 | 7243 | 4449 | 1 | 100 | 1 |
| 9 | 8530 | 6946 | 2 | 157 | 2 |
| 10 | 13352 | 6213 | 2 | 157 | 2 |

| PCB 04 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $g_l$ | $r_g$ | Feeder |
| 1 | 3413 | 5828 | 1 | 100 | 1 |
| 2 | 294 | 4235 | 1 | 100 | 1 |
| 3 | 22347 | 5155 | 2 | 157 | 2 |
| 4 | 4978 | 3340 | 2 | 157 | 2 |
| 5 | 7468 | 4329 | 1 | 100 | 1 |
| 6 | 16536 | 2259 | 2 | 157 | 2 |
| 7 | 7110 | 5798 | 2 | 157 | 2 |
| 8 | 11731 | 7604 | 1 | 100 | 1 |
| 9 | 1620 | 5298 | 2 | 157 | 2 |
| 10 | 24708 | 6405 | 1 | 100 | 1 |

| PCB 05 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $g_l$ | $r_g$ | Feeder |
| 1 | 10384 | 2140 | 2 | 157 | 2 |
| 2 | 7625 | 6435 | 1 | 100 | 1 |
| 3 | 21859 | 3200 | 2 | 157 | 2 |
| 4 | 375 | 9601 | 2 | 157 | 2 |
| 5 | 19199 | 7266 | 1 | 100 | 1 |
| 6 | 24271 | 4120 | 1 | 100 | 1 |
| 7 | 24752 | 7446 | 2 | 157 | 2 |
| 8 | 19722 | 2679 | 2 | 157 | 2 |
| 9 | 10966 | 4399 | 1 | 100 | 1 |
| 10 | 12458 | 9334 | 2 | 157 | 2 |

| PCB 06 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $g_l$ | $r_g$ | Feeder |
| 1 | 11286 | 6085 | 1 | 100 | 1 |
| 2 | 1097 | 158 | 1 | 100 | 1 |
| 3 | 680 | 164 | 1 | 100 | 1 |
| 4 | 7817 | 1901 | 1 | 100 | 1 |
| 5 | 322 | 5869 | 1 | 100 | 1 |
| 6 | 9599 | 576 | 2 | 157 | 2 |
| 7 | 17078 | 3676 | 2 | 157 | 2 |
| 8 | 2321 | 6315 | 2 | 157 | 2 |
| 9 | 883 | 7176 | 1 | 100 | 1 |
| 10 | 15310 | 6927 | 2 | 157 | 2 |

| PCB 07 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $g_l$ | $r_g$ | Feeder |
| 1 | 3026 | 2319 | 1 | 100 | 1 |
| 2 | 11269 | 2393 | 1 | 100 | 1 |
| 3 | 17897 | 498 | 1 | 100 | 1 |
| 4 | 22321 | 784 | 1 | 100 | 1 |
| 5 | 6828 | 6408 | 1 | 100 | 1 |
| 6 | 6369 | 1909 | 2 | 157 | 2 |
| 7 | 21640 | 8439 | 1 | 100 | 1 |
| 8 | 5809 | 1739 | 1 | 100 | 1 |
| 9 | 20122 | 1708 | 1 | 100 | 1 |
| 10 | 22710 | 9943 | 2 | 157 | 2 |

| PCB 08 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $g_l$ | $r_g$ | Feeder |
| 1 | 23356 | 1370 | 1 | 100 | 1 |
| 2 | 6611 | 8188 | 2 | 157 | 2 |
| 3 | 4008 | 4302 | 1 | 100 | 1 |
| 4 | 21821 | 8903 | 2 | 157 | 2 |
| 5 | 5947 | 7349 | 1 | 100 | 1 |
| 6 | 16146 | 6873 | 1 | 100 | 1 |
| 7 | 24172 | 3461 | 1 | 100 | 1 |
| 8 | 16623 | 1660 | 1 | 100 | 1 |
| 9 | 21760 | 1556 | 2 | 157 | 2 |
| 10 | 248 | 1911 | 1 | 100 | 1 |

| PCB 09 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $g_l$ | $r_g$ | Feeder |
| 1 | 7435 | 3759 | 1 | 100 | 1 |
| 2 | 1229 | 99 | 2 | 157 | 2 |
| 3 | 17330 | 4199 | 2 | 157 | 2 |
| 4 | 16253 | 7537 | 2 | 157 | 2 |
| 5 | 24575 | 7939 | 1 | 100 | 1 |
| 6 | 13817 | 9200 | 2 | 157 | 2 |
| 7 | 10002 | 8447 | 2 | 157 | 2 |
| 8 | 4970 | 3678 | 1 | 100 | 1 |
| 9 | 15630 | 6208 | 2 | 157 | 2 |
| 10 | 18334 | 7313 | 1 | 100 | 1 |

| PCB 10 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $g_l$ | $r_g$ | Feeder |
| 1 | 15683 | 7165 | 1 | 100 | 1 |
| 2 | 17477 | 5113 | 2 | 157 | 2 |
| 3 | 9930 | 7764 | 1 | 100 | 1 |
| 4 | 10341 | 4893 | 1 | 100 | 1 |
| 5 | 16380 | 1859 | 2 | 157 | 2 |
| 6 | 20940 | 7006 | 2 | 157 | 2 |
| 7 | 9290 | 9827 | 2 | 157 | 2 |
| 8 | 10631 | 8066 | 2 | 157 | 2 |
| 9 | 14867 | 7036 | 2 | 157 | 2 |
| 10 | 14143 | 4850 | 1 | 100 | 1 |

# Table G.2. PCB DATA FOR CASE A – 2.5*5 – 2 COMPONENTS

| PCB 01 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 47506 | 12308 | 1 | 100 | 1 |
| 2 | 11556 | 15838 | 1 | 100 | 1 |
| 3 | 30342 | 18436 | 2 | 157 | 2 |
| 4 | 24300 | 14764 | 1 | 100 | 1 |
| 5 | 44564 | 3526 | 1 | 100 | 1 |
| 6 | 38104 | 8114 | 1 | 100 | 1 |
| 7 | 22824 | 18710 | 1 | 100 | 1 |
| 8 | 926 | 18338 | 2 | 157 | 2 |
| 9 | 41070 | 8206 | 1 | 100 | 1 |
| 10 | 22236 | 17872 | 1 | 100 | 1 |

| PCB 02 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 764 | 16762 | 1 | 100 | 1 |
| 2 | 37340 | 392 | 2 | 157 | 2 |
| 3 | 22254 | 13626 | 1 | 100 | 1 |
| 4 | 46590 | 7590 | 2 | 157 | 2 |
| 5 | 23300 | 16636 | 1 | 100 | 1 |
| 6 | 20932 | 10056 | 2 | 157 | 2 |
| 7 | 42312 | 14190 | 1 | 100 | 1 |
| 8 | 26258 | 8578 | 2 | 157 | 2 |
| 9 | 10132 | 6092 | 2 | 157 | 2 |
| 10 | 33606 | 3794 | 2 | 157 | 2 |

| PCB 03 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 24828 | 14542 | 2 | 157 | 2 |
| 2 | 44988 | 6186 | 2 | 157 | 2 |
| 3 | 41082 | 16770 | 2 | 157 | 2 |
| 4 | 32246 | 11362 | 2 | 157 | 2 |
| 5 | 40898 | 7408 | 1 | 100 | 1 |
| 6 | 33012 | 14054 | 2 | 157 | 2 |
| 7 | 17098 | 10932 | 1 | 100 | 1 |
| 8 | 14486 | 8898 | 1 | 100 | 1 |
| 9 | 17060 | 13892 | 2 | 157 | 2 |
| 10 | 26704 | 12426 | 2 | 157 | 2 |

| PCB 04 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 6826 | 11656 | 1 | 100 | 1 |
| 2 | 588 | 8470 | 1 | 100 | 1 |
| 3 | 44694 | 10310 | 2 | 157 | 2 |
| 4 | 9956 | 6680 | 2 | 157 | 2 |
| 5 | 14936 | 8658 | 1 | 100 | 1 |
| 6 | 33072 | 4518 | 2 | 157 | 2 |
| 7 | 14220 | 11596 | 2 | 157 | 2 |
| 8 | 23462 | 15208 | 1 | 100 | 1 |
| 9 | 3240 | 10596 | 2 | 157 | 2 |
| 10 | 49416 | 12810 | 1 | 100 | 1 |

| PCB 05 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 20768 | 4280 | 2 | 157 | 2 |
| 2 | 15250 | 12870 | 1 | 100 | 1 |
| 3 | 43718 | 6400 | 2 | 157 | 2 |
| 4 | 750 | 19202 | 2 | 157 | 2 |
| 5 | 38398 | 14532 | 1 | 100 | 1 |
| 6 | 48542 | 8240 | 1 | 100 | 1 |
| 7 | 49504 | 14892 | 2 | 157 | 2 |
| 8 | 39444 | 5358 | 2 | 157 | 2 |
| 9 | 21932 | 8798 | 1 | 100 | 1 |
| 10 | 24916 | 18668 | 2 | 157 | 2 |

| PCB 06 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 22572 | 12170 | 1 | 100 | 1 |
| 2 | 2194 | 316 | 1 | 100 | 1 |
| 3 | 1360 | 328 | 1 | 100 | 1 |
| 4 | 15634 | 3802 | 1 | 100 | 1 |
| 5 | 644 | 11738 | 1 | 100 | 1 |
| 6 | 19198 | 1152 | 2 | 157 | 2 |
| 7 | 34156 | 7352 | 2 | 157 | 2 |
| 8 | 4642 | 12630 | 2 | 157 | 2 |
| 9 | 1766 | 14352 | 1 | 100 | 1 |
| 10 | 30620 | 13854 | 2 | 157 | 2 |

| PCB 07 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 6052 | 4638 | 1 | 100 | 1 |
| 2 | 22538 | 4786 | 1 | 100 | 1 |
| 3 | 35794 | 996 | 1 | 100 | 1 |
| 4 | 44642 | 1568 | 1 | 100 | 1 |
| 5 | 13656 | 12816 | 1 | 100 | 1 |
| 6 | 12738 | 3818 | 2 | 157 | 2 |
| 7 | 43280 | 16878 | 1 | 100 | 1 |
| 8 | 11618 | 3478 | 1 | 100 | 1 |
| 9 | 40244 | 3416 | 1 | 100 | 1 |
| 10 | 45420 | 19886 | 2 | 157 | 2 |

| PCB 08 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 46712 | 2740 | 1 | 100 | 1 |
| 2 | 13222 | 16376 | 2 | 157 | 2 |
| 3 | 8016 | 8604 | 1 | 100 | 1 |
| 4 | 43642 | 17806 | 2 | 157 | 2 |
| 5 | 11894 | 14698 | 1 | 100 | 1 |
| 6 | 32292 | 13746 | 1 | 100 | 1 |
| 7 | 48344 | 6922 | 1 | 100 | 1 |
| 8 | 33246 | 3320 | 1 | 100 | 1 |
| 9 | 43520 | 3112 | 2 | 157 | 2 |
| 10 | 496 | 3822 | 1 | 100 | 1 |

| PCB 09 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 14870 | 7518 | 1 | 100 | 1 |
| 2 | 2458 | 198 | 2 | 157 | 2 |
| 3 | 34660 | 8398 | 2 | 157 | 2 |
| 4 | 32506 | 15074 | 2 | 157 | 2 |
| 5 | 49150 | 15878 | 1 | 100 | 1 |
| 6 | 27634 | 18400 | 2 | 157 | 2 |
| 7 | 20004 | 16894 | 2 | 157 | 2 |
| 8 | 9940 | 7356 | 1 | 100 | 1 |
| 9 | 31260 | 12416 | 2 | 157 | 2 |
| 10 | 36668 | 14626 | 1 | 100 | 1 |

| PCB 10 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 31366 | 14330 | 1 | 100 | 1 |
| 2 | 34954 | 10226 | 2 | 157 | 2 |
| 3 | 19860 | 15528 | 1 | 100 | 1 |
| 4 | 20682 | 9786 | 1 | 100 | 1 |
| 5 | 32760 | 3718 | 2 | 157 | 2 |
| 6 | 41880 | 14012 | 2 | 157 | 2 |
| 7 | 18580 | 19654 | 2 | 157 | 2 |
| 8 | 21262 | 16132 | 2 | 157 | 2 |
| 9 | 29734 | 14072 | 2 | 157 | 2 |
| 10 | 28286 | 9700 | 1 | 100 | 1 |

## Table G.3. PCB DATA FOR CASE A – 10*2.5 – 2 COMPONENTS

| PCB 1 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_{c_l}$ | Feeder |
| 1 | 237530 | 61540 | 1 | 100 | 1 |
| 2 | 57780 | 79190 | 1 | 100 | 1 |
| 3 | 151710 | 92180 | 2 | 157 | 2 |
| 4 | 121500 | 73820 | 1 | 100 | 1 |
| 5 | 222820 | 17630 | 1 | 100 | 1 |
| 6 | 190520 | 40570 | 1 | 100 | 1 |
| 7 | 114120 | 93550 | 1 | 100 | 1 |
| 8 | 4630 | 91690 | 2 | 157 | 2 |
| 9 | 205350 | 41030 | 1 | 100 | 1 |
| 10 | 111180 | 89360 | 1 | 100 | 1 |

| PCB 2 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_{c_l}$ | Feeder |
| 1 | 3820 | 83810 | 1 | 100 | 1 |
| 2 | 186700 | 1960 | 2 | 157 | 2 |
| 3 | 111270 | 68130 | 1 | 100 | 1 |
| 4 | 232950 | 37950 | 2 | 157 | 2 |
| 5 | 116500 | 83180 | 1 | 100 | 1 |
| 6 | 104660 | 50280 | 2 | 157 | 2 |
| 7 | 211560 | 70950 | 1 | 100 | 1 |
| 8 | 131290 | 42890 | 2 | 157 | 2 |
| 9 | 50660 | 30460 | 2 | 157 | 2 |
| 10 | 168030 | 18970 | 2 | 157 | 2 |

| PCB 3 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_{c_l}$ | Feeder |
| 1 | 124140 | 72710 | 2 | 157 | 2 |
| 2 | 224940 | 30930 | 2 | 157 | 2 |
| 3 | 205410 | 83850 | 2 | 157 | 2 |
| 4 | 161230 | 56810 | 2 | 157 | 2 |
| 5 | 204490 | 37040 | 1 | 100 | 1 |
| 6 | 165060 | 70270 | 2 | 157 | 2 |
| 7 | 85490 | 54660 | 1 | 100 | 1 |
| 8 | 72430 | 44490 | 1 | 100 | 1 |
| 9 | 85300 | 69460 | 2 | 157 | 2 |
| 10 | 133520 | 62130 | 2 | 157 | 2 |

| PCB 4 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_{c_l}$ | Feeder |
| 1 | 34130 | 58280 | 1 | 100 | 1 |
| 2 | 2940 | 42350 | 1 | 100 | 1 |
| 3 | 223470 | 51550 | 2 | 157 | 2 |
| 4 | 49780 | 33400 | 2 | 157 | 2 |
| 5 | 74680 | 43290 | 1 | 100 | 1 |
| 6 | 165360 | 22590 | 2 | 157 | 2 |
| 7 | 71100 | 57980 | 2 | 157 | 2 |
| 8 | 117310 | 76040 | 1 | 100 | 1 |
| 9 | 16200 | 52980 | 2 | 157 | 2 |
| 10 | 247080 | 64050 | 1 | 100 | 1 |

| PCB 5 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_{c_l}$ | Feeder |
| 1 | 103840 | 21400 | 2 | 157 | 2 |
| 2 | 76250 | 64350 | 1 | 100 | 1 |
| 3 | 218590 | 32000 | 2 | 157 | 2 |
| 4 | 3750 | 96010 | 2 | 157 | 2 |
| 5 | 191990 | 72660 | 1 | 100 | 1 |
| 6 | 242710 | 41200 | 1 | 100 | 1 |
| 7 | 247520 | 74460 | 2 | 157 | 2 |
| 8 | 197220 | 26790 | 2 | 157 | 2 |
| 9 | 109660 | 43990 | 1 | 100 | 1 |
| 10 | 124580 | 93340 | 2 | 157 | 2 |

| PCB 6 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_{c_l}$ | Feeder |
| 1 | 112860 | 60850 | 1 | 100 | 1 |
| 2 | 10970 | 1580 | 1 | 100 | 1 |
| 3 | 6800 | 1640 | 1 | 100 | 1 |
| 4 | 78170 | 19010 | 1 | 100 | 1 |
| 5 | 3220 | 58690 | 1 | 100 | 1 |
| 6 | 95990 | 5760 | 2 | 157 | 2 |
| 7 | 170780 | 36760 | 2 | 157 | 2 |
| 8 | 23210 | 63150 | 2 | 157 | 2 |
| 9 | 8830 | 71760 | 1 | 100 | 1 |
| 10 | 153100 | 69270 | 2 | 157 | 2 |

| PCB 7 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_{c_l}$ | Feeder |
| 1 | 30260 | 23190 | 1 | 100 | 1 |
| 2 | 112690 | 23930 | 1 | 100 | 1 |
| 3 | 178970 | 4980 | 1 | 100 | 1 |
| 4 | 223210 | 7840 | 1 | 100 | 1 |
| 5 | 68280 | 64080 | 1 | 100 | 1 |
| 6 | 63690 | 19090 | 2 | 157 | 2 |
| 7 | 216400 | 84390 | 1 | 100 | 1 |
| 8 | 58090 | 17390 | 1 | 100 | 1 |
| 9 | 201220 | 17080 | 1 | 100 | 1 |
| 10 | 227100 | 99430 | 2 | 157 | 2 |

| PCB 8 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_{c_l}$ | Feeder |
| 1 | 233560 | 13700 | 1 | 100 | 1 |
| 2 | 66110 | 81880 | 2 | 157 | 2 |
| 3 | 40080 | 43020 | 1 | 100 | 1 |
| 4 | 218210 | 89030 | 2 | 157 | 2 |
| 5 | 59470 | 73490 | 1 | 100 | 1 |
| 6 | 161460 | 68730 | 1 | 100 | 1 |
| 7 | 241720 | 34610 | 1 | 100 | 1 |
| 8 | 166230 | 16600 | 1 | 100 | 1 |
| 9 | 217600 | 15560 | 2 | 157 | 2 |
| 10 | 2480 | 19110 | 1 | 100 | 1 |

| PCB 9 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_{c_l}$ | Feeder |
| 1 | 74350 | 37590 | 1 | 100 | 1 |
| 2 | 12290 | 990 | 2 | 157 | 2 |
| 3 | 173300 | 41990 | 2 | 157 | 2 |
| 4 | 162530 | 75370 | 2 | 157 | 2 |
| 5 | 245750 | 79390 | 1 | 100 | 1 |
| 6 | 138170 | 92000 | 2 | 157 | 2 |
| 7 | 100020 | 84470 | 2 | 157 | 2 |
| 8 | 49700 | 36780 | 1 | 100 | 1 |
| 9 | 156300 | 62080 | 2 | 157 | 2 |
| 10 | 183340 | 73130 | 1 | 100 | 1 |

| PCB 10 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_{c_l}$ | Feeder |
| 1 | 156830 | 71650 | 1 | 100 | 1 |
| 2 | 174770 | 51130 | 2 | 157 | 2 |
| 3 | 99300 | 77640 | 1 | 100 | 1 |
| 4 | 103410 | 48930 | 1 | 100 | 1 |
| 5 | 163800 | 18590 | 2 | 157 | 2 |
| 6 | 209400 | 70060 | 2 | 157 | 2 |
| 7 | 92900 | 98270 | 2 | 157 | 2 |
| 8 | 106310 | 80660 | 2 | 157 | 2 |
| 9 | 148670 | 70360 | 2 | 157 | 2 |
| 10 | 141430 | 48500 | 1 | 100 | 1 |

# Table G.4. PCB DATA FOR CASE A – 1*2.5 – 4 COMPONENTS

| PCB 01 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_q$ | $r_q$ | Feeder |
| 1 | 23753 | 6154 | 1 | 100 | 1 |
| 2 | 5778 | 7919 | 2 | 157 | 2 |
| 3 | 15171 | 9218 | 4 | 271 | 4 |
| 4 | 12150 | 7382 | 1 | 100 | 1 |
| 5 | 22282 | 1763 | 1 | 100 | 1 |
| 6 | 19052 | 4057 | 1 | 100 | 1 |
| 7 | 11412 | 9355 | 1 | 100 | 1 |
| 8 | 463 | 9169 | 3 | 214 | 3 |
| 9 | 20535 | 4103 | 2 | 157 | 2 |
| 10 | 11118 | 8936 | 1 | 100 | 1 |

| PCB 02 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_q$ | $r_q$ | Feeder |
| 1 | 382 | 8381 | 1 | 100 | 1 |
| 2 | 18670 | 196 | 1 | 100 | 1 |
| 3 | 11127 | 6813 | 3 | 214 | 3 |
| 4 | 23295 | 3795 | 3 | 214 | 3 |
| 5 | 11650 | 8318 | 2 | 157 | 2 |
| 6 | 10466 | 5028 | 2 | 157 | 2 |
| 7 | 21156 | 7095 | 3 | 214 | 3 |
| 8 | 13129 | 4289 | 4 | 271 | 4 |
| 9 | 5066 | 3046 | 4 | 271 | 4 |
| 10 | 16803 | 1897 | 3 | 214 | 3 |

| PCB 03 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_q$ | $r_q$ | Feeder |
| 1 | 12414 | 7271 | 4 | 271 | 4 |
| 2 | 22494 | 3093 | 4 | 271 | 4 |
| 3 | 20541 | 8385 | 3 | 214 | 3 |
| 4 | 16123 | 5681 | 4 | 271 | 4 |
| 5 | 20449 | 3704 | 1 | 100 | 1 |
| 6 | 16506 | 7027 | 4 | 271 | 4 |
| 7 | 8549 | 5466 | 2 | 157 | 2 |
| 8 | 7243 | 4449 | 2 | 157 | 2 |
| 9 | 8530 | 6946 | 4 | 271 | 4 |
| 10 | 13352 | 6213 | 3 | 214 | 3 |

| PCB 04 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_q$ | $r_q$ | Feeder |
| 1 | 3413 | 5828 | 1 | 100 | 1 |
| 2 | 294 | 4235 | 2 | 157 | 2 |
| 3 | 22347 | 5155 | 4 | 271 | 4 |
| 4 | 4978 | 3340 | 3 | 214 | 3 |
| 5 | 7468 | 4329 | 2 | 157 | 2 |
| 6 | 16536 | 2259 | 3 | 214 | 3 |
| 7 | 7110 | 5798 | 4 | 271 | 4 |
| 8 | 11731 | 7604 | 1 | 100 | 1 |
| 9 | 1620 | 5298 | 3 | 214 | 3 |
| 10 | 24708 | 6405 | 1 | 100 | 1 |

| PCB 05 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_q$ | $r_q$ | Feeder |
| 1 | 10384 | 2140 | 3 | 214 | 3 |
| 2 | 7625 | 6435 | 1 | 100 | 1 |
| 3 | 21859 | 3200 | 4 | 271 | 4 |
| 4 | 375 | 9601 | 3 | 214 | 3 |
| 5 | 19199 | 7266 | 1 | 100 | 1 |
| 6 | 24271 | 4120 | 1 | 100 | 1 |
| 7 | 24752 | 7446 | 3 | 214 | 3 |
| 8 | 19722 | 2679 | 3 | 214 | 3 |
| 9 | 10966 | 4399 | 2 | 157 | 2 |
| 10 | 12458 | 9334 | 3 | 214 | 3 |

| PCB 06 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_q$ | $r_q$ | Feeder |
| 1 | 11286 | 6085 | 1 | 100 | 1 |
| 2 | 1097 | 158 | 2 | 157 | 2 |
| 3 | 680 | 164 | 2 | 157 | 2 |
| 4 | 7817 | 1901 | 2 | 157 | 2 |
| 5 | 322 | 5869 | 1 | 100 | 1 |
| 6 | 9599 | 576 | 3 | 214 | 3 |
| 7 | 17078 | 3676 | 3 | 214 | 3 |
| 8 | 2321 | 6315 | 3 | 214 | 3 |
| 9 | 883 | 7176 | 2 | 157 | 2 |
| 10 | 15310 | 6927 | 4 | 271 | 4 |

| PCB 07 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_q$ | $r_q$ | Feeder |
| 1 | 3026 | 2319 | 2 | 157 | 2 |
| 2 | 11269 | 2393 | 2 | 157 | 2 |
| 3 | 17897 | 498 | 2 | 157 | 2 |
| 4 | 22321 | 784 | 2 | 157 | 2 |
| 5 | 6828 | 6408 | 2 | 157 | 2 |
| 6 | 6369 | 1909 | 3 | 214 | 3 |
| 7 | 21640 | 8439 | 1 | 100 | 1 |
| 8 | 5809 | 1739 | 1 | 100 | 1 |
| 9 | 20122 | 1708 | 2 | 157 | 2 |
| 10 | 22710 | 9943 | 4 | 271 | 4 |

| PCB 08 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_q$ | $r_q$ | Feeder |
| 1 | 23356 | 1370 | 3 | 214 | 3 |
| 2 | 6611 | 8188 | 4 | 271 | 4 |
| 3 | 4008 | 4302 | 2 | 157 | 2 |
| 4 | 21821 | 8903 | 4 | 271 | 4 |
| 5 | 5947 | 7349 | 2 | 157 | 2 |
| 6 | 16146 | 6873 | 2 | 157 | 2 |
| 7 | 24172 | 3461 | 2 | 157 | 2 |
| 8 | 16623 | 1660 | 2 | 157 | 2 |
| 9 | 21760 | 1556 | 4 | 271 | 4 |
| 10 | 248 | 1911 | 1 | 100 | 1 |

| PCB 09 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_q$ | $r_q$ | Feeder |
| 1 | 7435 | 3759 | 1 | 100 | 1 |
| 2 | 1229 | 99 | 4 | 271 | 4 |
| 3 | 17330 | 4199 | 3 | 214 | 3 |
| 4 | 16253 | 7537 | 3 | 214 | 3 |
| 5 | 24575 | 7939 | 1 | 100 | 1 |
| 6 | 13817 | 9200 | 3 | 214 | 3 |
| 7 | 10002 | 8447 | 4 | 271 | 4 |
| 8 | 4970 | 3678 | 2 | 157 | 2 |
| 9 | 15630 | 6208 | 3 | 214 | 3 |
| 10 | 18334 | 7313 | 2 | 157 | 2 |

| PCB 10 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_q$ | $r_q$ | Feeder |
| 1 | 15683 | 7165 | 1 | 100 | 1 |
| 2 | 17477 | 5113 | 3 | 214 | 3 |
| 3 | 9930 | 7764 | 2 | 157 | 2 |
| 4 | 10341 | 4893 | 1 | 100 | 1 |
| 5 | 16380 | 1859 | 3 | 214 | 3 |
| 6 | 20940 | 7006 | 4 | 271 | 4 |
| 7 | 9290 | 9827 | 3 | 214 | 3 |
| 8 | 10631 | 8066 | 4 | 271 | 4 |
| 9 | 14867 | 7036 | 4 | 271 | 4 |
| 10 | 14143 | 4850 | 1 | 100 | 1 |

# Table G.5. PCB DATA FOR CASE A –2*5 – 4 COMPONENTS

| PCB 01 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_q$ | Feeder |
| 1 | 47506 | 12308 | 1 | 100 | 1 |
| 2 | 11556 | 15838 | 2 | 157 | 2 |
| 3 | 30342 | 18436 | 4 | 271 | 4 |
| 4 | 24300 | 14764 | 1 | 100 | 1 |
| 5 | 44564 | 3526 | 1 | 100 | 1 |
| 6 | 38104 | 8114 | 1 | 100 | 1 |
| 7 | 22824 | 18710 | 1 | 100 | 1 |
| 8 | 926 | 18338 | 3 | 214 | 3 |
| 9 | 41070 | 8206 | 2 | 157 | 2 |
| 10 | 22236 | 17872 | 1 | 100 | 1 |

| PCB 02 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_q$ | Feeder |
| 1 | 764 | 16762 | 1 | 100 | 1 |
| 2 | 37340 | 392 | 1 | 100 | 1 |
| 3 | 22254 | 13626 | 3 | 214 | 3 |
| 4 | 46590 | 7590 | 3 | 214 | 3 |
| 5 | 23300 | 16636 | 2 | 157 | 2 |
| 6 | 20932 | 10056 | 2 | 157 | 2 |
| 7 | 42312 | 14190 | 3 | 214 | 3 |
| 8 | 26258 | 8578 | 4 | 271 | 4 |
| 9 | 10132 | 6092 | 4 | 271 | 4 |
| 10 | 33606 | 3794 | 3 | 214 | 3 |

| PCB 03 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_q$ | Feeder |
| 1 | 24828 | 14542 | 4 | 271 | 4 |
| 2 | 44988 | 6186 | 4 | 271 | 4 |
| 3 | 41082 | 16770 | 3 | 214 | 3 |
| 4 | 32246 | 11362 | 4 | 271 | 4 |
| 5 | 40898 | 7408 | 1 | 100 | 1 |
| 6 | 33012 | 14054 | 4 | 271 | 4 |
| 7 | 17098 | 10932 | 2 | 157 | 2 |
| 8 | 14486 | 8898 | 2 | 157 | 2 |
| 9 | 17060 | 13892 | 4 | 271 | 4 |
| 10 | 26704 | 12426 | 3 | 214 | 3 |

| PCB 04 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_q$ | Feeder |
| 1 | 6826 | 11656 | 1 | 100 | 1 |
| 2 | 588 | 8470 | 2 | 157 | 2 |
| 3 | 44694 | 10310 | 4 | 271 | 4 |
| 4 | 9956 | 6680 | 3 | 214 | 3 |
| 5 | 14936 | 8658 | 2 | 157 | 2 |
| 6 | 33072 | 4518 | 3 | 214 | 3 |
| 7 | 14220 | 11596 | 4 | 271 | 4 |
| 8 | 23462 | 15208 | 1 | 100 | 1 |
| 9 | 3240 | 10596 | 3 | 214 | 3 |
| 10 | 49416 | 12810 | 1 | 100 | 1 |

| PCB 05 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_q$ | Feeder |
| 1 | 20768 | 4280 | 3 | 214 | 3 |
| 2 | 15250 | 12870 | 1 | 100 | 1 |
| 3 | 43718 | 6400 | 4 | 271 | 4 |
| 4 | 750 | 19202 | 3 | 214 | 3 |
| 5 | 38398 | 14532 | 1 | 100 | 1 |
| 6 | 48542 | 8240 | 1 | 100 | 1 |
| 7 | 49504 | 14892 | 3 | 214 | 3 |
| 8 | 39444 | 5358 | 3 | 214 | 3 |
| 9 | 21932 | 8798 | 2 | 157 | 2 |
| 10 | 24916 | 18668 | 3 | 214 | 3 |

| PCB 06 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_q$ | Feeder |
| 1 | 22572 | 12170 | 1 | 100 | 1 |
| 2 | 2194 | 316 | 2 | 157 | 2 |
| 3 | 1360 | 328 | 2 | 157 | 2 |
| 4 | 15634 | 3802 | 2 | 157 | 2 |
| 5 | 644 | 11738 | 1 | 100 | 1 |
| 6 | 19198 | 1152 | 3 | 214 | 3 |
| 7 | 34156 | 7352 | 3 | 214 | 3 |
| 8 | 4642 | 12630 | 3 | 214 | 3 |
| 9 | 1766 | 14352 | 2 | 157 | 2 |
| 10 | 30620 | 13854 | 4 | 271 | 4 |

| PCB 07 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_q$ | Feeder |
| 1 | 6052 | 4638 | 2 | 157 | 2 |
| 2 | 22538 | 4786 | 2 | 157 | 2 |
| 3 | 35794 | 996 | 2 | 157 | 2 |
| 4 | 44642 | 1568 | 2 | 157 | 2 |
| 5 | 13656 | 12816 | 2 | 157 | 2 |
| 6 | 12738 | 3818 | 3 | 214 | 3 |
| 7 | 43280 | 16878 | 1 | 100 | 1 |
| 8 | 11618 | 3478 | 1 | 100 | 1 |
| 9 | 40244 | 3416 | 2 | 157 | 2 |
| 10 | 45420 | 19886 | 4 | 271 | 4 |

| PCB 08 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_q$ | Feeder |
| 1 | 46712 | 2740 | 3 | 214 | 3 |
| 2 | 13222 | 16376 | 4 | 271 | 4 |
| 3 | 8016 | 8604 | 2 | 157 | 2 |
| 4 | 43642 | 17806 | 4 | 271 | 4 |
| 5 | 11894 | 14698 | 2 | 157 | 2 |
| 6 | 32292 | 13746 | 2 | 157 | 2 |
| 7 | 48344 | 6922 | 2 | 157 | 2 |
| 8 | 33246 | 3320 | 2 | 157 | 2 |
| 9 | 43520 | 3112 | 4 | 271 | 4 |
| 10 | 496 | 3822 | 1 | 100 | 1 |

| PCB 09 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_q$ | Feeder |
| 1 | 14870 | 7518 | 1 | 100 | 1 |
| 2 | 2458 | 198 | 4 | 271 | 4 |
| 3 | 34660 | 8398 | 3 | 214 | 3 |
| 4 | 32506 | 15074 | 3 | 214 | 3 |
| 5 | 49150 | 15878 | 1 | 100 | 1 |
| 6 | 27634 | 18400 | 3 | 214 | 3 |
| 7 | 20004 | 16894 | 4 | 271 | 4 |
| 8 | 9940 | 7356 | 2 | 157 | 2 |
| 9 | 31260 | 12416 | 3 | 214 | 3 |
| 10 | 36668 | 14626 | 2 | 157 | 2 |

| PCB 10 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_q$ | Feeder |
| 1 | 31366 | 14330 | 1 | 100 | 1 |
| 2 | 34954 | 10226 | 3 | 214 | 3 |
| 3 | 19860 | 15528 | 2 | 157 | 2 |
| 4 | 20682 | 9786 | 1 | 100 | 1 |
| 5 | 32760 | 3718 | 3 | 214 | 3 |
| 6 | 41880 | 14012 | 4 | 271 | 4 |
| 7 | 18580 | 19654 | 3 | 214 | 3 |
| 8 | 21262 | 16132 | 4 | 271 | 4 |
| 9 | 29734 | 14072 | 4 | 271 | 4 |
| 10 | 28286 | 9700 | 1 | 100 | 1 |

# Table G.6. PCB DATA FOR CASE A – 10*25 – 4 COMPONENTS

PCB 01

| $l$ | X | Y | $c_l$ | $r_c$ | Feeder |
|---|---|---|---|---|---|
| 1 | 237530 | 61540 | 1 | 100 | 1 |
| 2 | 57780 | 79190 | 2 | 157 | 2 |
| 3 | 151710 | 92180 | 4 | 271 | 4 |
| 4 | 121500 | 73820 | 1 | 100 | 1 |
| 5 | 222820 | 17630 | 1 | 100 | 1 |
| 6 | 190520 | 40570 | 1 | 100 | 1 |
| 7 | 114120 | 93550 | 1 | 100 | 1 |
| 8 | 4630 | 91690 | 3 | 214 | 3 |
| 9 | 205350 | 41030 | 2 | 157 | 2 |
| 10 | 111180 | 89360 | 1 | 100 | 1 |

PCB 02

| $l$ | X | Y | $c_l$ | $r_c$ | Feeder |
|---|---|---|---|---|---|
| 1 | 3820 | 83810 | 1 | 100 | 1 |
| 2 | 186700 | 1960 | 1 | 100 | 1 |
| 3 | 111270 | 68130 | 3 | 214 | 3 |
| 4 | 232950 | 37950 | 3 | 214 | 3 |
| 5 | 116500 | 83180 | 2 | 157 | 2 |
| 6 | 104660 | 50280 | 2 | 157 | 2 |
| 7 | 211560 | 70950 | 3 | 214 | 3 |
| 8 | 131290 | 42890 | 4 | 271 | 4 |
| 9 | 50660 | 30460 | 4 | 271 | 4 |
| 10 | 168030 | 18970 | 3 | 214 | 3 |

PCB 03

| $l$ | X | Y | $c_l$ | $r_c$ | Feeder |
|---|---|---|---|---|---|
| 1 | 124140 | 72710 | 4 | 271 | 4 |
| 2 | 224940 | 30930 | 4 | 271 | 4 |
| 3 | 205410 | 83850 | 3 | 214 | 3 |
| 4 | 161230 | 56810 | 4 | 271 | 4 |
| 5 | 204490 | 37040 | 1 | 100 | 1 |
| 6 | 165060 | 70270 | 4 | 271 | 4 |
| 7 | 85490 | 54660 | 2 | 157 | 2 |
| 8 | 72430 | 44490 | 2 | 157 | 2 |
| 9 | 85300 | 69460 | 4 | 271 | 4 |
| 10 | 133520 | 62130 | 3 | 214 | 3 |

PCB 04

| $l$ | X | Y | $c_l$ | $r_c$ | Feeder |
|---|---|---|---|---|---|
| 1 | 34130 | 58280 | 1 | 100 | 1 |
| 2 | 2940 | 42350 | 2 | 157 | 2 |
| 3 | 223470 | 51550 | 4 | 271 | 4 |
| 4 | 49780 | 33400 | 3 | 214 | 3 |
| 5 | 74680 | 43290 | 2 | 157 | 2 |
| 6 | 165360 | 22590 | 3 | 214 | 3 |
| 7 | 71100 | 57980 | 4 | 271 | 4 |
| 8 | 117310 | 76040 | 1 | 100 | 1 |
| 9 | 16200 | 52980 | 3 | 214 | 3 |
| 10 | 247080 | 64050 | 1 | 100 | 1 |

PCB 05

| $l$ | X | Y | $c_l$ | $r_c$ | Feeder |
|---|---|---|---|---|---|
| 1 | 103840 | 21400 | 3 | 214 | 3 |
| 2 | 76250 | 64350 | 1 | 100 | 1 |
| 3 | 218590 | 32000 | 4 | 271 | 4 |
| 4 | 3750 | 96010 | 3 | 214 | 3 |
| 5 | 191990 | 72660 | 1 | 100 | 1 |
| 6 | 242710 | 41200 | 1 | 100 | 1 |
| 7 | 247520 | 74460 | 3 | 214 | 3 |
| 8 | 197220 | 26790 | 3 | 214 | 3 |
| 9 | 109660 | 43990 | 2 | 157 | 2 |
| 10 | 124580 | 93340 | 3 | 214 | 3 |

PCB 06

| $l$ | X | Y | $c_l$ | $r_c$ | Feeder |
|---|---|---|---|---|---|
| 1 | 112860 | 60850 | 1 | 100 | 1 |
| 2 | 10970 | 1580 | 2 | 157 | 2 |
| 3 | 6800 | 1640 | 2 | 157 | 2 |
| 4 | 78170 | 19010 | 2 | 157 | 2 |
| 5 | 3220 | 58690 | 1 | 100 | 1 |
| 6 | 95990 | 5760 | 3 | 214 | 3 |
| 7 | 170780 | 36760 | 3 | 214 | 3 |
| 8 | 23210 | 63150 | 3 | 214 | 3 |
| 9 | 8830 | 71760 | 2 | 157 | 2 |
| 10 | 153100 | 69270 | 4 | 271 | 4 |

PCB 07

| $l$ | X | Y | $c_l$ | $r_c$ | Feeder |
|---|---|---|---|---|---|
| 1 | 30260 | 23190 | 2 | 157 | 2 |
| 2 | 112690 | 23930 | 2 | 157 | 2 |
| 3 | 178970 | 4980 | 2 | 157 | 2 |
| 4 | 223210 | 7840 | 2 | 157 | 2 |
| 5 | 68280 | 64080 | 2 | 157 | 2 |
| 6 | 63690 | 19090 | 3 | 214 | 3 |
| 7 | 216400 | 84390 | 1 | 100 | 1 |
| 8 | 58090 | 17390 | 1 | 100 | 1 |
| 9 | 201220 | 17080 | 2 | 157 | 2 |
| 10 | 227100 | 99430 | 4 | 271 | 4 |

PCB 08

| $l$ | X | Y | $c_l$ | $r_c$ | Feeder |
|---|---|---|---|---|---|
| 1 | 233560 | 13700 | 3 | 214 | 3 |
| 2 | 66110 | 81880 | 4 | 271 | 4 |
| 3 | 40080 | 43020 | 2 | 157 | 2 |
| 4 | 218210 | 89030 | 4 | 271 | 4 |
| 5 | 59470 | 73490 | 2 | 157 | 2 |
| 6 | 161460 | 68730 | 2 | 157 | 2 |
| 7 | 241720 | 34610 | 2 | 157 | 2 |
| 8 | 166230 | 16600 | 2 | 157 | 2 |
| 9 | 217600 | 15560 | 4 | 271 | 4 |
| 10 | 2480 | 19110 | 1 | 100 | 1 |

PCB 09

| $l$ | X | Y | $c_l$ | $r_c$ | Feeder |
|---|---|---|---|---|---|
| 1 | 74350 | 37590 | 1 | 100 | 1 |
| 2 | 12290 | 990 | 4 | 271 | 4 |
| 3 | 173300 | 41990 | 3 | 214 | 3 |
| 4 | 162530 | 75370 | 3 | 214 | 3 |
| 5 | 245750 | 79390 | 1 | 100 | 1 |
| 6 | 138170 | 92000 | 3 | 214 | 3 |
| 7 | 100020 | 84470 | 4 | 271 | 4 |
| 8 | 49700 | 36780 | 2 | 157 | 2 |
| 9 | 156300 | 62080 | 3 | 214 | 3 |
| 10 | 183340 | 73130 | 2 | 157 | 2 |

PCB 10

| $l$ | X | Y | $c_l$ | $r_c$ | Feeder |
|---|---|---|---|---|---|
| 1 | 156830 | 71650 | 1 | 100 | 1 |
| 2 | 174770 | 51130 | 3 | 214 | 3 |
| 3 | 99300 | 77640 | 2 | 157 | 2 |
| 4 | 103410 | 48930 | 1 | 100 | 1 |
| 5 | 163800 | 18590 | 3 | 214 | 3 |
| 6 | 209400 | 70060 | 4 | 271 | 4 |
| 7 | 92900 | 98270 | 3 | 214 | 3 |
| 8 | 106310 | 80660 | 4 | 271 | 4 |
| 9 | 148670 | 70360 | 4 | 271 | 4 |
| 10 | 141430 | 48500 | 1 | 100 | 1 |

# Table G.7. PCB DATA FOR CASE B – 1*2.5 – 2 COMPONENTS

| PCB 01 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_q$ | Feeder |
| 1 | 23753 | 6154 | 1 | 100 | 4 |
| 2 | 5778 | 7919 | 1 | 100 | 4 |
| 3 | 15171 | 9218 | 2 | 157 | 8 |
| 4 | 12150 | 7382 | 1 | 100 | 4 |
| 5 | 22282 | 1763 | 1 | 100 | 4 |
| 6 | 19052 | 4057 | 1 | 100 | 4 |
| 7 | 11412 | 9355 | 1 | 100 | 4 |
| 8 | 463 | 9169 | 2 | 157 | 8 |
| 9 | 20535 | 4103 | 1 | 100 | 4 |
| 10 | 11118 | 8936 | 1 | 100 | 4 |

| PCB 02 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_q$ | Feeder |
| 1 | 382 | 8381 | 1 | 100 | 4 |
| 2 | 18670 | 196 | 2 | 157 | 8 |
| 3 | 11127 | 6813 | 1 | 100 | 4 |
| 4 | 23295 | 3795 | 2 | 157 | 8 |
| 5 | 11650 | 8318 | 1 | 100 | 4 |
| 6 | 10466 | 5028 | 2 | 157 | 8 |
| 7 | 21156 | 7095 | 1 | 100 | 4 |
| 8 | 13129 | 4289 | 2 | 157 | 8 |
| 9 | 5066 | 3046 | 2 | 157 | 8 |
| 10 | 16803 | 1897 | 2 | 157 | 8 |

| PCB 03 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_q$ | Feeder |
| 1 | 12414 | 7271 | 2 | 157 | 8 |
| 2 | 22494 | 3093 | 2 | 157 | 8 |
| 3 | 20541 | 8385 | 2 | 157 | 8 |
| 4 | 16123 | 5681 | 2 | 157 | 8 |
| 5 | 20449 | 3704 | 1 | 100 | 4 |
| 6 | 16506 | 7027 | 2 | 157 | 8 |
| 7 | 8549 | 5466 | 1 | 100 | 4 |
| 8 | 7243 | 4449 | 1 | 100 | 4 |
| 9 | 8530 | 6946 | 2 | 157 | 8 |
| 10 | 13352 | 6213 | 2 | 157 | 8 |

| PCB 04 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_q$ | Feeder |
| 1 | 3413 | 5828 | 1 | 100 | 4 |
| 2 | 294 | 4235 | 1 | 100 | 4 |
| 3 | 22347 | 5155 | 2 | 157 | 8 |
| 4 | 4978 | 3340 | 2 | 157 | 8 |
| 5 | 7468 | 4329 | 1 | 100 | 4 |
| 6 | 16536 | 2259 | 2 | 157 | 8 |
| 7 | 7110 | 5798 | 2 | 157 | 8 |
| 8 | 11731 | 7604 | 1 | 100 | 4 |
| 9 | 1620 | 5298 | 2 | 157 | 8 |
| 10 | 24708 | 6405 | 1 | 100 | 4 |

| PCB 05 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_q$ | Feeder |
| 1 | 10384 | 2140 | 2 | 157 | 8 |
| 2 | 7625 | 6435 | 1 | 100 | 4 |
| 3 | 21859 | 3200 | 2 | 157 | 8 |
| 4 | 375 | 9601 | 2 | 157 | 8 |
| 5 | 19199 | 7266 | 1 | 100 | 4 |
| 6 | 24271 | 4120 | 1 | 100 | 4 |
| 7 | 24752 | 7446 | 2 | 157 | 8 |
| 8 | 19722 | 2679 | 2 | 157 | 8 |
| 9 | 10966 | 4399 | 1 | 100 | 4 |
| 10 | 12458 | 9334 | 2 | 157 | 8 |

| PCB 06 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_q$ | Feeder |
| 1 | 11286 | 6085 | 1 | 100 | 4 |
| 2 | 1097 | 158 | 1 | 100 | 4 |
| 3 | 680 | 164 | 1 | 100 | 4 |
| 4 | 7817 | 1901 | 1 | 100 | 4 |
| 5 | 322 | 5869 | 1 | 100 | 4 |
| 6 | 9599 | 576 | 2 | 157 | 8 |
| 7 | 17078 | 3676 | 2 | 157 | 8 |
| 8 | 2321 | 6315 | 2 | 157 | 8 |
| 9 | 883 | 7176 | 1 | 100 | 4 |
| 10 | 15310 | 6927 | 2 | 157 | 8 |

| PCB 07 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_q$ | Feeder |
| 1 | 3026 | 2319 | 1 | 100 | 4 |
| 2 | 11269 | 2393 | 1 | 100 | 4 |
| 3 | 17897 | 498 | 1 | 100 | 4 |
| 4 | 22321 | 784 | 1 | 100 | 4 |
| 5 | 6828 | 6408 | 1 | 100 | 4 |
| 6 | 6369 | 1909 | 2 | 157 | 8 |
| 7 | 21640 | 8439 | 1 | 100 | 4 |
| 8 | 5809 | 1739 | 1 | 100 | 4 |
| 9 | 20122 | 1708 | 1 | 100 | 4 |
| 10 | 22710 | 9943 | 2 | 157 | 8 |

| PCB 08 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_q$ | Feeder |
| 1 | 23356 | 1370 | 1 | 100 | 4 |
| 2 | 6611 | 8188 | 2 | 157 | 8 |
| 3 | 4008 | 4302 | 1 | 100 | 4 |
| 4 | 21821 | 8903 | 2 | 157 | 8 |
| 5 | 5947 | 7349 | 1 | 100 | 4 |
| 6 | 16146 | 6873 | 1 | 100 | 4 |
| 7 | 24172 | 3461 | 1 | 100 | 4 |
| 8 | 16623 | 1660 | 1 | 100 | 4 |
| 9 | 21760 | 1556 | 2 | 157 | 8 |
| 10 | 248 | 1911 | 1 | 100 | 4 |

| PCB 09 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_q$ | Feeder |
| 1 | 7435 | 3759 | 1 | 100 | 4 |
| 2 | 1229 | 99 | 2 | 157 | 8 |
| 3 | 17330 | 4199 | 2 | 157 | 8 |
| 4 | 16253 | 7537 | 2 | 157 | 8 |
| 5 | 24575 | 7939 | 1 | 100 | 4 |
| 6 | 13817 | 9200 | 2 | 157 | 8 |
| 7 | 10002 | 8447 | 2 | 157 | 8 |
| 8 | 4970 | 3678 | 1 | 100 | 4 |
| 9 | 15630 | 6208 | 2 | 157 | 8 |
| 10 | 18334 | 7313 | 1 | 100 | 4 |

| PCB 10 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $c_l$ | $r_q$ | Feeder |
| 1 | 15683 | 7165 | 1 | 100 | 4 |
| 2 | 17477 | 5113 | 2 | 157 | 8 |
| 3 | 9930 | 7764 | 1 | 100 | 4 |
| 4 | 10341 | 4893 | 1 | 100 | 4 |
| 5 | 16380 | 1859 | 2 | 157 | 8 |
| 6 | 20940 | 7006 | 2 | 157 | 8 |
| 7 | 9290 | 9827 | 2 | 157 | 8 |
| 8 | 10631 | 8066 | 2 | 157 | 8 |
| 9 | 14867 | 7036 | 2 | 157 | 8 |
| 10 | 14143 | 4850 | 1 | 100 | 4 |

# Table G.8. PCB DATA FOR CASE B – 2*5 – 2 COMPONENTS

| PCB 01 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 47506 | 12308 | 1 | 100 | 4 |
| 2 | 11556 | 15838 | 1 | 100 | 4 |
| 3 | 30342 | 18436 | 2 | 157 | 8 |
| 4 | 24300 | 14764 | 1 | 100 | 4 |
| 5 | 44564 | 3526 | 1 | 100 | 4 |
| 6 | 38104 | 8114 | 1 | 100 | 4 |
| 7 | 22824 | 18710 | 1 | 100 | 4 |
| 8 | 926 | 18338 | 2 | 157 | 8 |
| 9 | 41070 | 8206 | 1 | 100 | 4 |
| 10 | 22236 | 17872 | 1 | 100 | 4 |

| PCB 02 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 764 | 16762 | 1 | 100 | 4 |
| 2 | 37340 | 392 | 2 | 157 | 8 |
| 3 | 22254 | 13626 | 1 | 100 | 4 |
| 4 | 46590 | 7590 | 2 | 157 | 8 |
| 5 | 23300 | 16636 | 1 | 100 | 4 |
| 6 | 20932 | 10056 | 2 | 157 | 8 |
| 7 | 42312 | 14190 | 1 | 100 | 4 |
| 8 | 26258 | 8578 | 2 | 157 | 8 |
| 9 | 10132 | 6092 | 2 | 157 | 8 |
| 10 | 33606 | 3794 | 2 | 157 | 8 |

| PCB 03 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 24828 | 14542 | 2 | 157 | 8 |
| 2 | 44988 | 6186 | 2 | 157 | 8 |
| 3 | 41082 | 16770 | 2 | 157 | 8 |
| 4 | 32246 | 11362 | 2 | 157 | 8 |
| 5 | 40898 | 7408 | 1 | 100 | 4 |
| 6 | 33012 | 14054 | 2 | 157 | 8 |
| 7 | 17098 | 10932 | 1 | 100 | 4 |
| 8 | 14486 | 8898 | 1 | 100 | 4 |
| 9 | 17060 | 13892 | 2 | 157 | 8 |
| 10 | 26704 | 12426 | 2 | 157 | 8 |

| PCB 04 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 6826 | 11656 | 1 | 100 | 4 |
| 2 | 588 | 8470 | 1 | 100 | 4 |
| 3 | 44694 | 10310 | 2 | 157 | 8 |
| 4 | 9956 | 6680 | 2 | 157 | 8 |
| 5 | 14936 | 8658 | 1 | 100 | 4 |
| 6 | 33072 | 4518 | 2 | 157 | 8 |
| 7 | 14220 | 11596 | 2 | 157 | 8 |
| 8 | 23462 | 15208 | 1 | 100 | 4 |
| 9 | 3240 | 10596 | 2 | 157 | 8 |
| 10 | 49416 | 12810 | 1 | 100 | 4 |

| PCB 05 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 20768 | 4280 | 2 | 157 | 8 |
| 2 | 15250 | 12870 | 1 | 100 | 4 |
| 3 | 43718 | 6400 | 2 | 157 | 8 |
| 4 | 750 | 19202 | 2 | 157 | 8 |
| 5 | 38398 | 14532 | 1 | 100 | 4 |
| 6 | 48542 | 8240 | 1 | 100 | 4 |
| 7 | 49504 | 14892 | 2 | 157 | 8 |
| 8 | 39444 | 5358 | 2 | 157 | 8 |
| 9 | 21932 | 8798 | 1 | 100 | 4 |
| 10 | 24916 | 18668 | 2 | 157 | 8 |

| PCB 06 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 22572 | 12170 | 1 | 100 | 4 |
| 2 | 2194 | 316 | 1 | 100 | 4 |
| 3 | 1360 | 328 | 1 | 100 | 4 |
| 4 | 15634 | 3802 | 1 | 100 | 4 |
| 5 | 644 | 11738 | 1 | 100 | 4 |
| 6 | 19198 | 1152 | 2 | 157 | 8 |
| 7 | 34156 | 7352 | 2 | 157 | 8 |
| 8 | 4642 | 12630 | 2 | 157 | 8 |
| 9 | 1766 | 14352 | 1 | 100 | 4 |
| 10 | 30620 | 13854 | 2 | 157 | 8 |

| PCB 07 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 6052 | 4638 | 1 | 100 | 4 |
| 2 | 22538 | 4786 | 1 | 100 | 4 |
| 3 | 35794 | 996 | 1 | 100 | 4 |
| 4 | 44642 | 1568 | 1 | 100 | 4 |
| 5 | 13656 | 12816 | 1 | 100 | 4 |
| 6 | 12738 | 3818 | 2 | 157 | 8 |
| 7 | 43280 | 16878 | 1 | 100 | 4 |
| 8 | 11618 | 3478 | 1 | 100 | 4 |
| 9 | 40244 | 3416 | 1 | 100 | 4 |
| 10 | 45420 | 19886 | 2 | 157 | 8 |

| PCB 08 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 46712 | 2740 | 1 | 100 | 4 |
| 2 | 13222 | 16376 | 2 | 157 | 8 |
| 3 | 8016 | 8604 | 1 | 100 | 4 |
| 4 | 43642 | 17806 | 2 | 157 | 8 |
| 5 | 11894 | 14698 | 1 | 100 | 4 |
| 6 | 32292 | 13746 | 1 | 100 | 4 |
| 7 | 48344 | 6922 | 1 | 100 | 4 |
| 8 | 33246 | 3320 | 1 | 100 | 4 |
| 9 | 43520 | 3112 | 2 | 157 | 8 |
| 10 | 496 | 3822 | 1 | 100 | 4 |

| PCB 09 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 14870 | 7518 | 1 | 100 | 4 |
| 2 | 2458 | 198 | 2 | 157 | 8 |
| 3 | 34660 | 8398 | 2 | 157 | 8 |
| 4 | 32506 | 15074 | 2 | 157 | 8 |
| 5 | 49150 | 15878 | 1 | 100 | 4 |
| 6 | 27634 | 18400 | 2 | 157 | 8 |
| 7 | 20004 | 16894 | 2 | 157 | 8 |
| 8 | 9940 | 7356 | 1 | 100 | 4 |
| 9 | 31260 | 12416 | 2 | 157 | 8 |
| 10 | 36668 | 14626 | 1 | 100 | 4 |

| PCB 10 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 31366 | 14330 | 1 | 100 | 4 |
| 2 | 34954 | 10226 | 2 | 157 | 8 |
| 3 | 19860 | 15528 | 1 | 100 | 4 |
| 4 | 20682 | 9786 | 1 | 100 | 4 |
| 5 | 32760 | 3718 | 2 | 157 | 8 |
| 6 | 41880 | 14012 | 2 | 157 | 8 |
| 7 | 18580 | 19654 | 2 | 157 | 8 |
| 8 | 21262 | 16132 | 2 | 157 | 8 |
| 9 | 29734 | 14072 | 2 | 157 | 8 |
| 10 | 28286 | 9700 | 1 | 100 | 4 |

### Table G.9. PCB DATA FOR CASE B – 10*25 – 2 COMPONENTS

PCB 01

| $l$ | X | Y | $c_l$ | $r_g$ | Feeder |
|---|---|---|---|---|---|
| 1 | 237530 | 61540 | 1 | 100 | 4 |
| 2 | 57780 | 79190 | 1 | 100 | 4 |
| 3 | 151710 | 92180 | 2 | 157 | 8 |
| 4 | 121500 | 73820 | 1 | 100 | 4 |
| 5 | 222820 | 17630 | 1 | 100 | 4 |
| 6 | 190520 | 40570 | 1 | 100 | 4 |
| 7 | 114120 | 93550 | 1 | 100 | 4 |
| 8 | 4630 | 91690 | 2 | 157 | 8 |
| 9 | 205350 | 41030 | 1 | 100 | 4 |
| 10 | 111180 | 89360 | 1 | 100 | 4 |

PCB 02

| $l$ | X | Y | $c_l$ | $r_g$ | Feeder |
|---|---|---|---|---|---|
| 1 | 3820 | 83810 | 1 | 100 | 4 |
| 2 | 186700 | 1960 | 2 | 157 | 8 |
| 3 | 111270 | 68130 | 1 | 100 | 4 |
| 4 | 232950 | 37950 | 2 | 157 | 8 |
| 5 | 116500 | 83180 | 1 | 100 | 4 |
| 6 | 104660 | 50280 | 2 | 157 | 8 |
| 7 | 211560 | 70950 | 1 | 100 | 4 |
| 8 | 131290 | 42890 | 2 | 157 | 8 |
| 9 | 50660 | 30460 | 2 | 157 | 8 |
| 10 | 168030 | 18970 | 2 | 157 | 8 |

PCB 03

| $l$ | X | Y | $c_l$ | $r_g$ | Feeder |
|---|---|---|---|---|---|
| 1 | 124140 | 72710 | 2 | 157 | 8 |
| 2 | 224940 | 30930 | 2 | 157 | 8 |
| 3 | 205410 | 83850 | 2 | 157 | 8 |
| 4 | 161230 | 56810 | 2 | 157 | 8 |
| 5 | 204490 | 37040 | 1 | 100 | 4 |
| 6 | 165060 | 70270 | 2 | 157 | 8 |
| 7 | 85490 | 54660 | 1 | 100 | 4 |
| 8 | 72430 | 44490 | 1 | 100 | 4 |
| 9 | 85300 | 69460 | 2 | 157 | 8 |
| 10 | 133520 | 62130 | 2 | 157 | 8 |

PCB 04

| $l$ | X | Y | $c_l$ | $r_g$ | Feeder |
|---|---|---|---|---|---|
| 1 | 34130 | 58280 | 1 | 100 | 4 |
| 2 | 2940 | 42350 | 1 | 100 | 4 |
| 3 | 223470 | 51550 | 2 | 157 | 8 |
| 4 | 49780 | 33400 | 2 | 157 | 8 |
| 5 | 74680 | 43290 | 1 | 100 | 4 |
| 6 | 165360 | 22590 | 2 | 157 | 8 |
| 7 | 71100 | 57980 | 2 | 157 | 8 |
| 8 | 117310 | 76040 | 1 | 100 | 4 |
| 9 | 16200 | 52980 | 2 | 157 | 8 |
| 10 | 247080 | 64050 | 1 | 100 | 4 |

PCB 05

| $l$ | X | Y | $c_l$ | $r_g$ | Feeder |
|---|---|---|---|---|---|
| 1 | 103840 | 21400 | 2 | 157 | 8 |
| 2 | 76250 | 64350 | 1 | 100 | 4 |
| 3 | 218590 | 32000 | 2 | 157 | 8 |
| 4 | 3750 | 96010 | 2 | 157 | 8 |
| 5 | 191990 | 72660 | 1 | 100 | 4 |
| 6 | 242710 | 41200 | 1 | 100 | 4 |
| 7 | 247520 | 74460 | 2 | 157 | 8 |
| 8 | 197220 | 26790 | 2 | 157 | 8 |
| 9 | 109660 | 43990 | 1 | 100 | 4 |
| 10 | 124580 | 93340 | 2 | 157 | 8 |

PCB 06

| $l$ | X | Y | $c_l$ | $r_g$ | Feeder |
|---|---|---|---|---|---|
| 1 | 112860 | 60850 | 1 | 100 | 4 |
| 2 | 10970 | 1580 | 1 | 100 | 4 |
| 3 | 6800 | 1640 | 1 | 100 | 4 |
| 4 | 78170 | 19010 | 1 | 100 | 4 |
| 5 | 3220 | 58690 | 1 | 100 | 4 |
| 6 | 95990 | 5760 | 2 | 157 | 8 |
| 7 | 170780 | 36760 | 2 | 157 | 8 |
| 8 | 23210 | 63150 | 2 | 157 | 8 |
| 9 | 8830 | 71760 | 1 | 100 | 4 |
| 10 | 153100 | 69270 | 2 | 157 | 8 |

PCB 07

| $l$ | X | Y | $c_l$ | $r_g$ | Feeder |
|---|---|---|---|---|---|
| 1 | 30260 | 23190 | 1 | 100 | 4 |
| 2 | 112690 | 23930 | 1 | 100 | 4 |
| 3 | 178970 | 4980 | 1 | 100 | 4 |
| 4 | 223210 | 7840 | 1 | 100 | 4 |
| 5 | 68280 | 64080 | 1 | 100 | 4 |
| 6 | 63690 | 19090 | 2 | 157 | 8 |
| 7 | 216400 | 84390 | 1 | 100 | 4 |
| 8 | 58090 | 17390 | 1 | 100 | 4 |
| 9 | 201220 | 17080 | 1 | 100 | 4 |
| 10 | 227100 | 99430 | 2 | 157 | 8 |

PCB 08

| $l$ | X | Y | $c_l$ | $r_g$ | Feeder |
|---|---|---|---|---|---|
| 1 | 233560 | 13700 | 1 | 100 | 4 |
| 2 | 66110 | 81880 | 2 | 157 | 8 |
| 3 | 40080 | 43020 | 1 | 100 | 4 |
| 4 | 218210 | 89030 | 2 | 157 | 8 |
| 5 | 59470 | 73490 | 1 | 100 | 4 |
| 6 | 161460 | 68730 | 1 | 100 | 4 |
| 7 | 241720 | 34610 | 1 | 100 | 4 |
| 8 | 166230 | 16600 | 1 | 100 | 4 |
| 9 | 217600 | 15560 | 2 | 157 | 8 |
| 10 | 2480 | 19110 | 1 | 100 | 4 |

PCB 09

| $l$ | X | Y | $c_l$ | $r_g$ | Feeder |
|---|---|---|---|---|---|
| 1 | 74350 | 37590 | 1 | 100 | 4 |
| 2 | 12290 | 990 | 2 | 157 | 8 |
| 3 | 173300 | 41990 | 2 | 157 | 8 |
| 4 | 162530 | 75370 | 2 | 157 | 8 |
| 5 | 245750 | 79390 | 1 | 100 | 4 |
| 6 | 138170 | 92000 | 2 | 157 | 8 |
| 7 | 100020 | 84470 | 2 | 157 | 8 |
| 8 | 49700 | 36780 | 1 | 100 | 4 |
| 9 | 156300 | 62080 | 2 | 157 | 8 |
| 10 | 183340 | 73130 | 1 | 100 | 4 |

PCB 10

| $l$ | X | Y | $c_l$ | $r_g$ | Feeder |
|---|---|---|---|---|---|
| 1 | 156830 | 71650 | 1 | 100 | 4 |
| 2 | 174770 | 51130 | 2 | 157 | 8 |
| 3 | 99300 | 77640 | 1 | 100 | 4 |
| 4 | 103410 | 48930 | 1 | 100 | 4 |
| 5 | 163800 | 18590 | 2 | 157 | 8 |
| 6 | 209400 | 70060 | 2 | 157 | 8 |
| 7 | 92900 | 98270 | 2 | 157 | 8 |
| 8 | 106310 | 80660 | 2 | 157 | 8 |
| 9 | 148670 | 70360 | 2 | 157 | 8 |
| 10 | 141430 | 48500 | 1 | 100 | 4 |

# Table G.10. PCB DATA FOR CASE B – 1*2.5 – 4 COMPONENTS

PCB 01

| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
|---|---|---|---|---|---|
| 1 | 23753 | 6154 | 1 | 100 | 4 |
| 2 | 5778 | 7919 | 2 | 157 | 8 |
| 3 | 15171 | 9218 | 4 | 271 | 16 |
| 4 | 12150 | 7382 | 1 | 100 | 4 |
| 5 | 22282 | 1763 | 1 | 100 | 4 |
| 6 | 19052 | 4057 | 1 | 100 | 4 |
| 7 | 11412 | 9355 | 1 | 100 | 4 |
| 8 | 463 | 9169 | 3 | 214 | 12 |
| 9 | 20535 | 4103 | 2 | 157 | 8 |
| 10 | 11118 | 8936 | 1 | 100 | 4 |

PCB 02

| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
|---|---|---|---|---|---|
| 1 | 382 | 8381 | 1 | 100 | 4 |
| 2 | 18670 | 196 | 1 | 100 | 4 |
| 3 | 11127 | 6813 | 3 | 214 | 12 |
| 4 | 23295 | 3795 | 3 | 214 | 12 |
| 5 | 11650 | 8318 | 2 | 157 | 8 |
| 6 | 10466 | 5028 | 2 | 157 | 8 |
| 7 | 21156 | 7095 | 3 | 214 | 12 |
| 8 | 13129 | 4289 | 4 | 271 | 16 |
| 9 | 5066 | 3046 | 4 | 271 | 16 |
| 10 | 16803 | 1897 | 3 | 214 | 12 |

PCB 03

| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
|---|---|---|---|---|---|
| 1 | 12414 | 7271 | 4 | 271 | 16 |
| 2 | 22494 | 3093 | 4 | 271 | 16 |
| 3 | 20541 | 8385 | 3 | 214 | 12 |
| 4 | 16123 | 5681 | 4 | 271 | 16 |
| 5 | 20449 | 3704 | 1 | 100 | 4 |
| 6 | 16506 | 7027 | 4 | 271 | 16 |
| 7 | 8549 | 5466 | 2 | 157 | 8 |
| 8 | 7243 | 4449 | 2 | 157 | 8 |
| 9 | 8530 | 6946 | 4 | 271 | 16 |
| 10 | 13352 | 6213 | 3 | 214 | 12 |

PCB 04

| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
|---|---|---|---|---|---|
| 1 | 3413 | 5828 | 1 | 100 | 4 |
| 2 | 294 | 4235 | 2 | 157 | 8 |
| 3 | 22347 | 5155 | 4 | 271 | 16 |
| 4 | 4978 | 3340 | 3 | 214 | 12 |
| 5 | 7468 | 4329 | 2 | 157 | 8 |
| 6 | 16536 | 2259 | 3 | 214 | 12 |
| 7 | 7110 | 5798 | 4 | 271 | 16 |
| 8 | 11731 | 7604 | 1 | 100 | 4 |
| 9 | 1620 | 5298 | 3 | 214 | 12 |
| 10 | 24708 | 6405 | 1 | 100 | 4 |

PCB 05

| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
|---|---|---|---|---|---|
| 1 | 10384 | 2140 | 3 | 214 | 12 |
| 2 | 7625 | 6435 | 1 | 100 | 4 |
| 3 | 21859 | 3200 | 4 | 271 | 16 |
| 4 | 375 | 9601 | 3 | 214 | 12 |
| 5 | 19199 | 7266 | 1 | 100 | 4 |
| 6 | 24271 | 4120 | 1 | 100 | 4 |
| 7 | 24752 | 7446 | 3 | 214 | 12 |
| 8 | 19722 | 2679 | 3 | 214 | 12 |
| 9 | 10966 | 4399 | 2 | 157 | 8 |
| 10 | 12458 | 9334 | 3 | 214 | 12 |

PCB 06

| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
|---|---|---|---|---|---|
| 1 | 11286 | 6085 | 1 | 100 | 4 |
| 2 | 1097 | 158 | 2 | 157 | 8 |
| 3 | 680 | 164 | 2 | 157 | 8 |
| 4 | 7817 | 1901 | 2 | 157 | 8 |
| 5 | 322 | 5869 | 1 | 100 | 4 |
| 6 | 9599 | 576 | 3 | 214 | 12 |
| 7 | 17078 | 3676 | 3 | 214 | 12 |
| 8 | 2321 | 6315 | 3 | 214 | 12 |
| 9 | 883 | 7176 | 2 | 157 | 8 |
| 10 | 15310 | 6927 | 4 | 271 | 16 |

PCB 07

| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
|---|---|---|---|---|---|
| 1 | 3026 | 2319 | 2 | 157 | 8 |
| 2 | 11269 | 2393 | 2 | 157 | 8 |
| 3 | 17897 | 498 | 2 | 157 | 8 |
| 4 | 22321 | 784 | 2 | 157 | 8 |
| 5 | 6828 | 6408 | 2 | 157 | 8 |
| 6 | 6369 | 1909 | 3 | 214 | 12 |
| 7 | 21640 | 8439 | 1 | 100 | 4 |
| 8 | 5809 | 1739 | 1 | 100 | 4 |
| 9 | 20122 | 1708 | 2 | 157 | 8 |
| 10 | 22710 | 9943 | 4 | 271 | 16 |

PCB 08

| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
|---|---|---|---|---|---|
| 1 | 23356 | 1370 | 3 | 214 | 12 |
| 2 | 6611 | 8188 | 4 | 271 | 16 |
| 3 | 4008 | 4302 | 2 | 157 | 8 |
| 4 | 21821 | 8903 | 4 | 271 | 16 |
| 5 | 5947 | 7349 | 2 | 157 | 8 |
| 6 | 16146 | 6873 | 2 | 157 | 8 |
| 7 | 24172 | 3461 | 2 | 157 | 8 |
| 8 | 16623 | 1660 | 2 | 157 | 8 |
| 9 | 21760 | 1556 | 4 | 271 | 16 |
| 10 | 248 | 1911 | 1 | 100 | 4 |

PCB 09

| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
|---|---|---|---|---|---|
| 1 | 7435 | 3759 | 1 | 100 | 4 |
| 2 | 1229 | 99 | 4 | 271 | 16 |
| 3 | 17330 | 4199 | 3 | 214 | 12 |
| 4 | 16253 | 7537 | 3 | 214 | 12 |
| 5 | 24575 | 7939 | 1 | 100 | 4 |
| 6 | 13817 | 9200 | 3 | 214 | 12 |
| 7 | 10002 | 8447 | 4 | 271 | 16 |
| 8 | 4970 | 3678 | 2 | 157 | 8 |
| 9 | 15630 | 6208 | 3 | 214 | 12 |
| 10 | 18334 | 7313 | 2 | 157 | 8 |

PCB 10

| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
|---|---|---|---|---|---|
| 1 | 15683 | 7165 | 1 | 100 | 4 |
| 2 | 17477 | 5113 | 3 | 214 | 12 |
| 3 | 9930 | 7764 | 2 | 157 | 8 |
| 4 | 10341 | 4893 | 1 | 100 | 4 |
| 5 | 16380 | 1859 | 3 | 214 | 12 |
| 6 | 20940 | 7006 | 4 | 271 | 16 |
| 7 | 9290 | 9827 | 3 | 214 | 12 |
| 8 | 10631 | 8066 | 4 | 271 | 16 |
| 9 | 14867 | 7036 | 4 | 271 | 16 |
| 10 | 14143 | 4850 | 1 | 100 | 4 |

# Table G.11. PCB DATA FOR CASE B – 2*5 – 4 COMPONENTS

| PCB 01 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 47506 | 12308 | 1 | 100 | 4 |
| 2 | 11556 | 15838 | 2 | 157 | 8 |
| 3 | 30342 | 18436 | 4 | 271 | 16 |
| 4 | 24300 | 14764 | 1 | 100 | 4 |
| 5 | 44564 | 3526 | 1 | 100 | 4 |
| 6 | 38104 | 8114 | 1 | 100 | 4 |
| 7 | 22824 | 18710 | 1 | 100 | 4 |
| 8 | 926 | 18338 | 3 | 214 | 12 |
| 9 | 41070 | 8206 | 2 | 157 | 8 |
| 10 | 22236 | 17872 | 1 | 100 | 4 |

| PCB 02 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 764 | 16762 | 1 | 100 | 4 |
| 2 | 37340 | 392 | 1 | 100 | 4 |
| 3 | 22254 | 13626 | 3 | 214 | 12 |
| 4 | 46590 | 7590 | 3 | 214 | 12 |
| 5 | 23300 | 16636 | 2 | 157 | 8 |
| 6 | 20932 | 10056 | 2 | 157 | 8 |
| 7 | 42312 | 14190 | 3 | 214 | 12 |
| 8 | 26258 | 8578 | 4 | 271 | 16 |
| 9 | 10132 | 6092 | 4 | 271 | 16 |
| 10 | 33606 | 3794 | 3 | 214 | 12 |

| PCB 03 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 24828 | 14542 | 4 | 271 | 16 |
| 2 | 44988 | 6186 | 4 | 271 | 16 |
| 3 | 41082 | 16770 | 3 | 214 | 12 |
| 4 | 32246 | 11362 | 4 | 271 | 16 |
| 5 | 40898 | 7408 | 1 | 100 | 4 |
| 6 | 33012 | 14054 | 4 | 271 | 16 |
| 7 | 17098 | 10932 | 2 | 157 | 8 |
| 8 | 14486 | 8898 | 2 | 157 | 8 |
| 9 | 17060 | 13892 | 4 | 271 | 16 |
| 10 | 26704 | 12426 | 3 | 214 | 12 |

| PCB 04 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 6826 | 11656 | 1 | 100 | 4 |
| 2 | 588 | 8470 | 2 | 157 | 8 |
| 3 | 44694 | 10310 | 4 | 271 | 16 |
| 4 | 9956 | 6680 | 3 | 214 | 12 |
| 5 | 14936 | 8658 | 2 | 157 | 8 |
| 6 | 33072 | 4518 | 3 | 214 | 12 |
| 7 | 14220 | 11596 | 4 | 271 | 16 |
| 8 | 23462 | 15208 | 1 | 100 | 4 |
| 9 | 3240 | 10596 | 3 | 214 | 12 |
| 10 | 49416 | 12810 | 1 | 100 | 4 |

| PCB 05 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 20768 | 4280 | 3 | 214 | 12 |
| 2 | 15250 | 12870 | 1 | 100 | 4 |
| 3 | 43718 | 6400 | 4 | 271 | 16 |
| 4 | 750 | 19202 | 3 | 214 | 12 |
| 5 | 38398 | 14532 | 1 | 100 | 4 |
| 6 | 48542 | 8240 | 1 | 100 | 4 |
| 7 | 49504 | 14892 | 3 | 214 | 12 |
| 8 | 39444 | 5358 | 3 | 214 | 12 |
| 9 | 21932 | 8798 | 2 | 157 | 8 |
| 10 | 24916 | 18668 | 3 | 214 | 12 |

| PCB 06 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 22572 | 12170 | 1 | 100 | 4 |
| 2 | 2194 | 316 | 2 | 157 | 8 |
| 3 | 1360 | 328 | 2 | 157 | 8 |
| 4 | 15634 | 3802 | 2 | 157 | 8 |
| 5 | 644 | 11738 | 1 | 100 | 4 |
| 6 | 19198 | 1152 | 3 | 214 | 12 |
| 7 | 34156 | 7352 | 3 | 214 | 12 |
| 8 | 4642 | 12630 | 3 | 214 | 12 |
| 9 | 1766 | 14352 | 2 | 157 | 8 |
| 10 | 30620 | 13854 | 4 | 271 | 16 |

| PCB 07 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 6052 | 4638 | 2 | 157 | 8 |
| 2 | 22538 | 4786 | 2 | 157 | 8 |
| 3 | 35794 | 996 | 2 | 157 | 8 |
| 4 | 44642 | 1568 | 2 | 157 | 8 |
| 5 | 13656 | 12816 | 2 | 157 | 8 |
| 6 | 12738 | 3818 | 3 | 214 | 12 |
| 7 | 43280 | 16878 | 1 | 100 | 4 |
| 8 | 11618 | 3478 | 1 | 100 | 4 |
| 9 | 40244 | 3416 | 2 | 157 | 8 |
| 10 | 45420 | 19886 | 4 | 271 | 16 |

| PCB 08 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 46712 | 2740 | 3 | 214 | 12 |
| 2 | 13222 | 16376 | 4 | 271 | 16 |
| 3 | 8016 | 8604 | 2 | 157 | 8 |
| 4 | 43642 | 17806 | 4 | 271 | 16 |
| 5 | 11894 | 14698 | 2 | 157 | 8 |
| 6 | 32292 | 13746 | 2 | 157 | 8 |
| 7 | 48344 | 6922 | 2 | 157 | 8 |
| 8 | 33246 | 3320 | 2 | 157 | 8 |
| 9 | 43520 | 3112 | 4 | 271 | 16 |
| 10 | 496 | 3822 | 1 | 100 | 4 |

| PCB 09 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 14870 | 7518 | 1 | 100 | 4 |
| 2 | 2458 | 198 | 4 | 271 | 16 |
| 3 | 34660 | 8398 | 3 | 214 | 12 |
| 4 | 32506 | 15074 | 3 | 214 | 12 |
| 5 | 49150 | 15878 | 1 | 100 | 4 |
| 6 | 27634 | 18400 | 3 | 214 | 12 |
| 7 | 20004 | 16894 | 4 | 271 | 16 |
| 8 | 9940 | 7356 | 2 | 157 | 8 |
| 9 | 31260 | 12416 | 3 | 214 | 12 |
| 10 | 36668 | 14626 | 2 | 157 | 8 |

| PCB 10 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $q_l$ | $r_q$ | Feeder |
| 1 | 31366 | 14330 | 1 | 100 | 4 |
| 2 | 34954 | 10226 | 3 | 214 | 12 |
| 3 | 19860 | 15528 | 2 | 157 | 8 |
| 4 | 20682 | 9786 | 1 | 100 | 4 |
| 5 | 32760 | 3718 | 3 | 214 | 12 |
| 6 | 41880 | 14012 | 4 | 271 | 16 |
| 7 | 18580 | 19654 | 3 | 214 | 12 |
| 8 | 21262 | 16132 | 4 | 271 | 16 |
| 9 | 29734 | 14072 | 4 | 271 | 16 |
| 10 | 28286 | 9700 | 1 | 100 | 4 |

# Table G.12. PCB DATA FOR CASE B – 10*25 – 4 COMPONENTS

| PCB 01 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $g_l$ | $r_g$ | Feeder |
| 1 | 237530 | 61540 | 1 | 100 | 4 |
| 2 | 57780 | 79190 | 2 | 157 | 8 |
| 3 | 151710 | 92180 | 4 | 271 | 16 |
| 4 | 121500 | 73820 | 1 | 100 | 4 |
| 5 | 222820 | 17630 | 1 | 100 | 4 |
| 6 | 190520 | 40570 | 1 | 100 | 4 |
| 7 | 114120 | 93550 | 1 | 100 | 4 |
| 8 | 4630 | 91690 | 3 | 214 | 12 |
| 9 | 205350 | 41030 | 2 | 157 | 8 |
| 10 | 111180 | 89360 | 1 | 100 | 4 |

| PCB 02 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $g_l$ | $r_g$ | Feeder |
| 1 | 3820 | 83810 | 1 | 100 | 4 |
| 2 | 186700 | 1960 | 1 | 100 | 4 |
| 3 | 111270 | 68130 | 3 | 214 | 12 |
| 4 | 232950 | 37950 | 3 | 214 | 12 |
| 5 | 116500 | 83180 | 2 | 157 | 8 |
| 6 | 104660 | 50280 | 2 | 157 | 8 |
| 7 | 211560 | 70950 | 3 | 214 | 12 |
| 8 | 131290 | 42890 | 4 | 271 | 16 |
| 9 | 50660 | 30460 | 4 | 271 | 16 |
| 10 | 168030 | 18970 | 3 | 214 | 12 |

| PCB 03 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $g_l$ | $r_g$ | Feeder |
| 1 | 124140 | 72710 | 4 | 271 | 16 |
| 2 | 224940 | 30930 | 4 | 271 | 16 |
| 3 | 205410 | 83850 | 3 | 214 | 12 |
| 4 | 161230 | 56810 | 4 | 271 | 16 |
| 5 | 204490 | 37040 | 1 | 100 | 4 |
| 6 | 165060 | 70270 | 4 | 271 | 16 |
| 7 | 85490 | 54660 | 2 | 157 | 8 |
| 8 | 72430 | 44490 | 2 | 157 | 8 |
| 9 | 85300 | 69460 | 4 | 271 | 16 |
| 10 | 133520 | 62130 | 3 | 214 | 12 |

| PCB 04 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $g_l$ | $r_g$ | Feeder |
| 1 | 34130 | 58280 | 1 | 100 | 4 |
| 2 | 2940 | 42350 | 2 | 157 | 8 |
| 3 | 223470 | 51550 | 4 | 271 | 16 |
| 4 | 49780 | 33400 | 3 | 214 | 12 |
| 5 | 74680 | 43290 | 2 | 157 | 8 |
| 6 | 165360 | 22590 | 3 | 214 | 12 |
| 7 | 71100 | 57980 | 4 | 271 | 16 |
| 8 | 117310 | 76040 | 1 | 100 | 4 |
| 9 | 16200 | 52980 | 3 | 214 | 12 |
| 10 | 247080 | 64050 | 1 | 100 | 4 |

| PCB 05 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $g_l$ | $r_g$ | Feeder |
| 1 | 103840 | 21400 | 3 | 214 | 12 |
| 2 | 76250 | 64350 | 1 | 100 | 4 |
| 3 | 218590 | 32000 | 4 | 271 | 16 |
| 4 | 3750 | 96010 | 3 | 214 | 12 |
| 5 | 191990 | 72660 | 1 | 100 | 4 |
| 6 | 242710 | 41200 | 1 | 100 | 4 |
| 7 | 247520 | 74460 | 3 | 214 | 12 |
| 8 | 197220 | 26790 | 3 | 214 | 12 |
| 9 | 109660 | 43990 | 2 | 157 | 8 |
| 10 | 124580 | 93340 | 3 | 214 | 12 |

| PCB 06 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $g_l$ | $r_g$ | Feeder |
| 1 | 112860 | 60850 | 1 | 100 | 4 |
| 2 | 10970 | 1580 | 2 | 157 | 8 |
| 3 | 6800 | 1640 | 2 | 157 | 8 |
| 4 | 78170 | 19010 | 2 | 157 | 8 |
| 5 | 3220 | 58690 | 1 | 100 | 4 |
| 6 | 95990 | 5760 | 3 | 214 | 12 |
| 7 | 170780 | 36760 | 3 | 214 | 12 |
| 8 | 23210 | 63150 | 3 | 214 | 12 |
| 9 | 8830 | 71760 | 2 | 157 | 8 |
| 10 | 153100 | 69270 | 4 | 271 | 16 |

| PCB 07 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $g_l$ | $r_g$ | Feeder |
| 1 | 30260 | 23190 | 2 | 157 | 8 |
| 2 | 112690 | 23930 | 2 | 157 | 8 |
| 3 | 178970 | 4980 | 2 | 157 | 8 |
| 4 | 223210 | 7840 | 2 | 157 | 8 |
| 5 | 68280 | 64080 | 2 | 157 | 8 |
| 6 | 63690 | 19090 | 3 | 214 | 12 |
| 7 | 216400 | 84390 | 1 | 100 | 4 |
| 8 | 58090 | 17390 | 1 | 100 | 4 |
| 9 | 201220 | 17080 | 2 | 157 | 8 |
| 10 | 227100 | 99430 | 4 | 271 | 16 |

| PCB 08 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $g_l$ | $r_g$ | Feeder |
| 1 | 233560 | 13700 | 3 | 214 | 12 |
| 2 | 66110 | 81880 | 4 | 271 | 16 |
| 3 | 40080 | 43020 | 2 | 157 | 8 |
| 4 | 218210 | 89030 | 4 | 271 | 16 |
| 5 | 59470 | 73490 | 2 | 157 | 8 |
| 6 | 161460 | 68730 | 2 | 157 | 8 |
| 7 | 241720 | 34610 | 2 | 157 | 8 |
| 8 | 166230 | 16600 | 2 | 157 | 8 |
| 9 | 217600 | 15560 | 4 | 271 | 16 |
| 10 | 2480 | 19110 | 1 | 100 | 4 |

| PCB 09 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $g_l$ | $r_g$ | Feeder |
| 1 | 74350 | 37590 | 1 | 100 | 4 |
| 2 | 12290 | 990 | 4 | 271 | 16 |
| 3 | 173300 | 41990 | 3 | 214 | 12 |
| 4 | 162530 | 75370 | 3 | 214 | 12 |
| 5 | 245750 | 79390 | 1 | 100 | 4 |
| 6 | 138170 | 92000 | 3 | 214 | 12 |
| 7 | 100020 | 84470 | 4 | 271 | 16 |
| 8 | 49700 | 36780 | 2 | 157 | 8 |
| 9 | 156300 | 62080 | 3 | 214 | 12 |
| 10 | 183340 | 73130 | 2 | 157 | 8 |

| PCB 10 | | | | | |
|---|---|---|---|---|---|
| $l$ | X | Y | $g_l$ | $r_g$ | Feeder |
| 1 | 156830 | 71650 | 1 | 100 | 4 |
| 2 | 174770 | 51130 | 3 | 214 | 12 |
| 3 | 99300 | 77640 | 2 | 157 | 8 |
| 4 | 103410 | 48930 | 1 | 100 | 4 |
| 5 | 163800 | 18590 | 3 | 214 | 12 |
| 6 | 209400 | 70060 | 4 | 271 | 16 |
| 7 | 92900 | 98270 | 3 | 214 | 12 |
| 8 | 106310 | 80660 | 4 | 271 | 16 |
| 9 | 148670 | 70360 | 4 | 271 | 16 |
| 10 | 141430 | 48500 | 1 | 100 | 4 |