New Approach to SMD Placement Optimization

New Approach to SMD Placement Optimization

Timo Liukkonen[1] and Aulis Tuominen[2]

[1]Nokia Corporation
Nokia Mobile Phones
Salo Operations
P.O. Box 86
24101 Salo
Finland

[2]Tampere University of Technology, Pori
P.O. Box 300
28101 Pori
Finland

Abstract

Placement optimization of printed wiring board for high-speed placement machines was studied from different angles of approach. Most commercial optimization software allocates components to various high-speed placement machines of the production line in a general way to get the placement work load between the machines as balanced as possible, paying most attention to part number distribution between the machines. A different approach is introduced to get the board area to be assembled as small as possible in each machine, and instead of part numbers, paying more attention to reference designators and coordinates. This method is based on a more careful study of the board layout prior to assembly. Arrangement of component feeder tables is also changed. Usually optimizers limit each part number to be allocated to only one feeder in the production line, but now multiple feeders for most of the part numbers are used in order to keep the placement area small in each machine. Commercial software systems do not yet support this kind of optimization, and therefore board layout is often considered too late, at a phase when part numbers have already been allocated to the machines, and final optimization of an individual machine is about to start. In the case study the new method was compared to two well-known and widely used optimization software packages, and promising results were seen in reduction of the placement time.

The International Journal of Microcircuits and Electronic Packaging, Volume 25, Number 3, 3rd Quarter, 2002 (ISSN 1063-1674)

Key words

SMD, pick-and-place, optimization

1. Introduction

Ultra high volume production lines used in today's electronics industry require the best possible methods to insure SMD (Surface Mounted Device) placement program optimization and line balancing. When data processor capacity gets doubled every 18 months, electronic products get more capacity for data processing as well [1]. This increases the number of components and the overall complexity of the printed wiring boards, increasing the requirements for the capacity growth of the production lines as well. Additional manufacturing capacity from optimization could have been used to produce other products, or produce the given products in less time, saving considerable production costs [2]. SMD Lines usually consist of multiple sequentially installed high-speed placement machines, placing components from 0402 chips to small integrated circuits, and at least one multi-functional machine placing *e.g.* connectors and high-accuracy circuits. Most of the components placed by multi-functional machines can't be placed reasonably by high-speed machines because of size, accuracy requirements, odd-shape form, and others. This makes it quite easy to distribute components between high-speed machines and multi-functional machines. It is much more difficult to distribute components between multiple high-speed machines.

Generally PWB (printed wiring board) assembly problems are divided into four different problem classes [3]:

1. One PWB type and one machine. *Single machine optimization* belongs to this class, subdivided into feeder setup problem and placement sequence problem.

2. One PWB type and many machines: the line-balancing problem for balancing the workload of one product among different machines.

3. Many different PWBs and one machine: the grouping problem for constructing common feeder-setups for several products.

4. Many different PWBs and many machines: the scheduling problem, which is subdivided to allocation of the PWBs to the machines and sequencing of the PWBs in the machines.

In this study we have concentrated only on ultra high volume lines i.e. on the first two classes, and on the high-speed machines.

Better optimization and line balancing can offer big economical opportunities. A carefully considered component placement sequence gives a shorter production time and increases the productivity, and the mechanical stress of the equipment is decreased because the movements become shorter [4]. Because of the minimized mechanical stress the machine can maintain its accuracy specification better, and thus produce with more stable placement quality.

2. Rotary turret type placement machines

A rotary turret type SMD placement machine has a moving XY-table, which transfers the PWB to correct position below the placement head during placement phase. Component feeders are arranged behind the machine in a feeder table, which transfers the correct feeder below the placement head during component pickup phase. Placement heads with various sizes of vacuum nozzles for component pickup are arranged in the turret, which revolves and moves the pickup nozzles from part pickup point to placement point in a continuous way. Figure 1 shows the operational principle of a turret machine.

2.1 Programming of multi-block PWBs

Multi-block PWBs, used typically in manufacturing of mobile phones, is a larger PWB panel composed of a number of smaller identical PWBs (see Figure 2). These smaller PWBs, called blocks, are arrayed by a certain distance (i.e. offset) from each other. Components can be assembled on this kind of multi-block PWB at least in three ways:

1. All blocks are placed as one big PWB and optimization may "move" freely between the blocks. The placement program includes coordinates for all the components of the multi-block PWB. This kind of assembly is often called "one-board" function.

2. Each block is placed separately, one after another. The placement program includes coordinates for the first block only, and the program is repeated at each offset. This kind of assembly is called "step-and-repeat" function.

3. Combination of types 1 and 2, so that a section of the PWB is placed using one-board function and the rest using step-and-repeat function

A problem in the first type of placement above can be too many movements between the blocks generated automatically by the optimization software in order to keep the amount of component feeders per part number as small as possible. This means that all components (i.e. circuit references) in a certain part number will be retrieved from the same feeder. In the second type the problem can be movement of the feeder table from the end to the beginning of it at start point of each block in order to repeat the placement steps.

2.2 Part distribution based on workload balancing

Usually part numbers have been distributed to feeders, and feeders to multiple high-speed machines through equal work load distribution. This means, that part numbers have been assigned to the machines, regardless of component positions in the PWB layout, so that placement time (the work load) is as balanced as possible between the machines. Each part number exists only in one feeder and in one machine of the line. This is the way in which the preliminary part distribution occurs in most commercial optimization software.

2.3 Part distribution based on board layout

In this new approach, we have studied the optimization of multi-block PWBs in high volume production. In most high-speed placement machine types the rotary turret revolver has multiple nozzles in each placing head for picking up components of different size and shape. In some machines the effective time required to change the nozzle is

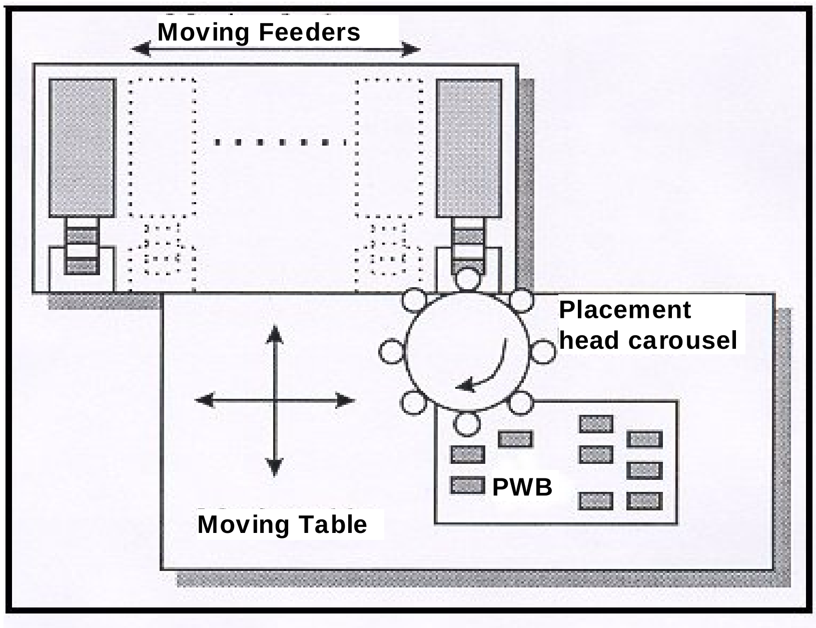

Figure 1. Rotary turret machine [4].

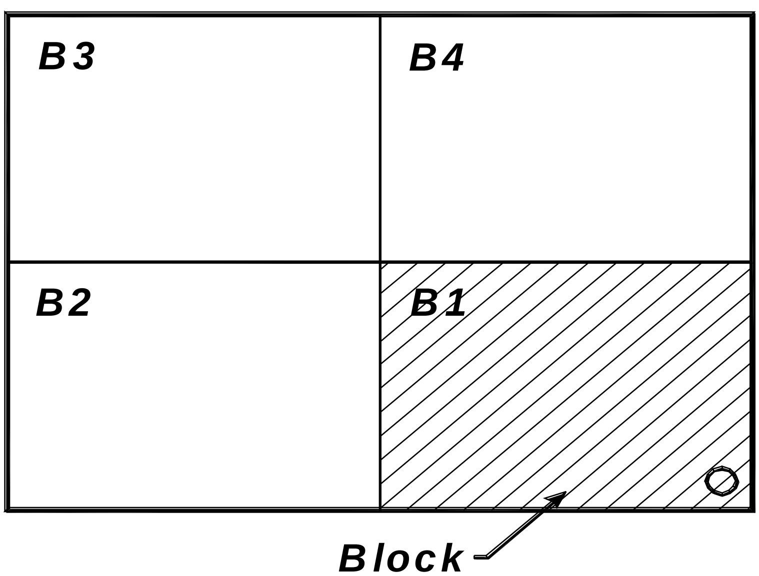

Figure 2. Example of a multi-block PWB.

always zero, or it has no effect to cycle time, if the nozzle is always changed to the one next to it. The Fuji CP6 series machine operates using the latter method.
In the study the optimization is started from the board layout. We distribute high speed components to different high-speed machines so that the first machine places only a small area at the bottom of the board, the second machine parts from the center and the next parts from top of the board, *etc.*, distributing parts from one PWB block into number of vertically located segments. The block area could, in case of different outer shapes, also be divided into horizontal segments in the same way. During optimization we try to keep the component count approximately the same in each machine.

2.4 Reverse sequence at each offset

When one block from the panel has been placed, we step to the next block and repeat the placement steps in reverse sequence to minimize any useless travel of the feeder table back and forth, *i.e*. we have to use reverse sequence at each offset. This means that the first PWB block is placed using feeder table slot numbers from 1 to 40, the second block from numbers 40 to 1, the third block from numbers 1 to 40, *etc.* This feature is essential if we want to place blocks separately, but it is not supported on some widely used software systems.

2.5 Multiple feeders for part numbers

In the study each machine places only a small area of the board, and because many part numbers are found in many areas of the board, from bottom, center or top, we have to place them by multiple machines in the production line, sometimes perhaps by all of them. This makes it almost impossible to use commercial optimization systems to distribute parts like we have done manually, because they limit the number of feeders per part number to one piece only.

2.6 High-speed machines placing parts with lower speed

Most high-speed parts are placed by the first placement machines and parts with lower speed are placed later in the line, so that placement speed slows down towards the end of the line.

When the first part with lower speed is picked up from the feeder and placement speed is accordingly slowed down, all those high-speed parts that have already been picked up, but are still on the way towards the placement position *i.e.* on half of the turret circle, are also automatically slowed down. This makes it reasonable to place all components with lower speed by the last high-speed machines in the line.

In some machines (*e.g*. Fuji CP6 series) certain placement rotations slow down the placement speed by up to 45%. Thus it was decided to place all parts with rotation of 180 degrees together by the last high-speed machines.

Some components were chosen to be placed at the end because, due to their physical dimensions, they require large nozzles and we wanted to keep the number of utilized nozzle types as small as possible in the first high-speed machines. Usually these components are so big and tend to move after placement that they have to be placed at the end anyway .

2.7 Final optimization of one machine program

Feeder arrangement and the final placement sequences are created manually using a graphical program editor that shows them both on the screen at the same time. During the sequencing we tried not to exceed the maximum allowed XY-table movement within maximum placement speed. To prevent too large XY-table movements the part numbers, components of which were located extensively on the placement area, were divided into two or three additional feeders based on XY-coordinates. Placement steps were written in reverse sequence at each block offset.

The program of the last high-speed machine placing the slowest components was divided into two sections: components assembled with max speed and components assembled with lower speed (caused by placement rotations, nozzle changes or parts physical dimensions). The first section was optimized with the step-and-repeat function and the second one with the one-board function. The one-board function was selected for the slowest components to limit turret speed deceleration to one time per panel.

3. Case study

A new product was programmed to a line consisting of six high-speed placement machines of the same speed and type, along with one multi-functional machine. At first components were chosen manually to the multi-functional machine based on their size, accuracy requirements and odd-shape form. Most of the components were assigned to be placed by the six high-speed machines using the manual methods and principles described in the previous sections. In Figure 3 we can see how these components were distributed: machine A places components from the bottom of each PWB block, machine B from the center, machine C from the top, machine D again from the bottom, machine E again from the top and finally machine F from the whole block area, because components with the lowest placement speed are located over the entire board.

Parts were distributed and programs optimized also by two well known, globally used, commercial software systems: Trilogy 5000 from Valor Computerized Systems (Finland) Oy and FujiCam from Tecnomatix Technologies Ltd. Trilogy is probably one of the most advanced systems used today and FujiCam is extremely widely used globally and has been used by Fuji for years. The only limitation made to these systems was that parts with lower speed should be placed by the last high-speed machine(s). In Figure 4 we can see the parts distribution generated by Trilogy. The difference in manual distribution is that components are spread extensively in all the machines. In Figure 5 we can see parts distribution generated by FujiCam. Here also the components are spread extensively in all the machines.

The final optimization is dominated by the selected optimization type: step-and-repeat function or one-board function. Figure 6 shows the final placement path made manually for machine A, using the step-and-repeat function, with each block assembled in reverse sequence compared to the previous one. The first five high-speed machines were optimized manually using the step-and-repeat function. Figures 7 and 8 show the final sequence made manually for the last high-speed machine, first for the components with maximum speed using step-and-repeat function and then for the components with lower speed using the one-board function. Figure 9 shows the placement sequence generated by Trilogy for machine A, using the one-board-function, and the

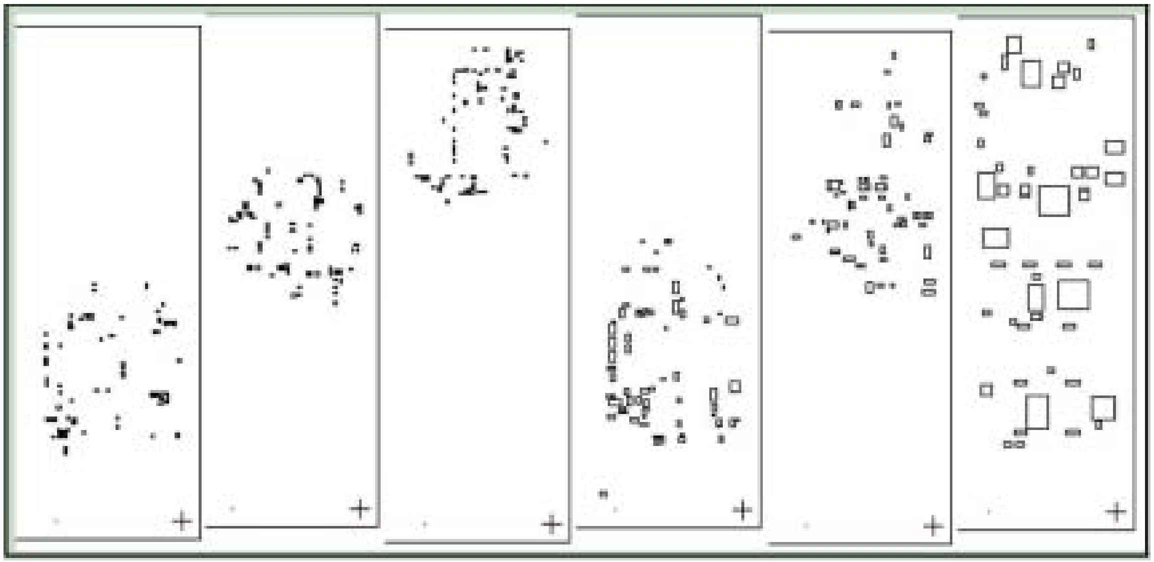

Figure 3. Manual parts distribution to six high-speed machines (A, B,…,F from left to right), only one block is illustrated from each machine program.

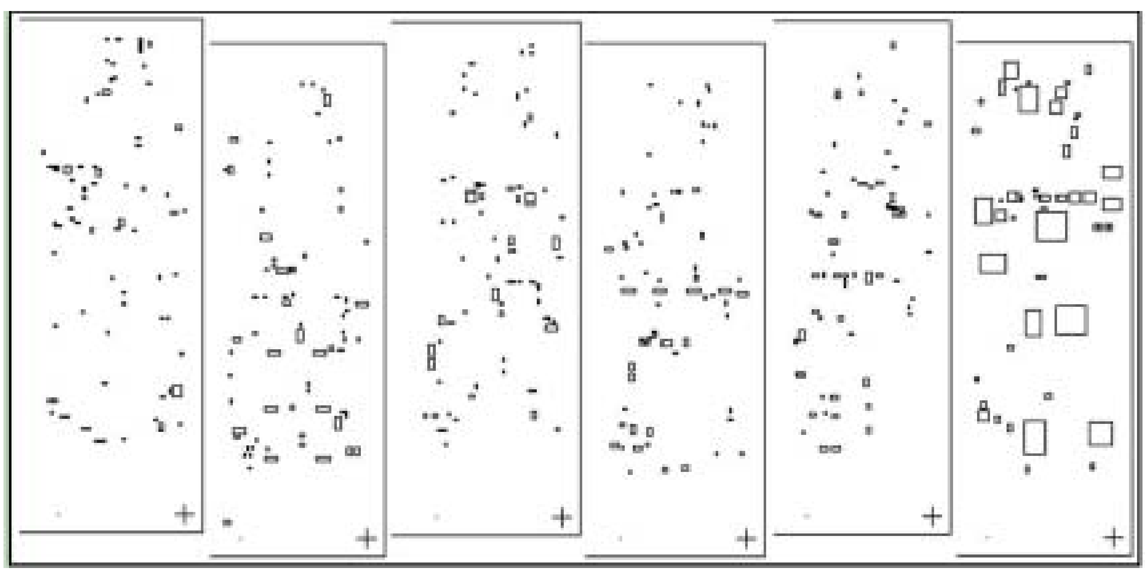

Figure 4. Parts distribution made by Trilogy.

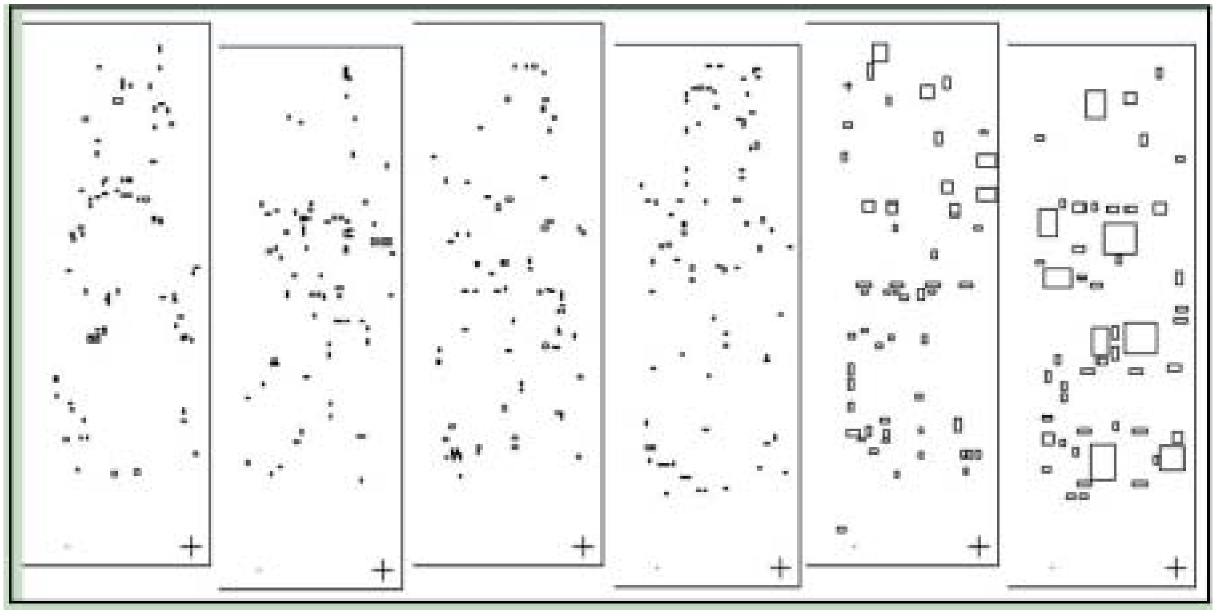

Figure 5. Parts distribution made by FujiCam.

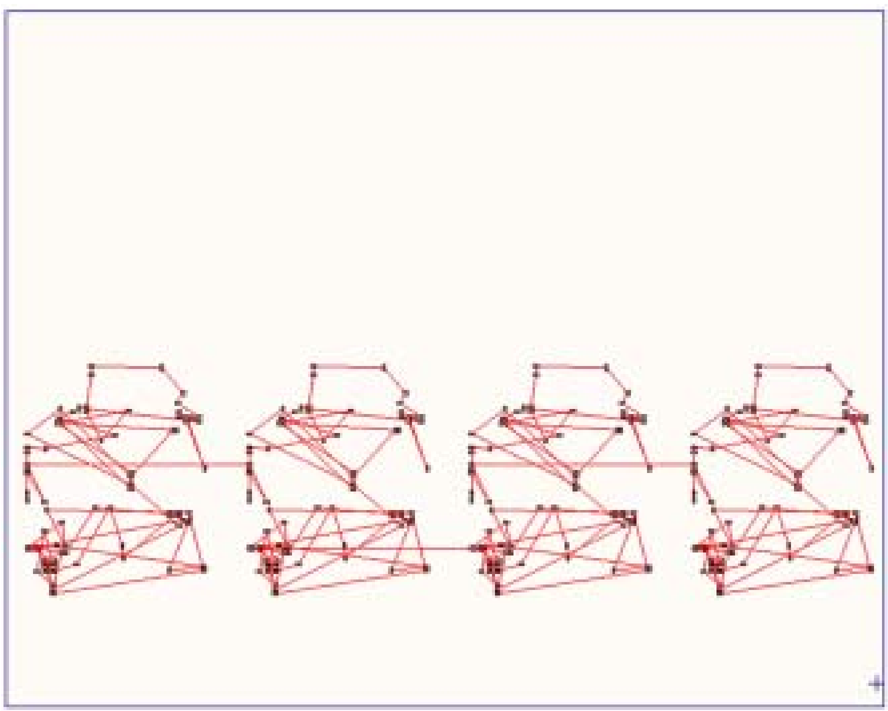

Figure 6. Placement path made manually for machine A.

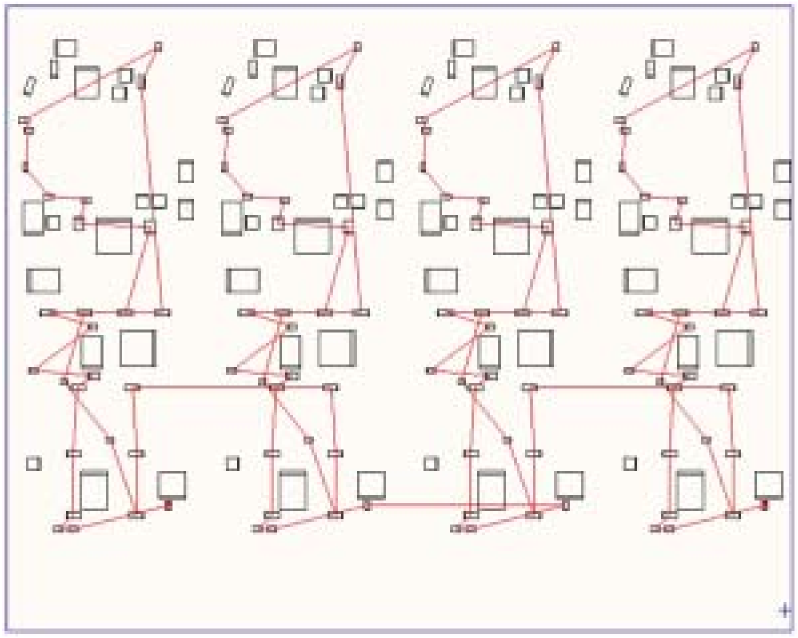

Figure 7. First part of placement path made manually for machine F (last high-speed).

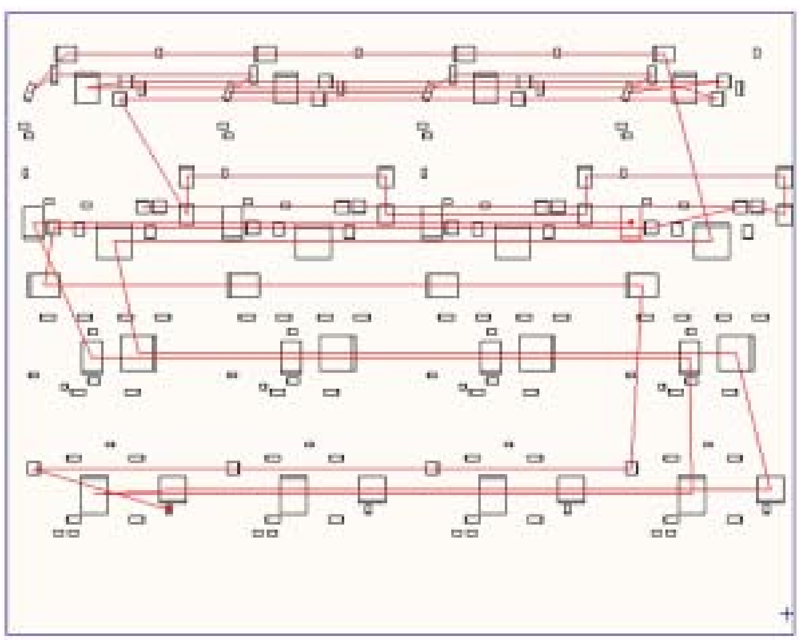

Figure 8. Second part of placement path made manually for machine F (last high-speed).

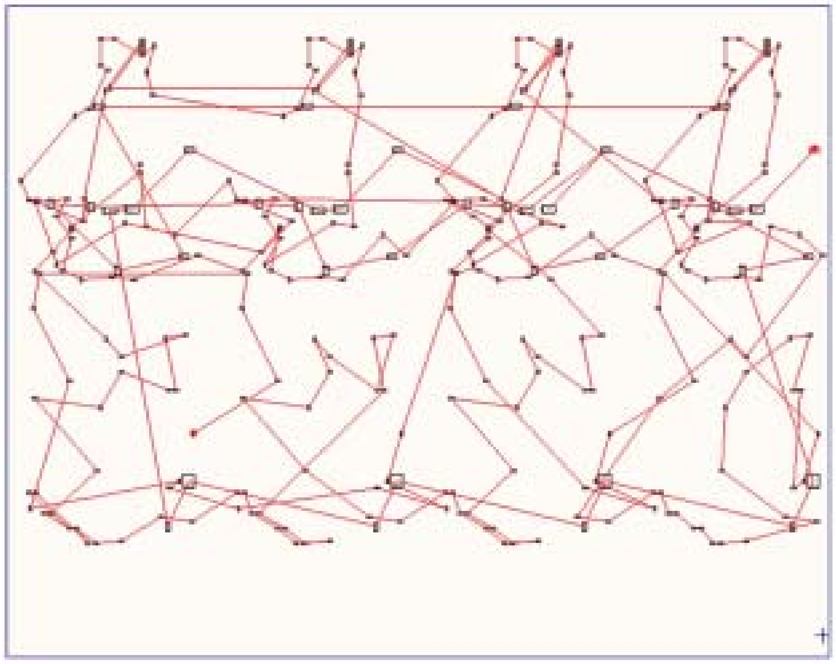

Figure 9. Final placement path generated by Trilogy for machine A.

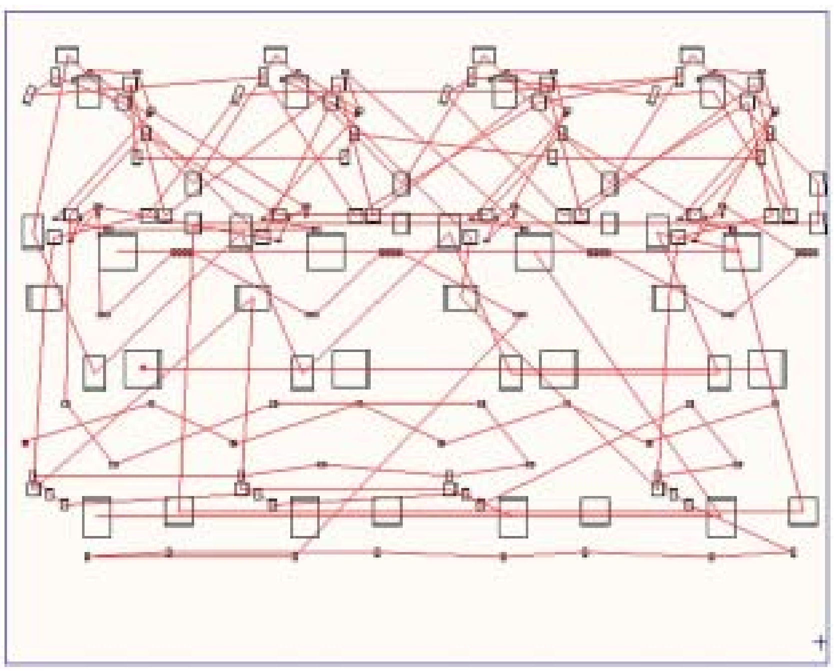

Figure 10. Final placement path generated by Trilogy for machine F (last high-speed).

sequence for the last high-speed machine is shown in Figure 10. All machines optimized by Trilogy were using the one-board function. FujiCam also generated all programs using the one-board function.

All programs were downloaded to placement machines and cycle times were measured multiple times using a stopwatch. The results are presented in Table 1. Trilogy has an interesting feature, allowing measured cycle times to be fed back to the system in order to rebalance the line. It can also split part numbers to multiple feeders in order to fine-tune the balancing of the line. The results can be seen from Table 1, in that line balancing and cycle time were much better with Trilogy after the second iteration. The fastest cycle time for the entire line can be achieved using manual line balancing and optimization. The bottleneck for that is given as a relative value of 100 in Table 1 and all the other results are compared (as a percentage) to this value. The bottleneck from Trilogy's programs is about 2.3% slower, but the line is very well balanced. Both FujiCam's cycle time and line balancing results are poorer; and the bottleneck has a cycle time of 9.2% slower than the best one.

4. Conclusions

The sequence generated automatically is sometimes not as good as the sequence designed by a human expert [5]. Our case study shows that this is indeed the case, especially with quite small multi-block PWBs in ultra-high volume lines. Most commercial software systems seem to take the board layout into deeper consideration too late, at a phase when part numbers have already been allocated to the machines (*i.e.* line balancing has been done), and final placement sequencing of an individual machine is about to start (*i.e*. single machine optimization).

In our case study manual line balancing and optimization produced the fastest cycle time for the whole line and the created placement programs were finally used for manufacturing of tens of thousands of products.

However manual line balancing methods could be improved. This is also seen in Table 1 as the potential for further balancing (average cycle time). We should find easier ways to distribute components using area based optimization, but at the same time keep the component count in each machine the same, because this seems to be the key factor to balancing, when, due to small placement area, most of the parts in each machine are placed with maximum speed. The time used for manual optimization in this case was about five working days for one SMT engineer.

Manual programming produced a lot of feeders compared to automatic program generation, because many part numbers exist in more than one machines. In order to avoid exceeding the maximum XY-table movement within full speed we ended up changing to another part number before having placed the last component from the current one. It should be better investigated when the XY-table movement is so large that it is wise to change to another part number instead of continuing with the current one.

Results from manual optimization are promising and encourage us to further develop and investigate the methods and at the same time create some kind of software tools to decrease the amount of manual work of this method.

References

[1] Rainisto, S., “Nokia repeää kahtia”, Talouselämä 2, 2003, pp. 20-24

[2] Gerits, P., et al., “Profitable Electronics Manufacturing Line Balancing”, Electronics Packaging & Production, September 2002, pp. 20-24

[3] Nevalainen, O., et al., ”Optimoinnilla tuottavuutta”, Prosessori 4, 1998, pp. 70-71

[4] Johnsson, M., “Operational and tactical level optimization in printed circuit board assembly”, Turku center for computer science, TUCS Dissertations, No 16, May 1999, pp. 26, 105

[5] Johnsson, M., et al., “Integrated Program Generation System for a PC-board Assembly Line”, Technical Report 52, Turku Centre for Computer Science, 1996, p.1