Effective Heuristic and Metaheuristic Approaches to Optimize Component Placement in Printed Circuit Board Assembly

Edmund K. Burke
University of Nottingham
School of Computer Science & IT
Jubilee Campus
Nottingham NG8 1BB, UK
ekb@cs.nott.ac.uk

Peter I. Cowling
University of Nottingham
School of Computer Science & IT
Jubilee Campus
Nottingham NG8 1BB, UK
pic@cs.nott.ac.uk

Ralf Keuthen
University of Nottingham
School of Computer Science & IT
Jubilee Campus
Nottingham NG8 1BB, UK
rxk@cs.nott.ac.uk

Abstract- This paper considers a scheduling problem arising for numerically controlled machines in the manufacture of printed circuit boards. In order to maximize the throughput rate of these machines the time taken for the pick and place sequence of components for each board has to be minimized. The determination of good pick and place sequences on a single placement machine gives rise to three related problems, namely the assignment of component types to feeder slots, the problem of tool changes and the sequence itself. In this paper we consider a sophisticated type of placement machinery equipped with multiple placement heads. We are going to introduce a suitable model for this complex scheduling problem and present tailored constructive and local search heuristics to determine good, locally optimal solutions. We embed the local search techniques proposed in the Variable Neighborhood Search metaheuristic framework to find still better locally optimal solutions and effectively search a larger part of the solution space. The heuristics have been implemented and applied to a panel of twelve test circuit boards to analyze their performance.

1 Introduction

Enormous growth in demand for consumer, commercial and industrial computer and other electronic devices has made Printed Circuit Board **(PCB)** assembly a fast growing market with an estimated worldwide economic value of nearly 350 billion US dollars[21]. As the variety of products in need of PCBs grows, the circuit board manufacturing industry relies more and more upon highly automated production processes to supply the increasing demand, including, in particular, numerically controlled machines for the placement of electronic components on the circuit board.

The recent introduction of Surface Mount Technology, which has nearly replaced pin-through-hole technology, for PCB assembly has enabled the mounting of an even larger number of electronic components on a single PCB. The increase of circuit board complexity has gone hand in hand with an improvement of the quality of placement machinery available in terms of speed and accuracy. However, assembly times and throughput rates of such a numerically controlled machine are highly dependent on the quality of the instructions given. This makes the machine's performance highly sensitive to its schedules and introduces high quality process planning as an important issue to increase throughput and cut down production costs.

In a printed circuit board production line, the circuit boards to be assembled usually travel on a conveyer belt through a line of component placement machines, each of them placing a number of electronic components on the circuit board. Assuming this type of production setup the production planning problem we are confronted with can be divided into three subproblems:

1. the allocation of component types to placement machines,
2. the assignment of component types to component feeder slots for each placement machine,
3. the component placement sequencing for each machine.

Many publications have addressed the problem of optimizing PCB assembly and various models and solution techniques have been proposed. One of the earliest, by Shellwell, Buzacott and Magazine [22], introduces a detailed model of a circuit board manufacturing facility. Ball and Magazine [2] concentrated on the sequencing problem, modelled as a rural postman problem [14], and the allocation of components to feeder slots. Bard et al. [3] presented a very detailed model for a single High Speed Chip shooter placement machine. A genetic algorithm-based approach to determine the component placement sequence was proposed by Kho and Ng [12]. A good literature review and description of various placement machinery types can be found in Moyer and Gupta [18] who also introduce a suitable model and heuristic for the sequencing problem of a High Speed Chip Shooter. Foulds and Hamacher [9] concentrated on the identification of optimal bin locations for component pick up. A combined model for sequencing and component assignment was suggested by Leipälä and Nevalainen [16] which was modelled as a quadratic assignment/Travelling Salesman Problem. An approach for all three problems mentioned above was presented by Crama et al. [7] considering the assembly of differ-

ent board types on a single assembly line of High Speed Chip Shooter placement machines.

For the remainder of this paper we are going to assume that the first of the problems listed above, the high level capacity planning problem of the allocation of component types to machines, has already been solved leaving the problems (2) and (3) for a single machine. Most publications considering this problems treat them as separated, often modelled as a quadratic assignment problem and a sequencing problem, and suggest a hierarchical approach to solve them. In contrast to this approach we are proposing to solve both problems together using suitable heuristics. We believe that this will produce better solutions overall since the assignment of component types to feeder slots has an significant influence on the quality of the placement sequence and vice versa. A further complicating factor is the highly sophisticated placement machine we are considering which is equipped with multiple placement heads, capable of picking several components from different feeder magazines simultaneously.

In this paper we are going to address subproblems (2) and (3) for a single multi-headed component placement machine. We will introduce a model for the machinery considered and propose heuristics to determine locally optimal solutions. We further demonstrate how this heuristics can be embedded into the Variable Neighborhood metaheuristic, as suggested by Hansen and Mladenović [10][11][17], to find better locally optimal solutions by searching a larger part of the solution space.

The paper is structured as follows. In section 2 we describe the placement machinery and production process. Section 3 introduces a suitable model. In section 4 we will describe heuristics to simultaneously determine placement sequences and assign components to feeder slots and show how they can be embedded efficiently into the Variable Neighborhood Search metaheuristic. The fifth section discusses the numerical results we have gained for the heuristics described in this paper applied to 12 test circuit boards of various sizes. The last section covers the conclusions we have drawn from this research and gives a brief overview of planned future research.

2 Process description

In this section we are going to describe the printed circuit board production process in more detail. Of special interest here is the placement machinery in use, as different types of machinery often vary significantly in their operation and therefore give rise to very different mathematical models. The typical component placement machine usually consists of four parts:

1. the work table where the circuit board is placed during the assembly process,
2. the feeder magazine(s) where components are stored for pick up
3. the placement head(s) to pick up and place components,
4. the automatic tool changer.

The type of component placement machine we are considering in this paper is a multi-headed component placement machine as illustrated in fig. 1. This type of machinery has also been considered in Crama et al. [8], Laarhoven and Zijm [13] and Lee et al [15]. Printed circuit boards are normally assembled on a production line. The printed circuit board is first moved onto the work table of the placement machine, usually by a conveyer belt, where it remains fixed at a predefined location during the placement process.

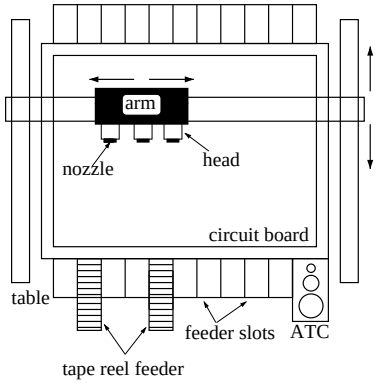


Figure 1: Multi-headed component placement machine

The components to be placed on the circuit board are stored, ready for pick up, in one or more feeder magazines located around the work table. A typical feeder magazine consists of a number of feeder slots often equipped with either tape reel or vibratory ski slope feeders to ensure the availability of components for placement on the board. For some types of placement machinery, such as the High Speed Chip Shooter [19][18] or the Dual Delivery placement machine [1], the feeder magazine must move to enable component pick up. For the type of machines considered here the feeder magazines are fixed.

Probably the most important part of the machine is the placement head which moves successively between feeder magazine and circuit board to pick up and place components. A special feature of the placement machine we are considering is that it is equipped with three placement heads mounted on a moving placement arm. Each of the three placement heads can use a variety of nozzles for vacuum placement or grippers and can rotate to orient the components for pick up or placement. The three placement heads enable the machine to move to the feeder magazine, pick up at most three components, either simultaneously or by moving along the feeder magazine, and then move back to the circuit board to place the components at their appropriate locations without having to return to the feeder magazine each time a component has been placed.

Due to differences in physical dimensions not all components can be placed by the same placement tool. This requires a change of the placement tools. Tool changes are typically carried out automatically by an automated tool gripper/nozzle changer, marked ATC in figure 1. We stress that tool changes are time consuming in comparison to the pick and place operations and are therefore usually only performed if absolutely necessary.

3 The model

In this section we are going to discuss a mathematical model for the production planning of multi-headed component placement machines. The approach considered here uses the model we proposed earlier in [5]. For a component placement machine equipped with multiple placement heads and fixed feeder magazine(s), as described in the previous section, we are confronted with the following three closely related scheduling problems.

(SP1) Component types have to be assigned to distinct slots in the feeder magazine(s).

(SP2) Suitable tools have to be assigned to each placement location, and

(SP3) a feasible component placement sequence has to be determined.

As already indicated in the introduction, most publications consider these three problems as separated from another and solve them independently. However, since the problems are strongly related to one another we suggest a different approach where we combine them into a multi level optimization problem. To introduce this approach we will make use of the following notation:

n	: number of placement locations,
m	: number of different component types,
f	: number of feeder slots available, where $f \geq m$,
o	: number of different placement tools available,
h	: number of placement heads available,
P	: Set of placement locations $\{p_1, p_2, \ldots, p_n\}$,
S	: Set of available feeder slots $\{s_1, s_2, \ldots, s_f\}$,
C	: Set of component types $\{c_1, c_2, \ldots, c_m\}$,
T	: Set of placement tools $\{t_1, t_2, \ldots, t_o\}$,
σ	: mapping $\sigma : C \to S$ assigning each component type to a nonempty set of feeder slots (no feeder slot is assigned more than one component type),
τ	: mapping $\tau : P \to T$ assigning a suitable tool for the component at each placement location,

The basic idea of our model for multi-headed placement machines is to consider the sequencing problem as a generalization of the Travelling Salesman Problem, a minimum cost hypertour problem, as illustrated in fig. 2.

Here, we define a multiple-pick up and the corresponding placement as an ordered sequence or hyperedge $H_i = (p_1^i, p_2^i, \ldots, p_{l_i}^i)$, where $p_j^i \in P$ and $l_i = |H_i| \leq h + 1$. A hyperedge $H_i = (p_1^i, p_2^i, \ldots, p_{l_i}^i)$ corresponds to a subsequence starting with empty placement heads at location p_1^i from where the placement arm moves to the feeder magazine to pick up the components needed to be placed at the locations $p_2^i, \ldots, p_{l_i}^i$. Then the placement arm moves back to the circuit board and places the components in the order $p_2^i, \ldots, p_{l_i}^i$ finishing with all heads empty at placement location $p_{l_i}^i$. The hypertour $\mathcal{S}= (H_1, H_2, \ldots, H_k)$ is then defined as an ordered sequence of overlapping hyperedges satisfying the following properties:

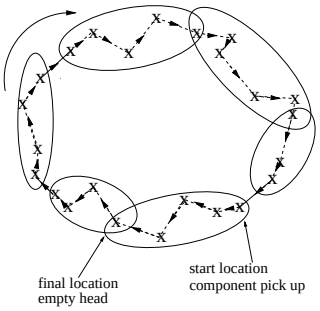


Figure 2: Hypertour Problem

$$2 \leq |H_i| \leq h+1,\ i = 1, \ldots, k, \tag{1}$$

$$p_{l_i}^i = p_1^{i+1},\ i = 1, \ldots, k-1, \tag{2}$$

$$p_{l_k}^k = p_1^1,$$

$$|H_i \sqcap H_{i+1}| = 1,\ i = 1, \ldots, k-1, \tag{3}$$

$$|H_k \sqcap H_1| = 1,$$

$$H_i \sqcap H_j = \emptyset,\ i, j = 1, \ldots, k,\ |i-j| > 1 \text{ and } \{i,j\} \neq \{1,k\}, \tag{4}$$

$$\bigsqcup_{j=1}^{k} H_j = \{p_1, p_2, \ldots, p_n\}, \tag{5}$$

where $H_i \sqcup H_j$ denotes the set of all the placements in either H_i or H_j and $H_i \sqcap H_j$ denotes the set of placements common to both. Keeping in mind that each hyperedge corresponds to a pick and place sequence, starting with empty heads at a placement location and finishing with empty heads at a placement location, after the pick and place operation has been performed, the inequalities (1) ensure each hyperedge consists of at least one but not more placements than the h placement heads available. Equalities (2) guarantee that the finishing location of a hyperedge is the starting location of its successor. (3) and (4) are necessary to avoid a placement location being visited more than once while (5) ensures that all placement locations to be visited .

To maximize the overall throughput rate of a single machine the aim is to minimize the placement arm travelling times of the pick and place sequences. In order to determine the quality of a sequence we need to introduce weights or distances between placements. Let $D^{\sigma,\tau}(H)$ denote the time

needed to perform the pick and place sequence defined by hyperedge H subject to a given assignment of components to feeder slots σ and tools to placement locations τ. Then we can state the problems **(SP1)**,**(SP2)** and **(SP3)** as a three level optimization problem which is given by:

$$\underset{\mathcal{S},\sigma,\tau}{\text{minimize}} \sum_{i=1}^{k} D^{\sigma,\tau}(H_i),$$

subject to constraints (1)-(5) introduced above. Note, that in the case of a single headed placement machine $|H_i| = 2, \forall i = 1, 2, \ldots, k$. This means that each hyperedge only consists of a single edge reducing the multi level optimization problem for fixed σ and τ to a Travelling Salesman Problem where the distances correspond to the travel time starting at a placement location to move to the feeder magazine, pick up a component, move back to the board and place it.

4 Heuristic approach

In this section we introduce some new heuristics to determine good locally optimal solutions for the complex optimization problem we described in the previous section. First we propose a constructive heuristic to determine a feasible solution for both the component assignment and the sequencing problem. Then we will introduce a local improvement heuristic to search the neighborhood of a feasible sequence for a local optimum. We are then going to show how this approach can then be embedded into the Variable Neighborhood Search metaheuristic to establish better sequences and component assignments.

To present the heuristics we introduce some assumptions which will be used throughout the remainder of this paper. In the following we are going to concentrate on the component assignment and sequencing problem and neglect the tool assignment problem, assuming that each placement head is equipped with a universal tool able to pick up and place components of all types. We further assume that the number of different component types m is equal to the number of available feeder slots f, i.e. that each component type is uniquely assigned to exactly one slot in the feeder magazine. These assumptions will simplify our exposition and can easily be relaxed so that our heuristic may be applied to practical problems where they do not hold.

4.1 Constructing an initial solution

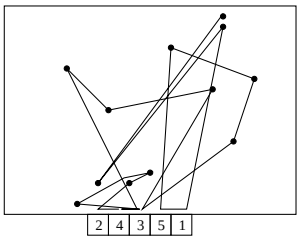
The constructive heuristic we introduce here is related to the well known Nearest Neighbor heuristic for the TSP. Our heuristic creates an initial solution to the problem, i.e. determine a feasible component assignment and placement sequence, of acceptable quality. This heuristic is detailed in figure 3. We start with all feeder slots unassigned to component types and all placement locations empty, with the placement arm at a randomly chosen feeder slot. At each step we construct a hyperedge assigning components to feeder slots as necessary. Each of the hyperedges constructed (except possibly the last one) will use all h placement heads, so that the corresponding hyperedges have cardinality $h + 1$. Note that the heuristic attempts simultaneously to minimize the distance between subsequent placement locations and between component feeder slots visited. This process is then repeated for as long as placement locations are available.

Set sequence $\mathcal{S} = \{\}$
No component types are assigned to feeder slots,
  i.e. $\sigma(s_i) = \{\}, \forall i = 1, 2, \ldots, f$
Select arbitrary feeder slot s_0,
While not all placement locations visited **Do**
  • set $H = \{\}$
  If $\sigma(s_0) = \{\}$
    ◦ select nearest placement location p to s_0 whose component type has not been assigned yet,
    ◦ Set $\sigma(s_0)$ to component type of p,
  Else
    ◦ select nearest non-visited placement location p to s_0 of component type $\sigma(s_0)$,
  End If
  • add p to H,
  While not all placement locations visited and head available **Do**
    ◦ select nearest feeder slot s^{next} to s_0 whose components have not all been placed yet,
    If $\sigma(s^{next}) = \{\}$
      ◇ select nearest placement location p^{next} to p whose component type has not been assigned yet,
      ◇ Set $\sigma(s^{next})$ to component type of p,
    Else
      ◇ select to p nearest non-visited placement location p^{next} of component type $\sigma(s^{next})$,
    End If
    ◦ add p^{next} to H,
    ◦ set $p = p^{next}$,
  End While
  • Add H as a hyperedge to sequence $\mathcal{S}$,
  • Select the nearest feeder slots to the last placement location of H whose componets have not all been placed yet as the new starting position s_0,
End While
Return Sequence $\mathcal{S}$.

Figure 3: The Nearest Neighbor Sequencing heuristic

An advantage of this algorithm is that it can determine a feasible sequence and also assign component types to feeder slots while constructing the sequence. It is, further, of low time complexity as the overall time complexity of the algorithm is $O(n^2)$, as usually $f \ll n$, i.e. the same complexity as the nearest neighbor heuristic for the Travelling Salesman Problem.

Figure 4 illustrates an example of this heuristic for a circuit board with 12 components to be placed of 5 different component types by a placement machine equipped with three placement heads. Though on first sight the sequence and component assignment found look rather good we have to keep in mind that this is a very small example. For boards with more insertions and feeder slots we can expect the constructive heuristic not to perform well enough, making further improvements, necessary and our results below will show this.



Nearest Neighbor Sequence

Figure 4: Nearest Neighbor Sequence for a 12 placement problem

4.2 Local Search

We will now propose a local search heuristic for the hypertour problem to improve initial solutions determined by the nearest neighbor construction heuristic described above. The basic idea of this algorithm is to successively apply simple sequence modifications for as long as improved sequences can be found. By applying modifications the neighborhood of a feasible solution is explored in order to reach a local optimum. The local search heuristic we are going to discuss here is called hyperopt and is based on a local improvement algorithm we suggested but did not experimentally test in [5]. This heuristic has, in a modified form already been applied to symmetric Travelling Salesman Problems[4][6] and performed very well in comparison to powerful TSP-solvers such as Lin-Kerninghan and $3-$opt [14][20].

Our local search heuristic will only try to optimize placement, fixing the assignment of components to feeder slots. In the next section we use Variable Neighborhood Search to recombine both problems to determine locally optimal sequences and component assignments of very high quality.

In the local search approach we suggest first two hyperedges are removed from the current sequence. For hyperedges of maximal length, where all placement heads are used, this leaves $(2h-2)$ placement locations unconnected to the sequence while the start and finishing locations of the chosen hyperedges remain attached to the sequence. The now unattached placement locations are then reconnected to the sequence by constructing new hyperedges consisting of optimal feasible subsequences. This process is illustrated in figure 5 for the case of a $3-$headed component placement machine. In figure 5(a) the initial hyperedges are shown, fig. 5(b) illustrates the unattached placement locations after the hyperedges have been removed while fig. 5(c) and (d) presents two possible outcomes of the reconnection process.

The neighborhood introduced by this heuristic is of size exponential in h and introduces many more possible moves than would be considered by, for example, simple swaps of placement locations. However, for the current generation of placement machines the number of placement heads is usually in the range 2 to 4, so the hyperedge reconstruction process is still computationally tractable.

We illustrate the performance of this algorithm in comparison to the nearest neighbor approach described above by applying hyperopt to the nearest neighbor sequence and component assignment of the small example problem shown in fig. 4. The resulting sequence was improved by about 35% in length if compared to the nearest neighbor approach and is illustrated in fig. 6. We give further comparisons in section 5.

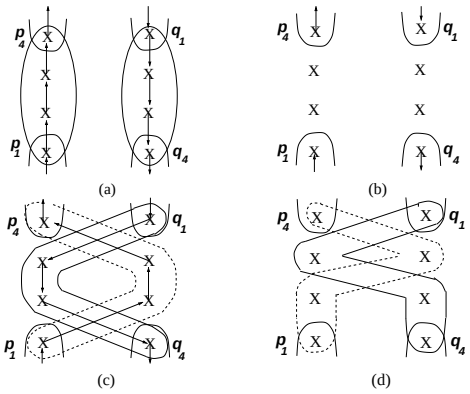
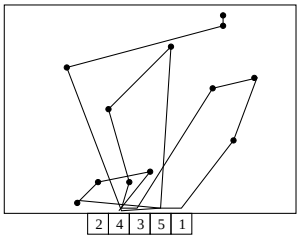


Figure 5: Schema of $3-$hyperopt moves



Hyperopt Sequence

Figure 6: Hyperopt Sequence for the 12 placement problem

A problem of the hyperopt local search technique proposed is that once a hypertour is chosen for each hyperedge the starting p_1^i and finishing placement locations $p_{l_i}^i$ stay fixed during the hyperopt local search, see p_1, p_4 and q_1, q_4 in figure 5. This may lead to a local optima of poor quality since the initial hypertour has been determined by the constructive heuristic described earlier.

In order to overcome this problem we suggest allowing swaps of these, otherwise fixed, placement locations with any other placement location of the hypertour. This means that the starting and finishing placement location of each hyperdege may be swapped with any other placement location in sequence $\mathcal{S}$ if this represents an improvement of the current sequence. For our implementation we chose to perform this process after each hyperopt iteration thus increasing the size of the neighborhood of our local search.

A further drawback concerning the multi level scheduling problem we stated earlier is that modifications are only applied to the placement sequence, neglecting the assignment of components to feeder slots. In order to simultaneously optimize the placement sequence and the assignment of component types to feeder slots, and improve solution quality we

embedded this local search heuristic in the Variable Neighborhood Search metaheuristic as described below.

4.3 Variable Neighborhood Search

The constructive heuristic and the local search technique described above show promise at determining good locally optimal solutions to the optimization problem we are confronted with. However, a drawback of the local search heuristic we proposed is that it only considers a part of the overall problem, namely the determination of good placement sequences. Further we may get trapped in a local optimum of poor quality. In order to overcome both problems we embed the local search technique into a Variable Neighborhood Search **(VNS)** heuristic as suggested by Hansen and Mladenović [10][11][17].

Initialization
- Use Nearest Neighbor construction heuristic to determine a feasible initial solution,
- Fix the maximal number of iterations as stopping criteria

Search:
Until stopping criteria met **Do**:
(1) Set $l = 2$
(2) **Do**
- Swap randomly l feeder slot assignments to create the neighboring solution S_n
- Apply local search to S_n to determine a local optimum S_n^*,

If local optimum S_n^* is of better quality than S^*,
- Set $S^* = S_n^*$ and **Goto (1)**,

Else
- Increase l by one and **Goto (2)**

End If
End Do
End Do Until

Figure 7: Variable Neighborhood Search for the PCB assembly problem

The basic idea of this metaheuristic approach is to systematically apply neighborhood changes to a locally optimal solution, either by changing the local search neighborhood altogether or by making limited changes using a larger neighborhood before reverting to the original local search method. A neighboring solution is then only accepted if it represents an improvement to the best solution found so far and is rejected otherwise. Hansen and Mladenović suggest a number of different approaches to change and explore neighborhoods.

In our implementation, the hyperopt approach is used as a local search technique while changes in the assignment of components to feeder slots are the larger neighborhood moves which "shake" the metaheuristic out of a local optimum and into a different part of the solution space. The algorithm is shown in figure 7.

A vital part of VNS is to "shake" the current solution strongly enough so that the local search heuristic will not fall back on an already visited local optimum but not so much that all information in the current solution is lost (and we have actually a random restart). The shakes we have chosen here can guarantee that will not happen, as the local search heuristic itself does not have any influence on the assignment of components to feeder slots.

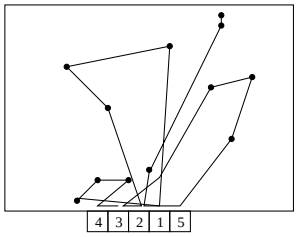


Figure 8: Sequence for the 12 placement problem using VNS

To illustrate the performance of this metaheuristic approach we also applied it to the example problem, which is shown in fig. 8. After just 20 "shakes" our metaheuristic was able to improve the initial solution by more than 52%. This provides an indication of the very large savings which are made possible by simultaneously considering the allocation of component types to feeder slots and the placement sequence of components.

In the next section we will to investigate the performance of the heuristic and metaheuristic approaches proposed here in more detail.

5 Experimental Results

The heuristics we described in the previous section have been implemented and applied to a test panel of twelve randomly generated circuit boards. The number of placements and component types ranged between 43 and 413 components to be placed and of up to 34 different component types. The placement locations and types are generated randomly. The number of placements n and component types m for each circuit board are shown in table 1. The placement machine considered here is equipped with three placement heads and one feeder magazine which is located next to the circuit board. We use the Euclidean distance metric between the placement and feeder locations.

The construction and local search heuristics have been implemented in exactly the way as they were described above. For the Variable Neighborhood approach we have fixed the number of iterations (or "shakes") to 100 and restricted the number of feeder assignment swaps within each "shake" to 6. In order to analyze the performance of our approaches we have also implemented two multi-start **(MS)** approaches. For both, the nearest neighbor heuristic is started from a random assignment of component types to feeders and then the local search described in section 4.2 applied to this initial solution. For the first multi-start approach, referred to as MS-1, this process is, as for VNS, iterated 100 times and the best solution is reported, while the second approach, MS-2, is given

the same CPU-time as needed by the VNS approach.

The performance of the four approaches is compared in table 1. This table shows the improvement in percent gained by local search, Variable Neighborhood Search and multi-start compared to the solution of the nearest neighbor sequencing and component assignment approach of section 4.1. The second and third column of this table show the dimensions of the artificial problems in number of placement locations n and different component types m.

We can see that local and Variable Neighborhood Search have been able to improve the initial sequence and assignment significantly. Especially the local search approach proved to be very successful as it was able to improve the solutions generated by our nearest neighbor construction heuristic by up to 35%. The Variable Neighborhood Search approach also proved to perform well as it enabled the heuristic to find even better local optima for all test problems. This approach gained results which are up to 40% better than the initial solutions from which the local and metaheuristic approaches were started. In comparison with both multi-start approaches, VNS was able to determine better locally optimal solutions for all but one test problem. Especially for the problems number $4, 5$ and 10 Variable Neighborhood Search proved to find significantly better solutions. For a mass product like printed circuit boards these gains could translate into many tens or hundreds of thousand of pounds in increased productivity.

boards			% improvement on constructive approach			
No.	n	m	Local Search	MS-1	MS-2	VNS
1	43	8	28.02	37.23	37.18	39.42
2	55	16	17.58	17.83	17.83	20.12
3	75	19	33.92	33.72	32.11	35.99
4	80	24	30.97	30.06	30.06	34.43
5	97	26	35.26	33.92	33.92	40.32
6	117	14	25.04	24.73	24.73	26.05
7	134	27	29.10	31.16	29.88	31.18
9	138	11	26.18	27.95	27.95	29.22
9	178	29	32.30	34.60	34.60	35.82
10	217	34	34.42	31.90	31.90	35.11
11	238	25	27.33	27.93	27.72	28.74
12	413	16	24.02	26.23	26.23	24.64
average:			28.67%	29.75%	29.50%	31.75%

Table 1: Experimental results: Board specifications and improvement in $\%$

The CPU-times of the Variable Neighborhood Search metaheuristic approach presented ranged from half a minute for the smallest up to several hours for the larger test problems. However, speed up and code optimization were, at this stage of our research, not a primary target. The Variable Neighborhood Search proposed in this paper performed considerably faster than multi-start. When both approaches are given the same CPU-time, as in the VNS and MS-2 columns of table 1, VNS was able to improve its advantage over multi-start even further. Since the "shakes" we have chosen enable VNS to reach an unexplored part of the solution space without losing too much information of the former locally optimal sequence, the local search proposed usually needed fewer iterations to reach a new local optimum than when started from a completely new initial solution as it did for multi-start.

6 Conclusions

In this paper we have presented a complex scheduling problem arising in the manufacture of printed circuit boards. The paper concentrates on the problem of assigning component types to feeder slots and the determination of locally optimal pick and place sequences for a multi-headed component placement machine. We used a model for a very sophisticated type of placement machinery, described earlier in [5], and heuristics were proposed to construct feasible solutions. These solutions were further improved by local search. In contrast to most publications concerning this subject we did not separate these two very closely related problems but treated them as a single problem and solved them simultaneously. We showed that large improvements are possible by combining the problems.

Our local search heuristic was embedded into the Variable Neighborhood Search metaheuristic as suggested by Hansen and Mladenović [10][11]. This enabled the heuristic to improve the placement sequences and component assignments still more by searching a larger part of the solution space. The computational results have been compared to a Multi-start approach also based on the local search heuristic presented in this paper. Here, Variable Neighborhood Search outperformed Multi-start for all but one of the test problems and was able to find locally optimal solutions of significantly better quality for some of them. For a product like printed circuit boards even a small reduction in assembly time of a few percent can lead to considerable cost savings.

Our future research will investigate ways to consider simultaneous tool changes in addition to the assignment of component types to feeders and the sequencing of placement locations. A further interesting area of future research may be to analyze the performance of different "shake" neighborhoods for the Variable Neighborhood Search approach presented and investigate "intelligent shakes" based on information obtained during the search process.

Bibliography

[1] J. Ahmadi, S. Grotzinger, and D. Johnson. Component allocation and partitioning for a dual delivery placement machine. *Operations Research*, 36(2):176–191, 1988.

[2] M. Ball and M. Magazine. Sequencing of insertion in printed circuit board assembly. *Operations Research*, 36(2):192–210, 1989.

[3] J.F. Bard, R.W. Clayton, and T.A. Feo. Machine setup and component placement in printed circuit board assembly. *International Journal of Flexible Manufacturing Systems*, 6(1):5–31, 1994.

[4] E.K. Burke, P. Cowling, and R. Keuthe. New local and variable neighborhood search heuristics for a sequencing problem in printed circuit board assembly applied to the travelling salesman problem. *University of Nottingham, Technical Report*, 1999.

[5] E.K. Burke, P. Cowling, and R. Keuthen. New models and heuristics for component placement in printed circuit board assembly. In *Proceedings of the 1999 IEEE International Conference on Information, Intelligence and Systems (ICIIS99)*, pages 133–140, 1999.

[6] E.K. Burke, P. Cowling, and R. Keuthen. Hyperopt: A new class of embedded local search heuristics applied to the travelling salesman problem. *University of Nottingham, Technical Report*, 2000.

[7] Y. Crama, O.E. Flippo, J. van de Klundert, and F.C.R. Spieksma. The assembly of printed circuit boards: A case with multiple machines and multiple board types. *European Journal of Operational Research*, 98:457–471, 1997.

[8] Y. Crama, A.W.J. Kolen, A.G. Oerlemans, and F.C.R. Spieksma. Throughput rate optmization in the automated assembly of printed circuit boards. *Annals of Operational Research*, 26:455–480, 1990.

[9] L.R. Foulds and H.M. Hamacher. Optimal bin location and sequencing in printed circuit board assembly. *European Journal of Operational Research*, 66(3):279–290, 1993.

[10] P. Hansen and N. Mladenović. An introduction to variable neighborhood search. In S. Voss et al., editor, *Proceeding of the 2nd International Conference on Metaheuristics-MIC97*, 1998.

[11] P. Hansen and N. Mladenović. Variable neighborhood search: Principles and applications. In *Invited papers at Euro XVI*, 1998.

[12] N.G. Khoo and T.K. Ng. A genetic algorithm-based planning system for PCB component placement. *International Journal of Production Economics*, 54:321–332, 1998.

[13] P.V. Laarhoven and W. Zijm. Production preparation and numerical control in PCB assembly. *Int. Journal of Flexible Manufacturing Systems*, 5:187–207, 1993.

[14] E.J. Lawler, J.K. Lenstra, A.H.G. Rinnoy Kan, and D.B. Shmoys. *The Travelling Salesman Problem: A guided tour of combinatorial optimization*. Wiley, 1985.

[15] S.H. Lee, T.H. Park, and B.H. Lee. A hierachical method to improve the productivity of a multi-head surface mounting machine. In *Proceedings of the IEEE Conference on Robotics and Automation*, 1999.

[16] T. Leipala and O. Nevalainen. Optimization of the movements of a component placement machine. *European Journal of Operational Research*, 38(2):167–177, 1989.

[17] N. Mladenović and P. Hansen. Variable neighborhood decomposition search. *Computers Oper. Res.*, 24:1097–1100, 1997.

[18] L.K. Moyer and S.M. Gupta. An efficient assembly sequencing heuristic for printed circuit board configurations. *Journal of Electronics Manufacturing*, 7(2):143–160, 1997.

[19] M.K. Ng. Heuristics approach to printed circuit board insertion problem. *Journal of the Operational Research Society*, 49(10):1051–1059, 1998.

[20] G. Reinelt. *The Travelling Salesman: Computational Solutions for TSP applications*. Springer, 1994.

[21] R. Sherman. *The worldwide PCB assembly market*. Electronic Trend Publications, San Jose, USA, 1998.

[22] S.F. Shevell, J.A. Buzacott, and M.J. Magazine. Simulation and analysis of a circuit board manufacturing facility. *Proceeding of the 1986 Winter Simulation Conference (December)*, Washington, D.C, 1986.