Annals of Operations Research 107, 321–338, 2001

Approximation Algorithms for Pick-and-Place Robots

ANAND SRIVASTAV asr@numerik.uni-kiel.de
Mathematisches Seminar, Universität zu Kiel, Ludewig-Meyn-Str. 4, 24098 Kiel, Germany

HARTMUT SCHROETER
Institut für Informatik, Universität Bonn, Römerstr. 2, 53113 Bonn, Germany

CHRISTOPH MICHEL
Commerzbank AG, ZHI Risc Management/Controlling, Neue Mainzer Str. 32-36, 60261 Frankfurt, Germany

Abstract. In this paper we study the problem of finding placement tours for pick-and-place robots, also known as the printed circuit board assembly problem with m positions on a board, n bins containing m components and n locations for the bins. In the standard model where the working time of the robot is proportional to the distances travelled, the general problem appears as a combination of the travelling salesman problem and the matching problem, and for $m = n$ we have an Euclidean, bipartite travelling salesman problem. We give a polynomial-time algorithm which achieves an approximation guarantee of $3 + \varepsilon$. An important special instance of the problem is the case of a fixed assignment of bins to bin-locations. This appears as a special case of a bipartite TSP satisfying the quadrangle inequality and given some fixed matching arcs. We obtain a 1.8 factor approximation with the stacker crane algorithm of Frederikson, Hecht and Kim. For the general bipartite case we also show a 2.0 factor approximation algorithm which is based on a new insertion technique for bipartite TSPs with quadrangle inequality. Implementations and experiments on "real-world" as well as random point configurations conclude this paper.

Keywords: pick-and-place robots, printed circuit board assembly problem, approximation algorithms, bipartite travelling salesman problem, matching, combinatorial optimization

1. Introduction

Pick-and-place robots are used for the automatic placement of electronic components on printed circuit boards. The optimization problem is to minimize the placement time of the robot. A mathematical model for the problem is the following.

1.1. *The printed circuit board assembly problem* PCBA(m, n)

We are given:

- m components partitioned into n sets $B_1, \ldots, B_n$, $m \geqslant n$, which we call *bins*. Let $\mathcal{B}$ be the set of bins. In real world, bins are geometrical objects like boxes or strips containing components. It will be convenient to color the components: let the components in bin B_i have color i.

- A finite set $\mathcal{L}$ of at least n points in the plane called *bin-locations.* For simplicity we assume $|\mathcal{L}| = n$.
- A set P of m points in the plane called *positions,* and a partition of P into n sets $P_1, \dots, P_n$ with cardinalities $|P_i| = |B_i|$ for all $i = 1, \dots, n$. The positions are also colored: the positions in P_i have color i. Let $N_i = |P_i|$ for all $i = 1, \dots, n$.

The i-colored positions correspond exactly to the locations on a printed circuit board on which the i-colored components must be placed. A *bin-assignment* A is a bijection $A : \mathcal{B} \to \mathcal{L}$. In real world, this is a physical placement of the bins on the bin-locations according to A. Given a bin-assignment A, a *placement tour* of the robot is defined as follows: the robot travels from a starting point to some non-empty bin B_i, picks an i-colored component, travels to an empty i-colored position, places the picked component on this position, travels to some non-empty bin B_j, and continues in this fashion until all components have been placed.

Given a bin-assignment A, let PCBA(m, n, A) denote the problem of finding a placement tour which takes minimum time $t(A)$. The PCBA(m, n) problem is the problem of finding the minimum $t(A)$ over all bin-assignments A. In other words, we have to determine *simultaneously* a bin-assignment and a placement tour such that the total working time is minimum.

Note that one difficulty of the PCBA(m, n) lies in the fact that a priori no good bin-assignment is known. Furthermore, the positions in the sets P_i can be arbitrarily scattered and do not necessarily form geometric clusters. The fact that a placement tour must be alternating between bins and positions seems to be the main difficulty in finding provably good approximation algorithms.

For the theoretical analysis we consider the standard model described in the literature, where the working time of the robot is assumed to be proportional to the distances travelled in L^1, L^2 or L^∞-norms. We will briefly call this the L^p-model.[1] Then, for certain choices of m and n, PCBA(m, n) reduces to the following combinatorial optimization problems.

- PCBA(n, n) is a special bipartite TSP. PCBA(m, n) can be considered as a generalization of such a bipartite TSP.
- Let A be a *fixed* bin-assignment. A induces N_i arcs leaving bin-location $A(B_i)$ and matching the N_i i-colored positions, for all $i = 1, \dots, n$. Let M be the set of all such arcs. For $m = n$, M is a matching. Every placement tour for PCBA(m, n, A) must traverse these fixed arcs. Thus PCBA(m, n, A) is related to the stacker crane problem (see [16]): a feasible solution for the stacker crane problem is a route visiting every node at least once and traversing the arcs of M in the prescribed direction while

[1] In this model we supress technological features such as robot arm acceleration, insertion/picking time, etc. Furthermore, we assume that all $n!$ bin-assignments are feasible. But even under this assumptions the L^p-model is realistic enough for some real-world assembly robots, and it helps to understand more complicated situations. Note that all above mentioned parameters can be modelled under appropriate cost functions and in the practical part of the paper we will include them.

a solution for PCBA(n, n, A) is a tour which visits every mode *exactly* once and traverses the arcs of M in the prescribed direction.

1.2. The size constrained bin-location problem

The optimization problem for the industrial robot (SIEMENS robot HS180) considered in this paper leads to an interesting new generalization of the PCBA(m, n) problem. While in the PCBA(m, n) problem all bin-assignments are feasible, robot HS180 allows only a certain subset of feasible assignments, for example restricted by the geometrical shape of bins and location regions. This is of complexity-theoretical significance. We will show in section 3 that the bin-assignment problem for size-constrained bins under a natural cost function is NP-hard. Let us denote by SC-PCBA(m, n) the printed circuit board assembly problem with size constrained bins. This generalization of the PCBA(m, n) problem will be discussed in sections 3 and 4.

1.3. Previous work

The work on the printed circuit board assembly problem has been focused on the development of practically efficient heuristics which iteratively solve the bin-assignment problem and the routing problem (see Drezner and Nof [9]). Ball and Magazine [3] used a fixed bin-assignment – based on the work of Drezner and Nof – and constructed a good tour in the Manhattan norm. Their problem appears as a Rural Postman problem, a generalization of the Chinese postman problem. Leipälä and Nevalainen [17] introduced a quadratic assignment problem as a model for a variant of the PCBA(m, n) problem. For $m = n$ their problem is the PCBA(n, n, A) problem in our notation. Foulds and Hamacher [11] and Francis et al. [12] considered the problem in which the bin-locations can be taken anywhere in the plane in contrast to the PCBA(m, n) problem with fixed, discrete bin-locations. To find the best location in the region, Foulds and Hamacher used a geometric algorithm. Based on the best bin-location they solved a TSP, and then a matching problem.

1.4. Results and organization of this paper

In section 2 we present:

- an $O(n^{O(1/\varepsilon)})$-time $3+\varepsilon$ factor approximation algorithm for the PCBA(m, n) problem, for all $\varepsilon > 0$;
- an $O(n^{O(1/\varepsilon)})$-time $2+\varepsilon$ factor approximation algorithm for the PCBA(n, n) problem, for all $\varepsilon > 0$;
- an $O(m^3)$-time 1.8 factor approximation algorithm for the PCBA(m, n, A) problem, when a fixed bin-assignment A is given;
- an $O(n^2)$-time 2.0 factor approximation algorithm for the computation of a tour with respect to a fixed directed matching in a complete bipartite graph $K_{n,n}$ where the

edge weights satisfy a quadrangle inequality. From this result a 2 factor $O(n^2)$-time algorithm for the PCBA(n, n, A) problem can be derived.

To show the first approximation result we give an algorithm which first computes a minimum weight bin-assignment, then finds a TSP tour among the bin-locations and finally merges the assignment edges with the tour.

The factor of $2 + \varepsilon$ in the second result comes from a better analysis of this algorithm in the special case $m = n$. In this case Anily and Hassin [1] and independently Michel [18] proved a factor of 2.5, and Motwani and Rao [6], and independently Frank et al. [14] showed the presently best worst-case approximation factor of 2. In a recent paper Baltz and Srivastav [4] proved that the worst-case performance of all of these algorithms cannot be better than 2.

The third result is obtained using the algorithm of Frederikson, Hecht and Kim [15] for the stacker crane problem in a straightforward way.

The fourth result is based on a new insertion technique for the construction of a directed travelling salesman tour containing a given directed matching in a bipartite graph in which the edge weights satisfy the quadrangle inequality. Note that with the $O(n^3)$ time algorithm of Frederikson, Hecht and Kim [15] an approximation factor of even 1.8 can be proved. Our insert algorithm gives the somewhat weaker 2-factor approximation, but needs only $O(n^2)$ time.

In section 3 we show the NP-completeness of the size constrained bin-location problem and discuss an interesting, polynomial-time solvable case.

Section 4 contains experimental data of two kinds.

Firstly, we have tested our approximation algorithm for the PCBA(n, n) problem on random instances as well as on instances arising in practice. Interestingly, in all of these experiments the approximation factor was less than 1.36 in contrast to the worst-case factor of $2 + \varepsilon$. For configurations arising from practical situations the approximation factor approached 1 as n increased.

Secondly, for the general problem SC-PCBA(m, n) and the application to robot HS180 we have implemented fast exchange heuristics and tested it against industrial solutions in which a TSP tour optimization with simulated annealing and a manual bin-assignment was used.[2] We were able to improve the optimizable (respectively total) assembly time of industrial solutions by 8% to 24% (respectively 4% to 12%). The interesting observation here was that the major improvement achieved by our heuristical algorithm is due to the bin-assignment and not to the TSP tour optimization.

2. Approximation algorithms

Let us start with a complexity-theoretical observation.

[2] Note that we cannot apply our approximation algorithm for PCBA(m, n) to the SC-PCBA(m, n) problem, because the size constrained bin-location problem is NP-hard, while the bin-assignment problem for PCBA(m, n) is solvable in polynomial time.

Proposition 2.1. The PCBA(n, n) problem is NP-hard.

Proof. Given an instance of the Euclidean TSP with n points, consider an instance of the PCBA(n, n) problem with $2n$ points where we simply double the given point set. This PCBA(n, n) problem is equivalent to the TSP problem. □

2.1. The general problem PCBA(m, n)

The first algorithm, which guarantees a $3+\varepsilon$ factor approximation for PCBA(m, n), has three steps: we find a minimum weight matching between bin-locations and positions, construct a TSP tour for the n bin-locations with Arora's PTAS [2], and finally we merge the matching with the tour.

Algorithm ROUTE.
Input: An instance of PCBA(m, n), $\varepsilon > 0$.
Output: A feasible tour T for PCBA(m, n).

1. We generate a complete bipartite graph $G = (V_1, V_2, E)$ with $|V_1| = |V_2| = n$: the vertices of V_1 represent the n bin-locations, the vertices of V_2 represent the n bins. Let d_{ik} be the distance of bin-location i to position k. For an edge $e = (i, j) \in V_1 \times V_2$ its weight is
$$c(e) := \sum_{k \in P_j} d_{ik}.$$
   - Find a minimum weight perfect matching $M = \{(i_k, j_k),\ k = 1, \dots, n\}$ in (G, c).
   - Place the bins on the bin-locations according to the matching M.

   Now we may identify bins and bin-locations and for both we will use the notation $B_1, \dots, B_n$.
2. Construct with Arora's PTAS a tour T_{board} for the points $\{B_1, \dots, B_n\}$ with $L(T_{\text{board}}) \leqslant (1+\varepsilon)Opt$ where Opt is the length of an optimal TSP tour for the points $\{B_1, \dots, B_n\}$. Let $T_{\text{board}} = (B_{j_1}, \dots, B_{j_n}, B_{j_{n+1}})$ be this tour where $B_{j_{n+1}} = B_{j_1}$.
   1. For all $l = 1, \dots, n$ do;
      Place all components contained in bin B_{j_l} on the N_{j_l} j_l-colored positions by travelling from B_{j_l} to the j_l-colored positions and back. Let p_{j_l} be the last visited j_l-colored position.
   2. From p_{j_l} travel to the bin-location $B_{j_{l+1}}$.
   3. Output is a tour T.

Figure 1 illustrates this algorithm.

Theorem 2.2. Let $\varepsilon > 0$ and let T_{opt} be an optimal tour for PCBA(m, n). ROUTE constructs in $\mathrm{O}(n^{\mathrm{O}(1/\varepsilon)})$ time a tour T for PCBA(m, n) with length $L(T)$ such that

(i) $L(T) \leqslant (3+\varepsilon)L(T_{\text{opt}})$ for $m > n$,

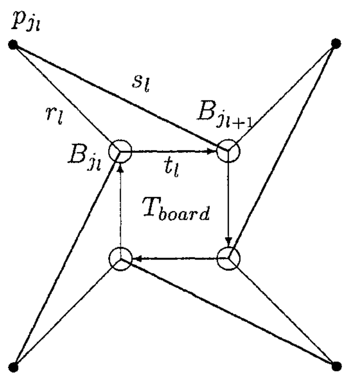

Figure 1. ROUTE for $m = n$.

(ii) $L(T) \leqslant (2+\varepsilon)L(T_{\text{opt}})$ for $m = n$.

Proof. By step 1 the bins have been placed on the bin-locations, thus in the following we may identify bin-locations and bins. The running time of Arora's algorithm is $\mathrm{O}(n^{\mathrm{O}(1/\varepsilon)})$, the minimum weight perfect matching algorithm takes $\mathrm{O}(n^3)$ time, the edge weights in step 1 are computed in $\mathrm{O}(nm)$ time, thus the total time is $\mathrm{O}(n^{\mathrm{O}(1/\varepsilon)})$ for sufficiently small $\varepsilon > 0$.

(i) By step 2,

$$L(T_{\text{board}}) \leqslant (1+\varepsilon)Opt \leqslant (1+\varepsilon)L(T_{\text{opt}}). \tag{1}$$

Let M be the set of matching edges $e_l = (i_l, j_l) \in V_1 \times V_2$ as constructed in step 1 of the algorithm, and set $L(M) = \sum_{l=1}^{n} c(e_l)$. Since M is a minimum weight perfect matching, and the optimal tour T_{opt} induces a perfect matching in (G, c),

$$L(M) \leqslant L(T_{\text{opt}}). \tag{2}$$

Remember, $T_{\text{board}} = (B_{j_1}, \ldots, B_{j_n}, B_{j_{n+1}})$ and $B_{j_{l+1}}$ is the bin visited after p_{j_l}. Let r_l be the edge (B_{j_l}, p_{j_l}), t_l the tour edge $(B_{j_l}, B_{j_{l+1}})$ and put $s_l = (p_{j_l}, B_{j_{l+1}})$.

Let T_l be the subtour of T between bin B_{j_l} and the positions from the set P_{j_l}. We have

$$\begin{aligned}
L(T) &= \sum_{l=1}^{n} L(T_l) + L(s_l) \\
&\overset{\substack{\text{(triangle}\\\text{inequality)}}}{\leqslant} \sum_{l=1}^{n} L(T_l) + L(r_l) + L(t_l) \\
&= \sum_{l=1}^{n} \big(L(T_l) + L(r_l)\big) + \sum_{l=1}^{n} L(t_l) \\
&= 2L(M) + L(T_{\text{board}}) \overset{(1),(2)}{\leqslant} (3+\varepsilon)L(T_{\text{opt}}).
\end{aligned}$$

(ii) Let $m = n$. The improved factor $2 + \varepsilon$ results from a better estimate of the matching M: observe that T_{opt} induces two perfect matchings M_1 and M_2 between the bin-locations and the positions. Hence

$$\min\bigl(L(M_1), L(M_2)\bigr) \leqslant \frac{1}{2}L(T_{\text{opt}}),$$

thus, since M is a minimum weight perfect matching between bin-locations and positions

$$L(M) \leqslant \frac{1}{2}L(T_{\text{opt}}).$$

The rest of the proof is as in (i). □

2.2. *Fixed bin-assignments*

An important special case of PCBA(m, n) in applications is the problem with a fixed, given bin-assignment A. For example, Ball and Magazine [3] analyzed such robots. Using the algorithm of Frederikson, Hecht and Kim [15] we have

Theorem 2.3. Let OPT be the length of an optimal solution for PCBA(m, n, A). Then a feasible tour T for PCBA(m, n, A) satisfying $L(T) \leqslant 1.8OPT$ can be constructed in O(m^3)-time.

Proof. Let M be the directed matching from the bin-locations to the positions induced by A as defined in the introduction.
Case 1: $m = n$. The stacker crane algorithm of Frederikson, Hecht and Kim [15] constructs in O(m^3) time a directed Eulerian tour R which traverses the arcs of M in the prescribed direction and satisfies $L(R) \leqslant 1.8OPT_{\text{stacker crane}}$. We can view R as a directed walk in which every edge of R appears exactly once. We use the following short cut technique in order to generate a feasible tour T for PCBA(m, m, A): while walking along R connect the head vertex of an arc in M with the tail vertex of the next arc from M in R. By the triangle inequality, $L(T) \leqslant L(R)$. Since the stacker crane problem is a relaxation of PCBA(m, m, A), $OPT_{\text{stacker crane}} \leqslant OPT_{\text{PCBA}(m,m,A)}$ and we are done.
Case 2: $m \geqslant n$. Consider for every bin-location i, N_i identical copies placed on the bin-location i. We connect the N_i i-colored positions with a perfect, directed matching M^* with the new N_i bin-locations. M^* defines a bin-assignment which we call A^*. In this function, we have generated an equivalent instance of the PCBA(m, m, A^*) problem. Now we apply case 1. □

2.3. *An insertion method*

Let us start with the following definition.

Definition 2.4. Let $G = (V, E)$ be a graph with edge weights c_{ij}. We say, in (G, c) the quadrangle inequality is valid, if for all 4-cycles $C_4 = (i, j, k, l, i),\ i, j, k, l \in V$,

$$c_{ij} \leqslant c_{jk} + c_{kl} + c_{li}.$$

We now give an $O(n^2)$-time insertion technique for the construction of a directed travelling salesman tour containing a given directed matching in a bipartite graph in which the edge weights satisfy the quadrangle inequality. The approximation guarantee of this algorithm is a factor of 2.0. As mentioned in the introduction, one can show also for this problem, following the pattern of the proof of theorem 2.3 and using the quadrangle inequality, an approximation factor of 1.8.

We have included the insertion approach to demonstrate how the insertion technique can be carried over to some graphs where the quadrangle inequality is valid.

Algorithm INSERT.
Input: A complete bipartite graph $G = (V_1, V_2, E),\ |V_1| = |V_2| = n$, a non-negative edge weight matrix $c = (c_{ij})_{\{i,j\}\in E}$ and a directed perfect matching $M = \{(a, b); a \in V_1, b \in V_2\}$, where (a, b) denotes the arc from a to b. Suppose that the edge weights are independent of the edge orientation, i.e., $c_{ij} = c_{ji}$ for all $\{i, j\} \in E$, and that (G, c) satisfies the quadrangle inequality.
Output: A directed travelling salesman tour T which traverses the arcs of M in the prescribed direction.

1. Minimum spanning tree construction:
   Consider the following complete graph H with edge weight matrix w: set $V(H) = M$. For $i, j \in V(H)$ with $i = (a, b),\ j = (a', b')$ define $w_{ij} = \min(c_{ab'}, c_{a'b})$, and let $w = (w_{ij})$. Construct in (H, w) a minimum weight spanning tree T_H.
2. Tour construction via insert:
   Initialization: $T_1 := \{(a_1, b_1), (b_1, a_1)\}$ for some $(a_1, b_1) \in M$.
   For $i = 2, \ldots, n$ do:
   Suppose the tour T_{i-1} containing the arcs $(a_1, b_1), \ldots, (a_{i-1}, b_{i-1}) \in M$ has been constructed. We extend T_{i-1} to T_i as follows.
   (a) Choose an arc $(a_i, b_i) \in M \setminus T_{i-1}$ which is connected to T_{i-1} by an edge of T_H (such an edge exists, because T_H is a spanning tree on $V(H) = M$).
   (b) Choose an arc $(u, v) \in T_{i-1} \setminus M$ such that
   $$c_{a_i b_i} + c_{u a_i} + c_{b_i v} - c_{uv}$$
   is minimum. (Note that by construction of T_{i-1}, we have $u \in V_2$ and $v \in V_1$.)
   (c) Replace (u, v) by the directed 3-path $P_3 = \{(u, a_i), (a_i, b_i), (b_i, v)\}$, and set $T_i := T_{i-1} \setminus \{(u, v)\} \cup P_3$.
3. Output $T = T_n$.

For a subset A of edges from G let $L(A) := \sum_{e\in A} c_e$ be the weight (or length) of A.

Theorem 2.5. INSERT constructs in $O(n^2)$-time a directed travelling salesman tour T in (G, c) which traverses the arcs of M in the prescribed direction such that

$$L(T) \leqslant 2L(T_{\text{opt}}).$$

Proof. Let T_{opt} be the optimal travelling salesman tour in (G, c) containing M. In the graph H, T_{opt} induces a spanning tree B, hence

$$L(T_H) \leqslant L(B). \tag{3}$$

By definition

$$L(B) + L(M) \leqslant L(T_{\text{opt}}). \tag{4}$$

Consider the construction of T_i from T_{i-1} in step 2 of INSERT. In T_H, (a_i, b_i) is connected with T_{i-1} by a unique edge $\{(a_i, b_i), (a, b)\}$ for some $(a, b) \in T_{i-1} \cap M$. To shorten notation denote by w_i its weight, so

$$w_i = \min(c_{a_i b}, c_{ab_i}).$$

Let (u, v) be as in step 2 of INSERT.

Claim. $L(T_i) - L(T_{i-1}) \leqslant 2(c_{a_i b_i} + w_i)$ for $i = 2, \ldots, n$.

Proof of the claim. Since $(u, v) \in T_{i-1} \setminus M$ minimizes $c_{a_i b_i} + c_{ua_i} + c_{b_i v} - c_{uv}$, we have for all arcs $(x, y) \in T_{i-1} \setminus M$ (where by construction of T_{i-1}, $x \in V_2$ and $y \in V_1$ holds)

$$\begin{aligned} L(T_i) - L(T_{i-1}) &= c_{a_i b_i} + c_{a_i u} + c_{b_i v} - c_{uv} \\ &\leqslant c_{a_i b_i} + c_{a_i x} + c_{b_i y} - c_{xy} \\ &= 2c_{a_i b_i} + c_{a_i x} + c_{b_i y} - c_{a_i b_i} - c_{xy} \\ &\leqslant 2\big(c_{a_i b_i} + \min(c_{a_i x}, c_{y b_i})\big). \end{aligned}$$

The last inequality follows from the quadrangle inequalities

$$\begin{aligned} &c_{a_i x} \leqslant c_{xy} + c_{b_i y} + c_{a_i b_i} \quad \text{and} \\ &c_{b_i y} \leqslant c_{a_i b_i} + c_{a_i x} + c_{xy} \end{aligned}$$

which imply $-c_{a_i b_i} - c_{xy} \leqslant c_{b_i y} - c_{a_i x}$ as well as $-c_{a_i b_i} - c_{xy} \leqslant c_{a_i x} - c_{b_i y}$. The claim is proved, if we can show $\min(c_{a_i x}, c_{y b_i}) \leqslant w_i$ for some $(x, y) \in T_{i-1} \setminus M$. In fact, in the tour T_{i-1} there are arcs $(b, g), (h, a) \in T_{i-1} \setminus M$ (see figure 2). If $w_i = c_{a_i b}$ then put $(x, y) := (b, g)$, otherwise if $w_i = c_{ab_i}$ then put $(x, y) := (h, a)$. In the first case

$$\min(c_{a_i x}, c_{y b_i}) = \min(c_{a_i b}, c_{g b_i}) \leqslant c_{a_i b} = w_i.$$

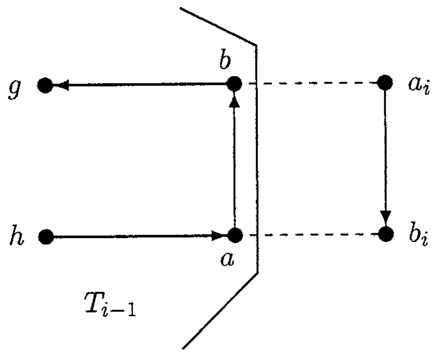

Figure 2.

The same argument applies for the case $w_i = c_{ab_i}$ and the claim is proved. Set $M_1 = M \setminus \{(a_1, b_1)\}$. We have already observed that for each inserted arc (a_i, b_i) the weight w_i is well-defined and corresponds to a unique edge of the tree T_H. Now

$$\begin{aligned} L(T_n) &= L(T_1) + \sum_{i=2}^{n} L(T_i) - L(T_{i-1}) \\ &\overset{\text{(claim)}}{\leqslant} 2c_{a_1b_1} + 2\sum_{i=2}^{n}(c_{a_ib_i} + w_i) \\ &\leqslant 2\big(L(M) + L(T_H)\big) \\ &\overset{(3)}{\leqslant} 2\big(L(M) + L(B)\big) \\ &\overset{(4)}{\leqslant} 2L(T_{\mathrm{opt}}). \end{aligned}$$

The running time is dominated by the time needed for the construction of the minimal spanning tree T_H which is $\mathrm{O}(n^2)$. □

With this theorem one can easily derive:

Corollary 2.6. For the PCBA(n, n, A) problem a tour T containing M can be constructed in $\mathrm{O}(n^2)$ time such that $L(T) \leqslant 2OPT$.

3. Size constrained bins

The bin-location problem with size constraints is defined as follows: we are given a rectangle S in the Euclidian plane having length s and width w, partitioned into subrectangles $R_1, \dots, R_s$ of unit length and width w. S is the location region for the bins, and the R_i are the bin-locations. We are given n bins of width w, but different lengths, say $l_1, \dots, l_n$ where $l_1 + \dots + l_n = l$ and $l \leqslant s$. A placement of a bin B_i on a location R_j is feasible, if the bin fits on S, i.e., $j - 1 + l_i \leqslant s$ and the required space, i.e., the rectangles

$R_j, \dots, R_{j-1+[l_i]}$, are not occupied by any other bin. It is important that we have a cost function Z for measuring bin-assignments. Let $C = (c_{ij}),\ 1 \leqslant i \leqslant n,\ 1 \leqslant j \leqslant s$, be a non-negative cost matrix, where c_{ij} is the cost for the assignment of bin B_i to bin-location R_j. Let x_{ij} be the $0/1$ variable which is 1 iff bin B_i is assigned to rectangle R_j and define

$$Z(x) = \sum_{i,j} c_{ij} x_{ij}.$$

For a bin-assignment A let us denote by $Z(A)$ the value of Z. Let $\vec{l}$ denote the vector $(l_1, \dots, l_n)$. The size constrained bin-location problem – SCBL$(n, \vec{l}, s, Z)$ – is to find a feasible bin-assignment (all bins must be assigned) with minimum value for Z. The SCBL$(n, \vec{l}, s, Z)$ problem is NP-hard. We give a proof suggested by Triesch [21] (see also [20]). Closely related to SCBL$(n, \vec{l}, s, Z)$ is the problem of scheduling jobs with equal length. For this problem Crama and Spieksma [8] gave an NP-hardness proof using a reduction from the 3-dimensional matching problem.

Theorem 3.1. SCBL$(n, \vec{l}, s, Z)$ is NP-hard.

Proof. We give a reduction from the 3-partition problem. An instance of 3-PARTITION consists of two integers m and b, a set $C = \{a_1, \dots, a_{3m}\} \subseteq \mathbb{Z}^+$ with $\frac{1}{4}b < a_i < \frac{1}{2}b$ for all i and $\sum_i a_i = mb$. The decision problem is: “Is there a partition of C, $C = C_1 \cup \dots \cup C_m$ such that $\sum_{a_j \in C_i} a_j = b$ for all i?”

We construct an equivalent instance of SCBL$(n, \vec{l}, s, Z)$ as follows. Set $n = 3m$, $l_i = a_i$ for $i = 1, \dots, 3m$, $l = mb$. Thus the bin-locations are indexed from left to right by $\{1, \dots, mb\}$. Let I_k for $k = 1, \dots, m$ be the index sets $I_k = \{(k-1)b + 1, \dots, kb\}$. We say, the assignment $i \to j$ is proper, if $j - 1 + l_i \in I_k$. Set

$$c_{ij} := \begin{cases} 0 & \text{if } i \to j \text{ is proper,} \\ 1 & \text{otherwise.} \end{cases}$$

Let x_{ij} be the $0/1$ variable which is 1 if bin B_i is assigned to bin-location R_j and zero otherwise.

Observe that the question: “Is there a feasible assignment for SCBL$(n, \vec{l}, s, Z)$ with $Z < 1$” is equivalent to the decision version of 3-PARTITION. □

If all bins have the same length, the problem reduces to an ordinary assignment problem. Note that the same is true for a special case of the scheduling problem considered by Crama and Spieksma [8].

Proposition 3.2. If $l_i = a$ for all bins B_i, then an optimal solution for SCBL$(n, \vec{l}, s = na, Z)$ can be found in $O(n^3)$ time.

Proof. Let V_1 denote the set of bins and let V_2 be the bin-locations which are the intervals $P_1, \dots, P_n$ of length a:

$$P_1 = [0, a], \quad P_2 = [a, 2a], \quad \dots, \quad P_n = [(n-1)a, na].$$

Let $\{i, P_j\} \in V_1 \times V_2$ be an edge of the complete bipartite graph G on (V_1, V_2) and define its weight as the cost of assigning bin B_i to the bin-location P_j. The optimal solution of SCBL$(n, \vec{l}, s = na, Z)$ is a minimum weight perfect matching in G. $\square$

From proposition 3.2 it is straightforward to derive an $O(k!n^3)$ time optimal algorithm for SCBL when k bins have length t_0 and $n-k$ bins have length t_1 (t_0, t_1 being non-negative integers).

4. Implementation and experimental results

4.1. Experimental results for algorithm ROUTE *for* PCBA(n, n)

In this section we introduce relaxations for the PCBA(n, n) problem which can be solved in polynomial time yielding lower bounds for the length of an optimal solution of PCBA(n, n). The performance of algorithm ROUTE for the PCBA(n, n) problem is measured with respect to these lower bounds.[3] Since Arora's PTAS has a quite high time complexity, in the actual implementation of ROUTE we computed the TSP tour among the bin-locations with the fast 3-opt heuristic. The tests were performed on three kinds of random point configurations in the Euclidean plane. Apart from uniformly distributed points (figure 3), we simulated typical PCBA settings in two non-uniform configurations (figures 4 and 5). In detail, the coordinates of points in these three situations were independently and randomly chosen from

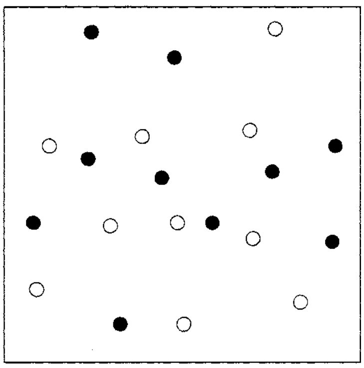

Figure 3. Configuration 1. On uniformly distributed points, ROUTE yields tours of significantly shorter length than guaranteed by the worst-case bound of $(2 + \varepsilon)OPT$. The approximation achieved is at most $1.36OPT$ in the worst-case and $1.12OPT$ in the average case, respectively.

[3] For the general problem ($m > n$) we do not have efficiently computable and good lower bounds for the optimum, thus a qualification of the experimental performance of ROUTE with respect to the optimal solution is difficult. For $m = n$ we give linear programming lower bounds [4].

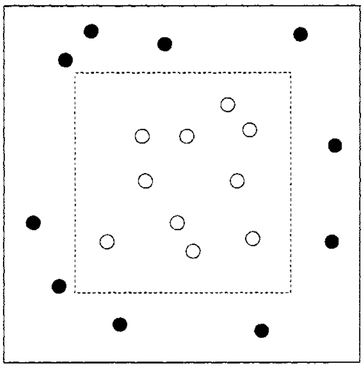

Figure 4. Configuration 2. In this more realistic situation, ROUTE produces tours of length at most $1.031 OPT$ on average. Note that the tour lengths seem to tend towards the optimum as the number of points increases: from 120 points on, the computed tours typically are only 1% off the optimum.

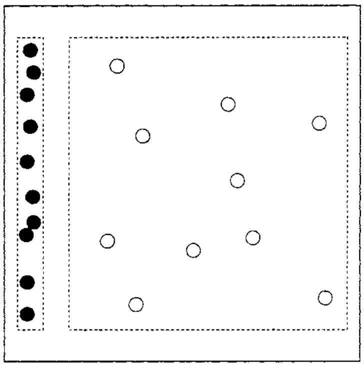

Figure 5. Configuration 3. This is a practically relevant configuration of points where our algorithm produces tours very close to the optimum. From 60 points on the optimum is exceeded by no more than 1%. Again, the tour lengths seem to tend to the optimum as the number of points increases. For 140 points the average deviation from the optimum is less than 0.1%.

- $\{1, \ldots, 500\}^2$ (both, red and blue[4]),
- $\{125, \ldots, 375\}^2$ (red) and $\{1, \ldots, 500\}^2 \setminus \{125, \ldots, 375\}^2$ (blue),
- $\{61, \ldots, 560\} \times \{1, \ldots, 500\}$ (red) and $\{1, \ldots, 10\} \times \{1, \ldots, 500\}$ (blue).

For each configuration we generated 5 times 100 instances on 20 up to 100 bicolored points. We restricted ourselves to 50 instances on 120 and 140 points, respectively. Maximum and average tour lengths are listed in table 1.

The tour lengths are given as factors of OPT^*, where OPT is the maximum of the length of an optimal *B*2-*tree* and an optimal 2-*matching,* both of which are lower bounds for the length OPT of an optimal tour for PCBA(n, n): a B2-tree is a degree constrained spanning tree, where each red vertex has at most two neighbors. Frank's [13] refined

[4] Red points are the bin-locations and blue points are the positions.

Table 1

Number of points	ROUTE-tour (maximum)	ROUTE-tour (average)
Configuration 1		
20	1.360	1.104
40	1.247	1.100
60	1.234	1.116
80	1.200	1.111
100	1.192	1.110
120	1.193	1.110
140	1.173	1.111
Configuration 2		
20	1.109	1.031
40	1.048	1.018
60	1.032	1.015
80	1.025	1.012
100	1.035	1.012
120	1.020	1.010
140	1.022	1.010
Configuration 3		
20	1.0595	1.0169
40	1.0580	1.0079
60	1.0075	1.0041
80	1.0057	1.0027
100	1.0035	1.0018
120	1.0024	1.0013
140	1.0019	1.0010

version of Edmonds' [10] weighted matroid intersection algorithm allows to compute a minimum weight B2-tree in time $O(n^7)$. Note that deleting an edge from a PCBA(n, n) tour yields a B2-tree justifying the use of an optimum B2-tree as a lower bound. Similarly, the length of a PCBA(n, n) tour is bounded from below by the minimum length of the edges of a graph, where each vertex has degree two. Such a 2-matching can be constructed in $O(n^3 \log n)$-time using a capacitated transportation algorithm [19].

4.2. *Experimental results for the* SC-PCBA(m, n) *problem and Robot HS180*

In this section we give a heuristic for the SC-PCBA(m, n) problem when the bin-assignment problem is an SCBL$(n, \vec{l}, s, Z)$ problem. Note that our approximation algorithms ROUTE for the PCBA(m, n) problem cannot be used for the SC-PCBA(m, n) problem, because the size constrained bin-location problem is NP-hard, whereas for the PCBA(m, n) problem it is solvable in polynomial time.

The algorithm for the SC-PCBA(m, n) problem iterates the following procedure: we start with a bin-assignment, say A, then compute a TSP cost matrix $C(A)$ depending on A for the m positions, find a placement tour T, and then compute a better bin-

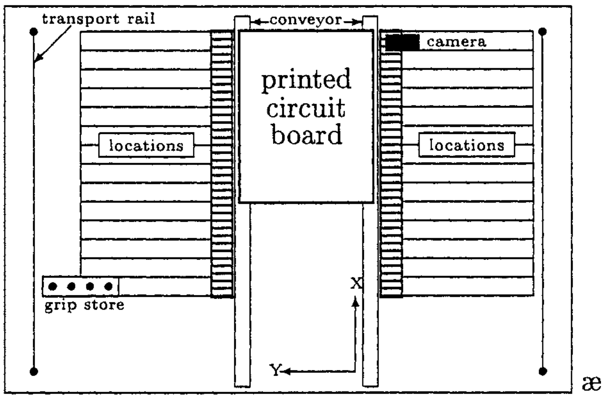

Figure 6. SIEMENS HS180.

assignment A' with respect to T. The last step is done as follows: for A' we have a new cost matrix $C(A')$. We say, A' is a better bin-assignment than A with respect to T, if the length of T computed with respect to $C(A')$ is smaller than the length of T computed with respect to $C(A)$.

We have implemented the algorithm for the robot HS180 and tested it on a set of 15 printed circuit boards against an industrial procedure in which TSP optimization with simulated annealing and a manual bin-assignment was used.

In the following we discuss our implementation which takes the technological features of HS180 into account. A schematic description of HS180 is shown in figure 6. Given a bin-assignment A, the standard heuristical method for $m = n$ is to solve an n-city TSP for the positions $P = \{p_1, \ldots, p_n\}$, where the component c_{ij} of the TSP cost matrix $C(A)$ is defined as the Tschebyscheff norm of $p_i - p_j$. This makes sense for robot arms moving synchronously in x and y direction. The robot arm of HS180 has different acceleration in x and y direction. We will define a weighted Tschebyscheff norm to cover this particular feature.

The TSP *cost matrix.* Let $p_i = (x_i, y_i)$, and $p_j = (x_j, y_j)$ be two positions. The Tschebyscheff norm of $p_i - p_j$ is

$$\|p_i - p_j\| = \max\{|x_i - x_j|, |y_i - y_j|\}.$$

This is an appropriate measure for the travelling time of the robot from p_i to p_j when the travelling time is proportional to the distances travelled. Since the x and y accelerations for the robot arm of HS180 are not the same, we define

$$\|p_i - p_j\|_t = \max\{t_x(|x_i - x_j|), t_y(|y_i - y_j|)\},$$

where $t_x(d)$ respectively $t_y(d)$ is the time needed to cover a distance d in x respectively y direction.

Now suppose that a bin-assignment A is given. Then the robot must travel from a position p_i to a position p_j via the bin B_j (which is placed on bin-location j according to A), and we can define travelling costs between the m positions.

We define the cost (or distance) c_{ij} between position p_i and p_j by

$$c_{ij} = \|p_i - B_j\|_t + \|p_j - B_j\|_t$$

(here we identify the point in the plane for the location j with B_j).

With the cost matrix $C(A) = (c_{ij})_{1 \leqslant i,j \leqslant m}$ we may consider an m-city TSP among the positions.

Bin-exchange heuristic. Suppose a travelling salesman tour T between the positions with respect to a bin-assignment A is given. The length of T is computed with respect to $C(A)$. We define the cost $\alpha_T(B)$ of a new bin-assignment B with respect to T as the length of T with respect to the cost matrix $C(B)$.

Our algorithm starts with a bin-assignment, computes a tour, improves the tour, computes a new and better bin-assignment with respect to the last tour and starts a new iteration. Both, tour and assignment optimization are done with exchange heuristics. For the bin-assignment problem (SCBL) we use the following exchange heuristic. Consider an instance of SCBL$(n, \vec{l}, s, Z)$ with $s = l$ (if $s > l$ introduce $s - l$ dummy bins of unit length).

Algorithm BINCHANGE(k).

1. For an initial bin-assignment A and a new bin-assignment B let $\Delta Z = Z(A) - Z(B)$ be the difference of the cost function. Define a super-bin of length i as a set of pairwise adjacent placed bins on S of total length i.
2. For $i = 1, \ldots, k$ do: exchange two super-bins of length i with maximum ΔZ. $\square$

It is clear that BINCHANGE(k) terminates after O(l^2) steps. We are ready to state the assembly algorithm.

Algorithm ASSEMBLY.
Input: An instance of the problem PCBA(m, n) where the bin-location problem is a size constrained bin-location problem SCBA$(n, \vec{l}, s, Z)$. Let k be an integer.
Output: A feasible placement tour.

1. Choose an initial bin-assignment and an initial tour.
2. Iterate until the assembly time cannot be improved:
   (a) improve the assignment with BINCHANGE(k),
   (b) improve the tour with the 3-opt exchange heuristic. $\square$

Table 2

Number of components	Running time (seconds)	Improvements (seconds)	Improvements (%)
221	374.14	15.67	4.16
132	165.40	7.85	4.68
46	79.04	3.72	4.70
163	161.94	7.69	4.74
71	88.76	4.39	4.94
89	103.76	5.3	5.10
38	52.64	2.80	5.31
110	128.06	7.1	5.54
149	141.76	7.94	5.60
69	66.54	3.74	5.62
55	74.94	4.25	5.67
49	68.34	3.92	5.73
248	240.20	20.19	8.40
19	39.10	4.88	12.48
20	20.92	2.67	12.76

Parameters and results. For the tested printed circuit boards and the HS180 the parameters were: $10 \leqslant n \leqslant 30$, $20 \leqslant m \leqslant 250$, bin lengths $1 \leqslant l_i \leqslant 6$, and $k = 6$. The average assembly time for a printed circuit board was 2 minutes. The optimizable assembly time was improved by 8% to 24% compared with the industrial assembly time. About 85% of the improvement has been achieved by a better bin-assignment, and only about 15% by 3-opt. Since the constant time in our examples is 50% of the total assembly time, the overall improvements are between 4% and 12% (see table 2), on average we have an overall gain of 6%. For TSP like problems this is considerably good. In view of the annual production of printed circuits boards the application of combinatorial optimization was successful.

5. Conclusion

While many research activities have focused on the design of practically efficient algorithms for the printed circuit board assembly problem, a satisfactory theory is still missing. We hope that our results can motivate research efforts also in this direction. There are two problems which we find interesting for future work:

1. Is there a polynomial time approximation scheme for PCBA(m, n)?
2. Is there a constant factor approximation algorithm for the size constrained bin-location problem?

Acknowledgments

We thank Bernhard Korte for introducing us to the PCBA problem in a joint project with the Klöckner & Moeller GmbH, Bonn. We thank the Klöckner & Moeller GmbH, Andras Frank, Alessandro Panconesi and Eberhard Triesch for many helpful discussions.

References

[1] S. Anily and R. Hassin, The swapping problem, Networks 22 (1992) 11–18.

[2] S. Arora, Polynomial time approximation schemes for Euclidian TSP and other geometric problems, in: *Proceedings of the 37th Annual IEEE Symposium on the Foundation of Computer Science* (1996) pp. 2–12.

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

[4] A. Baltz and A. Srivastav, Approximation algorithms for the Euclidean bipartite TSP, Technical report, Mathematisches Seminar, Universität zu Kiel (2001).

[5] A. Baltz, T. Schoer and A. Srivastav, On the bi-partite random travelling salesman problem and its assignment relaxation, in: *Lecture Notes in Computer Science* 2129 (Springer, 2001) pp. 192–201.

[6] P. Chalasani, R. Motwani and A. Rao, Approximation algorithms for robot grasp and delivery, in: *Proceedings of the 2nd Internat. Workshop on Algorithmic Foundation of Robotics* (1996).

[7] N. Christofides, Worst-case analysis of a new heuristic for the travelling salesman problem, Report 338, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, PA.

[8] Y. Crama and F.C.R. Spieksma, Scheduling jobs of equal length: Complexity, facets and computational results, Mathematical Programming 72 (1996) 207–227.

[9] Z. Drezner and S.Y. Nof, On optimizing bin packing and insertion plans for assembly robots, IIEE Transactions 16 (1984) 262–270.

[10] J. Edmonds, Matroid intersection, in: *Discrete Optimization I*, eds. P.L. Hammer, E.L. Johnson and B. Korte, Annals of Discrete Mathematics 4 (North-Holland, Amsterdam, 1979) pp. 39–49.

[11] L.R. Foulds and H.W. Hamacher, Optimal bin-location and sequenzing in printed circuit board assembly, Universität Kaiserslautern (June 1990).

[12] R.L. Francis, H.W. Hamacher, C.Y. Lee and S. Yeralan, Finding placement sequences and bin-locations for Cartesian robots, Research Report, Department of Industrial and Systems Engineering, University of Florida (1991).

[13] A. Frank, A weighted matroid intersection algorithm, Journal of Algorithms 2 (1981) 328–336.

[14] A. Frank, B. Korte, E. Triesch and J. Vygen, On the bipartite travelling salesman problem, Report No. 98866-OR, University of Bonn (1998).

[15] G.N. Frederickson, M.S. Hecht and C.E. Kim, Approximation algorithms for some routing problems, SIAM J. Comput. 7 (1978) 178–193.

[16] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys, *The Traveling Salesman Problem* (Wiley, 1992).

[17] T. Leipälä and O. Nevalainen, Optimization of the movements of a component placement machine, European Journal of Operational Research 38 (1989) 167–177.

[18] C. Michel, Optimierung des Fertigungsprozesses von Leiterplatinen durch approximative Algorithmen, Diplomarbeit, Forschungsinstitut für Diskrete Mathematik, Universität Bonn (1993).

[19] G.L. Nemhauser and L.A. Wolsey, *Integer and Combinatorial Optimization* (Wiley, New York, 1988).

[20] H. Schroeter, Optimierung der Depot-Anordnung bei der Bestückung von Leiterplatinen, Diplomarbeit, Forschungsinstitut für Diskrete Mathematik, Universität Bonn (1993).

[21] E. Triesch, Private communication (1993).