International Journal of Computer Integrated Manufacturing
Vol. 00, No. 00, Month 200x, 1–21

RESEARCH ARTICLE

Estimating the production time of a PCB assembly job without solving the optimized machine control

Frans Vainio[a], Tapio Pahikkala[a], Mika Johnsson[b], Olli S. Nevalainen[a], and Timo Knuutila[c]

[a] *Computer Science, Department of Information Technology and Turku Centre for Computer Science (TUCS), University of Turku, FIN-20014 Turku, Finland*;
[b] *Mentor Graphics Oy, Ruukinkatu 4, FIN-20540 Turku, Finland*;
[c] *Technology Research Center, Brahea Centre, University of Turku, FIN-20014 Turku, Finland*

(*Received 00 Month 200x; final version received 00 Month 200x*)

Production planning and control of the printed circuit board (PCB) assembly includes several decisions dealing with, for example, grouping of PCB jobs, allocation of PCB batches to machine lines, sequencing of batches and load balancing of lines. The production time of a PCB job for a given placement machine is a key factor in this context and it must be quickly and accurately estimated, possibly millions of times in a single planning task, to avoid erroneous decisions. The commonly used nominal tact time based estimators are very rough and the machine simulators too slow. Therefore, the purpose of this study is to give better machine specific estimators that avoid the construction the actual machine control program. Two new estimators are proposed for gantry machines, one based on the information given by the manufacturer about the operations of the placement head, and the other on the regularized least-squares regression method trained with a set of PCB placement jobs. In practical evaluation with 95 PCB jobs, the mean absolute percentage error of the first and second methods are 3.75% and 6.52%, respectively, while that of the tact time based approach is more than 17%. This indicates a great potential of the proposed methods as production time estimators.

Keywords: placement machine, PCB assembly, nozzle assignment, bottleneck, surface mount technology

*Corresponding author. Email: tapio.pahikkala@utu.fi

ISSN: 0951-192x print/ISSN 1362-3052 online

DOI: 10.1080/0951192YYxxxxxxxx
http://www.informaworld.com

1. Introduction

Assembly of electronic devices on printed circuit boards (PCB) is performed by the means of production lines consisting of several automated placement machines. To keep the agreements and realize the fixed schedules, production planning involves various decisions, such as scheduling the PCB jobs to meet the due dates of the products or balancing the work load of line machines. It is typical of applications like these that production times for different alternative partitionings of the assembly task to the line machines are repeatedly asked for, possibly even millions of times during a single planning task. Here, complete or even partial machine control optimization (i.e. defining feeder allocation, nozzles, component pick-ups, placement sequence, nozzle changes etc.) along with a machine simulator is often too slow and one needs an approximate method that quickly and accurately estimates the time for a large number of candidate partitionings. Design of this kind of production time estimators is studied in the present work.

The variety of different component placement machine types is very large in present days; vPlan by Mentor Graphics (2012) recognizes 770 machine types. To keep the study concrete, it is reasonable to concentrate the discussion to a particular class of machines. Gantry type placement machines having multi-nozzle collect-and-place heads are, due to their high production performance, flexibility, accuracy, and low cost, used increasingly in Printed Circuit Board (PCB) assembly. The machine types can further be classified into single and multiple head machines as well as into single, double, and multiple placement station machines. Placement station refers to the location where the component placements are performed. For example, ASM Siplace machines, on which this work focuses, has two revolver heads and a single placement station.

The task of controlling PCB assembly operations have been studied in numerous articles (see references in Laakso et al. [2002], Grunow et al. [2004], Michalos et al. [2010]). The involved problems are challenging due to their computational complexity and hierarchical nature. The PCB assembly design and control problems of the tactical level depend on the numbers of PCB types to be produced and on the machines in use (Johnsson 1999). On the lowest level (1 PCB type and 1 machine) there are detailed technical problems, which deal with the optimal control of an individual placement machine of a particular type (Grunow et al. 2004, Rong et al. 2009, Raduly-Baka et al. 2010), selecting the nozzle assortment (Raduly-Baka et al. 2008), optimization of the feeder assignment (Pyöttiälä et al. 2009), and optimization of the nozzle selection (Pyöttiälä et al. 2006). Higher levels of the problem hierarchy include strategic planning of the whole PCB assembly plant (Rogers and Warrington 2004), reconfiguration problem of the modular placement machines (Grunow et al. 2003, Toth et al. 2010, Rong et al. 2011), job grouping (Knuutila et al. 2004, Yilmaz et al. 2007), and production line balancing problem (Lapierre et al. 2000, Emet et al. 2010).

As an example, consider the line balancing problem. Publications on the topic include commonly the assumption of constant pick and placement times. As pointed out by many investigations (Laakso et al. 2002, Grunow et al. 2004, Yilmaz et al. 2007), the actual pick and placement times depend on the machine type (operational principle, number of placement heads and number of nozzles per head) and on the actual machine setup. Moreover, the pick and placement times depend strongly on the nozzle-to-head assignment, in particular on the number of copies of different nozzle types in the multi-nozzle head and on the component types (small or large shape). Different component types must be manipulated with suitable nozzle types, and hence the nozzle-to-head assignment should match the component shapes of the PCB job. A diverse nozzle setting in the head tends to require several pick-and-place cycles. Further, when the production program is object to dynamic changes and deadlines of the products are tight, a solution is often needed in very quickly, and hence fast production time estimators are needed.

Most of the above studies use simplified abstractions of the machines and boards which limits their direct applicability in practice. The production time is often calculated directly from the nominal tact time (components per hour) given by the manufacturer and possibly augmented with a slowdown coefficient, or with linear models based on the number of the components on the PCB. These approaches are fast to use, and they are often applied when the optimization method calculates assembly time estimates for a large set of alternatives as done, for example, in scheduling, line balancing, or job grouping. Their drawback is that the influence of the machine setting details like, compatibility between nozzles and components, component feeder arrangement, nozzle allocation and sequence on the assembly heads, component positions on the PCB, etc., are not taken account of. This may lead to suboptimal results and, in turn, to increased production costs, and in the end weakens the position of the manufacturer.

The purpose of the present study is to search for more exact estimates by taking advantage of the information given by the manufacturer about the operations of the placement head, and by considering the nozzle-to-head assignments and other features of the assembly task like frequencies of the different component types and nozzle types. Two assembly time estimators are proposed, namely a simulator for the operations of the placement heads and a greedy regularized least squares regression algorithm trained with a set of PCB placement jobs. While the second method runs in linear time the simulator consumes quadratic time, which is small in comparison to the time taken by the control code optimization along with a machine simulator.

To concretize the work, a large set of real production tasks for an ASM Siplace two-head machine were taken to estimate the assembly times (a similar machine has earlier been Siemens Siplace and the data are actually for that). Each production task is a list of components to be placed on the PCB. For each component are given component type, reference designator (placement position in x- and y-directions) and the nozzle type. The data includes also the real assembly time for each of the two heads and the estimated times can therefore be compared with the real times. Although the observations and conclusions are based on a particular gantry-type machine, they can be used more generally for other revolver head machines.

2. Production time estimators

The manufacturing time of a PCB job has been estimated in previous literature in the following four different ways. These ways are also summarized, with some representative references, in Table 1.

First, the most direct method is to solve the detailed control program of the job, set up the machine for production, perform the assembly of the components and observe the time from the clock of the machine. A bit cheaper method may be applicable if the machine allows the so-called “cold operation”-mode where the actual pick-ups and placements are performed without moving the real components. These two methods give actual production times (omitting the breaks for component reel changes). On the other hand, they are impractical because preparation and performing the tests is laborious and time consuming. Further, a machine or the whole production line may not be available for such tests. Due to these drawbacks, these kind of experimentations are of restricted use only. One such a case is practical fine tuning the production of a product. References on this topic are hard to find in scientific periodicals.

Second, the estimation method can trust on a machine simulator which takes a complete control program of the PCB job including all details which are needed by the real machine. The program is an output of the automated control program generator. The simulator then mimics the operations of the true machine (Tirpak et al. 2002, Kallio et al. 2012, Siplace 2012, Mentor Graphics 2012). This kind of simulators give very accurate results, and thus they are valuable when evaluating the design of the production process. As a drawback, their construction is hard, machine parameters may need reverse engineering and most of all, one has to trust on the software for generating the true control program of PCB assembly. For the last property the simulator is not suited for use as a principal cost function of an optimizer which evaluates a large set of candidate solutions.

Third, it is possible to approximate the cost of a PCB job on a more coarse level without knowing the real machine control program (Altinkemer et al. 2000, Pyöttiälä et al. 2013). This kind of estimator suits well to be used in solution algorithms for sub problems of the assembly control program. Estimators vary greatly in their complexity according to their use situations. A common characteristic of them is that they get the number of components, some parameters of the different machine operation speeds and machine details as their input and use simplified assumptions on the operations principle of the machine and on the control program of the PCB job. Even here the formulas for the production time may become rather involved. On the other hand, this approach has the benefit that one can use it on very rough level of abstraction. For example, when considering the feeder assignment it may be possible to model the movement lengths on the PCB area by averages and concentrate on the movements above the feeder bank and on the revolver head rotations (Pyöttiälä et al. 2013). An extreme abstraction is to let the production time to be simply the ratio of the number of component placements divided by the observed average number of component placements per hour. Though commonly used, this estimate may lead to weak results in particular when nozzle changes cannot be avoided or there are "slow" and "fast" component placements done by the same machine.

Fourth, an estimator can be constructed by the means of regression modelling methods. In this approach, a set of typical PCB jobs are collected and used to train the estimator. The production times of the jobs are observed from the machine clock along with a number of the parameters on the PCB. If the control programs of the jobs have been optimized, one can construct in this way a regression model which models the optimal production times of the given set of jobs for the given machine with its current machine setting, see Laakso et al. (2002), Wu and Ji (2010), Yilmaz et al. (2009) and Vainio et al. (2010), for linear and non-linear neural network regressions. These estimators are useful when a large number of optimized solutions are evaluated for line balancing or production scheduling purposes.

Linear regression models are constructed in Laakso et al. (2002) for a turret machine with a moving feeder unit and PCB holding table and single stationary revolver head. The total number of component placements and the number of different component types F was used as the parameters. The effect of different nozzle types was omitted. Coefficient for determination $R2$ ranged from 82.4 to 92.2 depending on the number of parameters (in the above order).

The above machine type was also used by Wu and Ji (2010) who reached $R2 = 0.999$ with a non-linear model for predicting the time t of the form

$$t = a_0 + a_1 N + a_2 \sqrt{NAF}, \tag{1}$$

where a_0, a_1 and a_2 are coefficients, N is the number of components placed in the PCB job, and A is the area of the smallest rectangle surrounding the components on the PCB.

The average placement time per component was estimated in Yilmaz et al. (2009) for a collect-and-place machine with a revolver head in a single gantry. The effect of different nozzles is not visible in this estimate.

A multi-layer perceptron was trained for a two-gantry machine with revolving heads by Vainio et al. (2010). The input parameters of the neural network were N, F, d (the number of different component shapes), A, $\max\{L\}$ (maximal side length of the PCB, which for a rectangular PCB is $L = \sqrt{A}$), f (component size variance). The Bayesian regularization of the network parameters was applied to avoid overfitting of the model.

The intention in the present study is to construct a practical machine operation time estimator, which is faster to use in applications like the PCB assembly line balancing or scheduling than the approaches requiring to solve the fully optimized control of the machine separately for each PCB job. To reach this goal, two approaches are proposed. Approximation of the head movements by a nearest neighbor rule, and use of modern least-squares regressors. Real production data of a two-gantry machine is used for testing the estimation performance of both approaches and for training the regression method.

The nearest neighbourhood rule models the result of a machine optimizer in a simple way by determining heuristically a component placement route and using the nominal machine parameters in the time calculations. While the aim of the research in the machine control optimization is to develop optimal feeder and nozzle assignments and component placement ordering, the aim in this study is to only estimate the production time. The nearest neighbour method operates thus between the nominal placement rate approach (Bentzen 2000) and the detailed modeling (Seth et al. 2009).

To train a regressor, a modification of the regularized least-squares technique (Hoerl and Kennard 1970) is used. The technique has the ability of choosing the most important parameters to the model in an automated way. This property makes the model construction much easier in comparison to previous approaches.

3. Characteristics of rotary head assembly machines

A rotary head type assembly machine has one or several rotary collect-and-place assembly heads, called also revolver heads, (cf. ASM Siplace and Fuji NXT, AIM and XP series). In order to keep the discussion coherent, it focuses on ASM machines that use two placement heads in turns to place components on the same PCB. The terminology of machine classification is still evolving and therefore these kind of machines have been placed to the class of multi-head placement machines (of type one), collect-and-place machines (having a revolver head for nozzles), or collect-and-place machines (having an array of nozzles), see Ayob and Kendall (2008) for more in depth discussion. Both heads work simultaneously; as one head collects (maximally 12) components from the stationary feeder bank, the other head places its collected components (maximally 12) to their correct positions on the stationary PCB. Because the two heads are working simultaneously but avoiding collisions, the "unproductive" time of collecting components is minimized or practically eliminated. One of the two heads is supposed to be almost always "productive" placing components on the PCB.

An ASM machine can also have four placement heads. These machines, in fact, consist of two different placement stations and each of them has two 12-nozzle collect-and-place heads. These are called multi-station machines.

3.1 *Collect-and-place rotary heads*

The high *placement performance* (i.e. speed and accuracy) of the gantry-type placement machines is achieved by using collect-and-place heads. A collect-and-place head operates with multiple nozzles to minimize the number of unproductive movements from the feeder to the PCB and vice versa. For example ASM machines are equipped with compact rotary (revolver) heads. The exact position and orientation of the collected components on nozzles is measured in the fly using an integrated component camera mounted above the head. The head can be therefore rotated only in one direction.

It is possible that there are several collect-and-place revolver head types available for a placement machine (e.g. the head of ASM Siplace can have 6, 12 or 20 spindles to carry nozzles). Each head type can handle a spectrum of feasible component shapes (package forms and sizes). The feasibility can be expressed by a function:

$$F(h,s) = \begin{cases} 1 \text{ if head } h \text{ is capable of manipulating component shape } s \\ 0 \text{ otherwise} \end{cases} \tag{2}$$

Technical specifications of the placement heads set restrictions on the component shapes. Therefore the component types that are assigned to the head must fulfill the restrictions given by F. Further, there are restrictions dealing with the order of the revolver heads on the assembly line; smaller (lower) components have to be placed first. Otherwise, there will be collisions between the head and already placed higher components.

Because the revolver head can rotate only in one (forward) direction, the collect-and-place operations are FIFO-like where the sequence of the component collections defines also the sequence of the component placements. Therefore, collection and placement steps can not be optimized separately.

3.2 *Component shapes and nozzle types*

The electronics components are packaged in numerous different shapes. As mentioned before, a certain nozzle type is required to pick and place components of a given shape. The things are however more complicated; one nozzle type may handle several component shapes and one component shape may be handled using several nozzle types. Thus, in addition to the so-called primary nozzle type it may be possible to trust on a secondary type. It is question of the reliability of the nozzle-component fastening during the head movements due to the high acceleration values and centrifugal forces.

Allocating different nozzle types to a placement head stands for selecting a set of different nozzle types and the number of copies of each type. The allocation greatly affects the route lengths of the head. In order to demonstrate this, consider a single PCB type to be processed by a single machine; nozzles are not changed during the assembly; and the minimal number of pick-and-place cycles is constructed. Let us first illustrate the effect of different nozzle types on the placement performance. Suppose that a head is equipped with 5 different nozzle types and the nozzle set for the 12-spindle head is the following (ASM standard nozzle types):

spindles	nozzle type
1	911
2-6	905
7-10	923
11	915
12	917

The head has to collect and place 43 components on a PCB. The dimensions of the PCB are 119 mm x 175 mm. To do the placements, the head has to perform 4 collect and place cycles. Figure 1 illustrates the real placement cycles (the data is an excerpt from an optimized production control program).

As one can see, the placement routes are very confusing and seemingly unfavourable. Because the placement head is equipped with 5 different nozzle types and the head can rotate in one direction only, route planning has to be made separately for each nozzle type. The resulting route is therefore the sequence of five partial routes in the same placement order as the nozzle types are in the placement head. In this example there are three nozzle types 911, 915 and 917, having only a single copy.

On the other hand, if all the nozzles were of the same (universal) type or if the head could rotate in both directions (without significant delay) one could get shorter routes on the same PCB area; see Figure 2 for the same component placements as in Figure 1. Euclidean metrics are used for calculating the distances in this example case. The total length of the sub-routes in Figure 1 is 1212.8 mm whereas in Figure 2 it is only 439.1 mm.

This simple example raises the question whether one may cast an efficient estimator for a gantry-type assembly machine in the presence of multiple nozzle types in the revolver heads and the heads rotating in one direction only. In particular, one wants to avoid the time consuming generation of the full control program and still reach a reasonable accuracy.

4. Nearest neighbour of certain type (NNCT) algorithm

Let us consider a PCB assembly job J which calls for placements of m components $c_{i=1}^{m}$, each of which can be written as a tuple $c_i = (x_i, y_i, u_i, v_i)$, where x_i and y_i are the coordinates (called reference designator) of the component on the PCB, u_i is the component type, and v_i the primary nozzle type demanded by the component. The task in assembly control optimization is to find assignment of feeders and nozzles, and a permutation $W = (i_1, i_2, \ldots, i_m)$ of the job J such that the total manufacturing time is minimal for this particular order of placements. When using a multi-nozzle placement machine the processing runs in pick-and-place cycles, called task blocks. W can therefore be expressed as a sequence of the task blocks $W = (W_1, W_2, \ldots, W_k)$ where the task blocks form a partitioning of the permutation W. Here the length of a task block is restricted by the nozzle capacity of the head.

Recall that the placement station considered in this paper has two gantries working on the same PCB area by turns so that one is collecting components from the feeder bank while the other is placing them. Therefore, in order to approximate the total assembly time, one can mostly ignore the component collection time, because it is usually no larger than the placement time. That is, the collecting head has enough time to complete the collection phase and then to move to the waiting position near PCB before the placing head has completed its placements. Accordingly, the assembly times are estimated based only on the component placements on PCBs. A second and even more important reason is that the production data does not give us any information about the feeder arrangements.

Due to the lack of this information, one also has to generate the component placement orders in a simplified way. The nozzle arrangements are here based only on the nozzle type frequencies calculated from the job data. The nozzles are supposed to be in the head in groups of the same type and in decreasing order of group sizes. Thus, the most frequent nozzle type is filled with components first, then the next frequent nozzle type etc. In the following, it is supposed that the component placements have been allocated to the both of the heads. This is because the test data was obtained from real optimized production. Allocation of the placements to the heads should in general be done but this is bypassed here.

The above arrangement leads us to use a modification of the nearest neighbour search (NNS) heuristics in the route construction. The basic NNS heuristics can be used to find a suboptimal solution to the vehicle routing problem. In this algorithm one selects a starting point for the placement route and then computes the distances from the starting point to each other remaining point and proceeds to the nearest one. The same is iterated by starting from the found new point and maintaining the route and the set of unvisited points. A modified version of the NNS algorithm (called nearest neighbour of certain type, NNCT) is used to plan placement routes of a revolver type placement head. Now, the total route consists of a set of separate but interconnected sub-routes. The head type defines the maximal number of the route points of a sub-route (6, 12 or 20 route points). Sub-routes are open, because the collection of the components on the feeder bank (depot) is not considered. As component shapes require pertinent nozzle types, one has to search component positions with shape matching to the current nozzle type in the revolver. The number of different nozzle types and frequencies of each nozzle type is fixed when the nozzles are assigned to the head. The nozzle set is changed only when the product (PCB type) is changed. The closest placement position for the first nozzle of the head is selected as starting point for each sub-route (task block).

The assembly time calculation for each head and head pair consists of three steps:

(1) Assigning nozzle types according to their usage frequencies to the head. First, the total number of the components m to be placed on the PCB is determined for the assembly job. Next, the number of different nozzle types needed k and the number of component placements p_z $(z = 1, \dots, k)$ for each nozzle type z is counted. One copy of each nozzle type is initially allocated to the head. The remaining $H - k$ spindles of the head are then allocated to different nozzle types in proportion to the ratios $e_z = p_z/m$. It is supposed that the nozzles are stored in groups of the same type in decreasing order of group sizes.

(2) Searching placement positions for the nozzles using NNCT algorithm. First, the set of components is divided into k subsets according to the nozzle type needed. Then, for each placement tour and for each placement of the tour, NNCT algorithm is used to find the nearest placement position for the current nozzle type from the nozzle type sequence of the head. Note that some of the nozzles may be bypassed during some later tours, that is, there are no more components to place with that nozzle type.

(3) Calculating assembly times using maximum (i.e. L_∞) metrics. Concerning the movement between the placement locations of the jth and $(j+1)$th components of the hth tour $(h = 1, 2, \dots, r)$, there are three movement types to take into consideration. Namely, the x-movement $t_x^{h,j}$ and y-movement $t_y^{h,j}$ on the PCB area, as well as the head rotation $t_r^{h,j}$. The x and y -movements are calculated from the known speed and acceleration values given by ASM, that is, they are assumed to roughly follow the three-part pattern consisting of acceleration, moving

with maximum speed, and deceleration, the middle part being absent for short movements. The head rotation is performed by a stepping motor, whose steps each require a certain constant time. The time spent on head rotation thus is a multiple of the constant, depending on the order of the nozzles on the head. The actual movement time $t_c^{h,j}$ spent for moving between the placement locations of the jth and $(j+1)$th components is the maximum of these three quantities:

$$t_c^{h,j} = \max\left(t_x^{h,j}, t_y^{h,j}, t_r^{h,j}\right). \tag{3}$$

Note also that for the first and last components of the tour, this movement is between PCB and the feeder area, which is numbered as being a part of the head interchange time t_s, that is, the time spent for switching from using head 1 to head 2 and vice versa. For each component to be placed, there is also an additional, independent z-movement t_p, indicating the time spent on the actual placement of the component on PCB. Both t_s and t_p are assumed to be constants.

Accordingly, the total time T_h of the hth placement tour is:

$$T_h = t_s + \sum_{j=1}^{N_h-1} t_c^{h,j} + N_h t_p\,. \tag{4}$$

Note that the actual number of components N_h for each pick-and-place tour must be calculated for each tour separately, because it depends on the compatibility of the components with the nozzles (e.g. the components are nozzle specific as they can only be placed with certain nozzles). Moreover, even in cases where only a single nozzle type is required, the number of components to be placed may not necessarily be a multiple of the placement heads capacity. These constrains, in turn, also affect the overall number of tours.

The total time for performing all r placement tours of a PCB is:

$$T = \sum_{h=1}^{r} T_h\,. \tag{5}$$

The constant parameters are: rotation time $t_r = 0.07$ sec, nozzle placement time $t_p = 0.05$ sec, head interchange time $t_s = 0.3$ sec.

5. Regression modelling

This section discusses another approach to the estimation of the optimal production times of PCB jobs from the input of the control code optimization. Modern regression methods are simple but surprisingly efficient tools for automatizing tasks difficult for a human expert to model. As will be supported by the experimental results, the above described NNCT method taking advantage of rules hand-crafted by a human expert performs significantly worse than a simple data driven regression approach. In contrast to the previous section, the knowledge about the operations principle of the machine is not utilized. Instead, machine learning techniques are used to look from a large set of candidate features which ones are the most useful in modelling the production of the PCBs in a training data set.

The base regression algorithm employed here is in the literature called as regularized least-squares (RLS) or ridge regression (Hoerl and Kennard 1970). Given a set $\mathcal{Z} = \{(\boldsymbol{x}^1, t^1), \ldots, (\boldsymbol{x}^m, t^m)\}$ consisting of m input-label pairs, where $\boldsymbol{x}^i$ are data points and t^i their real valued labels (i.e. the production times of boards), training of the RLS can be considered as solving the following optimization problem:

$$T(\mathcal{Z}, \mathcal{S}) = \operatorname*{argmin}_{\mathbf{w} \in \mathbb{R}^{|\mathcal{S}|}, w_0 \in \mathbb{R}} \left\{ \sum_{(\boldsymbol{x}, t) \in \mathcal{Z}} (\boldsymbol{x}_{\mathcal{S}}^{\mathrm{T}} \mathbf{w} + w_0 - t)^2 + \lambda (\mathbf{w}^{\mathrm{T}} \mathbf{w} + w_0^2) \right\}, \tag{6}$$

where $\boldsymbol{x}_{\mathcal{S}} \in \mathbb{R}^{|\mathcal{S}|}$ is the feature representation of a data point $\boldsymbol{x}$ determined by the set $\mathcal{S}$ of feature indices. The first term in the objective function measures the squared regression error on the training set and the second term, called the regularizer, penalizes models with too large norms. The trade-off between these two is controlled with the regularization parameter $\lambda > 0$. Throughout the experiments, the value of the regularization parameter λ is set to 1, which was found to be appropriate with cross-validation (CV). The regression functions obtained by solving problems of type (6) are of the form

$$f(\boldsymbol{x}) = \boldsymbol{x}_{\mathcal{S}}^{\mathrm{T}} \mathbf{w} + w_0, \tag{7}$$

where $\boldsymbol{x}$ is a column vector for which the prediction is to be made, $\mathbf{w} \in \mathbb{R}^{|\mathcal{S}|}$ is a column vector representation of the learned model, and w_0 is called the intercept. The feature representation of the data and the size $|\mathcal{S}|$ of the linear models under consideration are determined by the feature selection process, which is described in Algorithm **1**.

Algorithm 1 Greedy forward feature selection for regularized least-squares

procedure GreedyRLS($\mathcal{Z}$)
    $\mathcal{S} \leftarrow \emptyset$
    $\epsilon \leftarrow \infty$
    loop
        $b \leftarrow \operatorname{argmin}_{i \in \{1,\ldots,n\} \setminus \mathcal{S}} H(\mathcal{Z}, \mathcal{S} \cup \{i\})$
        $\epsilon_b \leftarrow H(\mathcal{Z}, \mathcal{S} \cup \{b\})$
        if $\epsilon_b >= \epsilon$ **then**
            break
        end if
        $\mathcal{S} \leftarrow \mathcal{S} \cup \{b\}$
        if $|\mathcal{S}| = n$ **then**
            break
        end if
        $\epsilon \leftarrow \epsilon_b$
    end loop
    return $\mathcal{S}$
end procedure

Some of the present authors have recently proposed an algorithm, a linear time feature selection algorithm for RLS called greedy RLS (Pahikkala et al. 2012). Here, linear time refers to the computational complexity $O(mnk)$ of greedy RLS training, where m is the number of training examples, n is the overall number of feature candidates in the training data, and k is the number of features selected by the algorithm (e.g. $k = |\mathcal{S}|$). The value of k can either be a constant given by the user (indicating that the selection process stops exactly when k features have been selected), or it can be determined by the selection criterion (indicating that the selection process stops when selecting more features does not improve the criterion anymore). In the experiments, the latter approach is adopted since it eliminates the extra degree of freedom associated to selecting k. The outcome of the greedy RLS algorithm is exactly equivalent to that obtained by performing a wrapper-based feature selection for RLS regression with leave-one-out cross-validation (LOOCV) as a selection criterion (see Kohavi and John [1997]), which is achieved via computational short-cuts rather than computationally expensive retraining of the model from scratch.

In Algorithm **1**, H denotes the LOOCV criterion that is computed during each selection round for each set $\mathcal{S} \cup \{i\}$, where $\mathcal{S}$ is the set of feature indices already selected during the previous iterations and i denotes an index of a feature not in $\mathcal{S}$. The value of H for $\mathcal{S} \cup \{i\}$ is the LOOCV error of a RLS regressor trained using the features $\mathcal{S} \cup \{i\}$, that is, it is equivalent to the value obtained from

$$H(\mathcal{Z}, \mathcal{S} \cup \{i\}) = \sum_{(\mathbf{x},t) \in \mathcal{Z}} (f_{\mathcal{Z} \setminus (\mathbf{x},t), \mathcal{S} \cup \{i\}}(\boldsymbol{x}) - t)^2, \tag{8}$$

where the function $f_{\mathcal{Z} \setminus (\boldsymbol{x},t), \mathcal{S} \cup \{i\}}$ denotes the RLS regressor of type (7) whose parameters $\mathbf{w}$ and w_0 are obtained from (6) with the data $\mathcal{Z} \setminus (\boldsymbol{x}, t)$ and with the features indexed by $\mathcal{S} \cup \{i\}$. For a more detailed description about the computational short-cuts employed in the algorithm, especially those used to speed up the computation of (8), see Pahikkala et al. (2012).

Cross-validation and its extreme form, LOOCV, are well-known methods for evaluating the quality of models inferred from data (see e.g. Elisseeff and Pontil [2003]). In LOOCV, each training example is held out from the training data at a time and used for evaluation while the rest of the data is used for training, and the results found in this way are averaged. Since only one training example is removed from the training set during each cross-validation round, LOOCV is known to be an almost unbiased estimator of the learning performance. However, there is an inherent problem here. Since greedy RLS uses LOOCV as a feature selection criterion, it cannot be used for evaluating the prediction performance of the learned model, because LOOCV would then over-fit to the selection process. To take account of this, one can resort to the so-called nested cross-validation (see e.g. Varma and Simon [2006]), in which an outer LOOCV loop is used for prediction performance evaluation so that the test sets of the outer LOOCV are independent of the feature selection process done within the training set with the inner LOOCV. This outer LOOCV loop is illustrated in Algorithm **2**.

Algorithm 2 Outer LOOCV for regression performance estimation

$\epsilon \leftarrow 0$
for $i = 1, \ldots, m$ **do**
  $\mathcal{Z}^i \leftarrow \{(\boldsymbol{x}^1, t^1), \ldots, (\boldsymbol{x}^m, t^m)\} \setminus \{(\boldsymbol{x}^i, t^i)\}$
  $\mathcal{S} \leftarrow$GreedyRLS$(\mathcal{Z}^i, n)$
  $\mathbf{w}, w_0 \leftarrow T(\mathcal{Z}^i, \mathcal{S})$
  $p \leftarrow {\boldsymbol{x}^i_{\mathcal{S}}}^{\mathrm{T}} \mathbf{w} + w_0$
  $\epsilon \leftarrow \epsilon + |p - t^i|$
end for
return ϵ

6. Experimental setup

The data used in the experiments consist of a set of 95 PCB jobs, whose assembly times range from 11 to 96 seconds and the number of placed components from 36 to 538 components. Altogether, there are 520 and 16 different component and nozzle types, respectively, encountered in the data.

The NNCT method is not data driven in the sense that it relies on rules designed by a human expert rather than those inferred from a set of training data. Thus, to test the prediction performance of the method, it is simply evaluated on the whole data set of 95 jobs.

The tact time 20,000 components per hour of the machine reported by the manufacturer is used as a non data driven baseline approach. With this approach, the placement time in seconds is predicted simply by multiplying the number of components to be placed with 0.18.

A large set of potential features for describing the characteristics of the boards to be assembled is generated for the greedy RLS regression method. The first feature is the number of components placed to the board under consideration and the second is the number of different nozzles used in the placement. In addition, one feature per each different component type and nozzle type is generated, that is, altogether 520 and 16 component and nozzle type features, respectively. The value of a component type feature for a board is the number of components of that type placed on the board, and the value of a nozzle type feature indicates how many times the nozzle type is used when assembling the board. Thus, the overall number of features available for training the regressor is 538. In addition to these, it is also tested whether the other features used by Vainio et al. (2010) and Wu and Ji (2010) that were applicable on the data used in this study, but they did not have any noticeable effect on the prediction performance.

While the first two features always have a nonzero value, both the component type and nozzle type features are sparse, that is, they have a nonzero value only on a small subset of the data points. Nevertheless, it turns out that as a group they are helpful in improving the accuracy of the regression model. It should be noted that all the features used here are available just prior to the optimization of the control program. In addition to the above features of the PCB jobs, the observed production time of the optimized control programs is given for each training point.

In contrast to the NNCT method, training the greedy RLS regression models require training data, which must be kept independent from the data used for prediction performance estimation in order to avoid biased results. As discussed above, the independence is ensured via performing a nested LOOCV, in which the outer CV is reserved for performance evaluation, and the inner CV is used for feature selection and is separately performed during each round of the outer CV.

In addition to the two above methods, a univariate and a two-variate regression models of the form

$$f(x) = w_0 + w_1 N, \qquad f(x) = w_0 + w_1 N + w_2 v, \tag{9}$$

are tested as baseline methods, where N is the number of components placed on the board and v the number of different nozzle types used during the job x, and w_1, w_1 and w_2 are the regression parameters obtained via ordinary least-squares regression on the training data. These two features were selected, because they were found to carry the strongest signal during the feature selection (see Section 7), and N is also used in other related works, such as by Vainio et al. (2010) and Wu and Ji (2010). The testing of these baselines was performed with an ordinary LOOCV without nesting, because the method did not involve any feature selection.

The prediction performances of the considered methods are measured with three standard measures. Namely, the root mean squared error, the mean absolute error, and the mean absolute percentage error. The measures are evaluated on the whole data set between the true observed assembly times and the predicted ones. The non-parametric Wilcoxon signed ranks test (Wilcoxon 1945) is used for testing the statistical significances of the performance differences between the considered methods. The test is evaluated on the sets of absolute prediction errors on the 95 PCB jobs.

7. Results

The times estimated by NNCT, greedy RLS regression, univariate regression, two-variate regression, and tact time -based methods are illustrated in Figure 3, Figure 4, Figure 5, and Figure 6, and Figure 7, respectively. The jobs in all illustrations are sorted into an ascending order by the real observed assembly times, and the real assembly time is denoted in the figures with plus signs and a line connecting them. The lines with circles, in turn, show the corresponding estimated times.

A summary of the prediction performances for all three methods is given in Table 2. All the performance differences in the table were found to be statistically significant with the Wilcoxon signed ranks test except the difference between the NNCT and the univariate regression method. The NNCT method is clearly better than the simple tact time -based prediction approach, and while the NNCT fails to achieve the prediction performance of the regression approaches, it is not data driven, and hence it does not require any training data in order to adapt to a completely new and different set of PCB jobs. In contrast, the regression methods require training data before they can be deployed for solving a new type of prediction task but, according to the results, provide more accurate predictions when properly trained.

The number of components in the set of 95 PCB jobs is so small that the estimation times for both the NNCT and regression approaches are small, making both methods usable for practical tasks. Namely, the estimator needed about 2 seconds to estimate the assembly times of 190 assembly heads forming the 95 head pairs (PCB jobs), and the estimation time required by the regression approaches is so close to zero that it can not be reliably measured, both timings depending more on the programming language than the model complexity. This is in contrast to the time required for generating the fully optimized control program for the PCB jobs, which is in practise substantial (minutes).

The statistics about the features selected during different rounds of the outer LOOCV are presented in Table 3. Recall, that the feature selection and the training of greedy RLS models are done within an inner LOOCV loop separately during each iteration of the outer LOOCV loop. As a result, the models learned during different rounds of the outer loop may contain different sets of selected features, and hence the statistics of the most commonly selected features are reported rather than a single set. The first feature, indicating the number of components placed on the board, is clearly the most powerful one, as it is selected first during each iteration of the outer LOOCV loop. The second feature, the number of different nozzle types used during the placement process, is almost as useful, since it is also constantly selected as the second feature. The feature indices start to differ at the third iteration, after which the selected features correspond to the numbers of certain components or nozzle types used in the assembly process. The feature indices ranging from 3 to 522 refer to component type features and the ones from 523 to 538 refer to the nozzle type features. The average size of the learned models is approximately 74 features out of 538.

8. Concluding remarks

Prediction of the production times of optimized PCB assembly jobs prior to the actual control code optimization was considered in this work. In particular, the aim was to develop estimators which are accurate and fast to compute, and predict the real assembly times of the optimized control programs without knowing the details of the final machine setup and the optimized machine control program (i.e. feeder assignment, nozzle setting, and placement control).

The two new estimators are based on completely different ideas. The NNCT-estimator is a stripped down machine control optimizer that uses PCB job data (list of component types, positions and nozzle types needed) to assign the different nozzle types to the assembly head and to calculate the assembly times based on movement distances, speeds and accelerations of the head. The method is based on rules handcrafted by a human expert, and hence it does not require training data in the way the data-driven methods do. Thus, it can be considered as a kind of a real machine simulator, while being at least one magnitude faster. It can easily be adapted for other speed characteristics and heads with different number of nozzles.

The second estimator is based on advanced regression analysis technique which does not require any technical knowledge about the assembly machine in use, but relies on a set of training data consisting of PCB jobs with known assembly times, and to a rich set of features consisting of component and nozzle types in addition to their counts. Feature selection is performed automatically by the regressor. The regression estimator requires a representative training set and thus the accuracy of the estimates depends on this data. On one hand, the estimator is not able to estimate assembly times for other head types without retraining. On the other hand, it is very accurate and easy to construct. Further, the regression model is extremely fast to use because as its the estimation time is linear in the number of features used and this number is, in turn, reduced by the greedy feature selection algorithm used for training the estimator. In contrast, the estimation time of the NNCT method grows roughly quadratically in the number of components to be places, because of the nearest neighbour search. Construction of the regression estimator is done off-line and the time for doing this is not counted to the online prediction costs.

The experimental results show that the mean absolute percentage error 6.52% of the NNCT method is considerably lower than the error 17.42% of the estimator based on the manufacturer given tact time only, indicating that it pays to take advantage of the information given by the manufacturer about the operations of the placement head. In addition, the error 3.75% of the proposed data driven regression model based on the extended feature set is lower than the error 6.06% of the classical regression approaches based on the number of components and nozzles only, again indicating the usefulness of the extra features. Furthermore, it can be concluded that the data driven regression approaches are to be preferred over the handcrafted ones whenever there is suitable data available for training them.

Acknowledgments

The authors would like to thank the anonymous reviewers for their very insightful comments. This work has been supported by the Academy of Finland (grant 134020).

References

Altinkemer, K., Kazaz, B., Köksalan, M., and Moskowitz, H. (2000). Optimization of printed circuit board manufacturing: Integrated modeling and algorithms. *European Journal of Operational Research*, 124(2):409–421.

Ayob, M. and Kendall, G. (2008). A survey of surface mount device placement machine optimisation: Machine classification. *European Journal of Operational Research*, 186(3):893–914.

Bentzen, B. S. (2000). Smd placement. Available from: http://web.archive.org/web/20060316170120/http://www.smtinfocus.com/PDF/SMD_placement.pdf.

Elisseeff, A. and Pontil, M. (2003). Leave-one-out error and stability of learning algorithms with applications. In Suykens, J., Horvath, G., Basu, S., Micchelli, C., and Vandewalle, J., editors, *Advances in Learning Theory: Methods, Models and Applications*, volume 190 of *NATO Science Series III: Computer and Systems Sciences*, chapter 6, pages 111–130. IOS Press, Amsterdam, Netherlands.

Emet, S., Knuutila, T., Alhoniemi, E., Maier, M., Johnsson, M., and Nevalainen, O. S. (2010). Workload balancing in printed circuit board assembly. *International Journal of Advanced Manufacturing Technology*, 50(9–12):1175–1182.

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

Grunow, M., Günther, H.-O., Schleusener, M., and Yilmaz, I. O. (2004). Operations planning for collect-and-place machines in pcb assembly. *Computers & Industrial Engineering*, 47(4):409–429.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67.

Johnsson, M. (1999). *Operational and tactical level optimization in printed circuit board assembly*. PhD thesis, University of Turku.

Kallio, K., Johnsson, M., and Nevalainen, O. S. (2012). Estimating the operation time of flexible surface mount placement machines. *Production Engineering, Research and Development*, 6(3):319–328.

Knuutila, T., Hirvikorpi, M., Johnsson, M., and Nevalainen, O. (2004). Grouping pcb assembly jobs with feeders of several types. *International Journal of Flexible Manufacturing Systems*, 16(2):151–167.

Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97:273–324.

Laakso, T., Johnsson, M., Johtela, T., Smed, J., and Nevalainen, O. (2002). Estimating the production times in pcb assembly. *Journal of Electronics Manufacturing*, 11(2):161–70.

Lapierre, S. D., Debargis, L., and Soumis, F. (2000). Balancing printed circuit board assembly line systems. *International Journal of Production Research*, 38(16):3899–3911.

Mentor Graphics (2012). vplan by mentor graphics. Available from:

www.mentor.com/pcb-manufacturing-assembly/products/vplan/.

Michalos, G., Makris, S., Papakostas, N., Mourtzis, D., and Chryssolouris, G. (2010). Automotive assembly technologies review: challenges and outlook for a flexible and adaptive approach. *CIRP Journal of Manufacturing Science and Technology*, 2(2):81–91.

Pahikkala, T., Okser, S., Airola, A., Salakoski, T., and Aittokallio, T. (2012). Wrapper-based selection of genetic features in genome-wide association studies through fast matrix operations. *Algorithms for Molecular Biology*, 7(1):11.

Pyöttiälä, S., Knuutila, T., Johnsson, M., and Nevalainen, O. (2013). Minimizing the assembly cycle time on a revolver gantry machine. *Computers & OR*, 40(11):2611–2624.

Pyöttiälä, S., Knuutila, T., Johnsson, M., and Nevalainen, O. S. (2009). Solving the feeder assignment on a revolver-head gantry machine. In Filipe, J., Cetto, J. A., and Ferrier, J.-L., editors, *Proceedings of 6th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2009)*, volume 1 - Intelligent Control Systems and Optimization, pages 75–80. INSTICC Press.

Pyöttiälä, S., Knuutila, T., and Nevalainen, O. S. (2006). The selection of nozzles for minimizing the number of pickups on a multi-head placement machine. Technical Report 738, Turku Centre for Computer Science.

Raduly-Baka, C., Knuutila, T., Johnsson, M., and Nevalainen, O. S. (2008). Selecting the nozzle assortment for a gantry-type placement machine. *OR Spectrum*, 30(3):493–513.

Raduly-Baka, C., Knuutila, T., Johnsson, M., and Nevalainen, O. S. (2010). Construction of component tapes for radial placement machines. *Computers & Operations Research*, 37(8):1488–1499.

Rogers, P. and Warrington, R. (2004). Production planning for surface mount technology lines. *International Journal of Production Research*, 42(13):2693–2718.

Rong, A., Toth, A., Nevalainen, O. S., Knuutila, T., and Lahdelma, R. (2009). Pcb assembly line modeling with modular placement machines. In Sun, H., Jiao, R., and Xie, M., editors, *Proceedings of the 2009 IEEE International Conference on Industrial Engineering and Engineering Management*, pages 1097–1101. IEEE.

Rong, A., Toth, A., Nevalainen, O. S., Knuutila, T., and Lahdelma, R. (2011). Modeling the machine configuration and line-balancing problem of a pcb assembly line with modular placement machines. *International Journal of Advanced Manufacturing Technology*, 54(1–4):349–360.

Seth, A., Klabjan, D., and Ferreira, P. M. (2009). A novel local search integer-programming-based heuristic for pcb assembly on collect-and-place machines. Technical report, Northwestern University.

Siplace (2012). Asm assembly systems. Available from: http://www.siplace.com/.

Tirpak, T., Mohapatra, P., Nelson, P., and Rajbhandari, R. (2002). A generic classification and object-oriented simulation toolkit for smt assembly equipment. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 32(1):104–122.

Toth, A., Knuutila, T., and Nevalainen, O. S. (2010). Reconfiguring flexible machine modules of a pcb assembly line. *Production Engineering Research and Development*, 4(1):85–94.

Vainio, F., Maier, M., Knuutila, T., Alhoniemi, E., Johnsson, M., and Nevalainen, O. S. (2010). Estimating printed circuit board assembly times using neural networks. *International journal of production research*, 48(8):2201–2218.

Varma, S. and Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, 7(91).

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics*, 1:80–83.

Wu, Y. and Ji, P. (2010). A solution method for the component allocation problem in printed circuit board assembly. *Assembly Automation*, 30(2):155–163.

Yilmaz, I. O., Grunow, M., Günther, H.-O., and Yapan, C. (2007). Development of group setup strategies for makespan minimization in pcb assembly. *International Journal of Production Research*, 45(4):871–897.

Yilmaz, I. O., Günther, H.-O., and Jain, S. (2009). Simulation of mixed-model pcb assembly lines with group setup and bypass conveyors. *The International Journal of Advanced Manufacturing Technology*, 42(3–4):335–347.

Table 1. Methods for estimating the total component placement time

Method	Accuracy	Speed	Suits for line balancing	Used in
Test runs with real placements in “cold-operation” mode	High	Very slow	No	Fine-tuning of mass products
Machine simulation (Vainio et al. 2010)	High	Slow	No	Fine tuning of production
Machine parameter -based averaging (Toth et al. 2010)	Varies from high to low	Fast	Yes	Various applications
Machine learning approach (Laakso et al. 2002, Wu and Ji 2010, Vainio et al. 2010) and this article	Depends on the training set and on the model	Fast	Yes	Various applications

Table 2. Prediction performances of the five evaluated prediction approaches, namely the nearest neighbour of certain type (NNCT), greedy regularized least-squares (Greedy RLS), the univariate regression (UV regr.) and two-variate regression (TV regr.) approaches given in (9), and the use of the tact time (Tact time). The prediction performances are measured in root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE).

	NNCT	Greedy RLS	UV regr.	TV regr.	Tact time
RMSE	3.249	1.752	3.130	2.267	6.184
MAE	2.540	1.263	2.566	1.867	5.419
MAPE	6.52	3.75	7.27	6.06	17.42

Table 3. Statistics of the most commonly selected features during the first five rounds of the selection processes performed during the 95 rounds of the outer LOOCV loop. The rows are indexed by the rounds of the feature selection procedure, i.e. the number of features in the model. The second column contains lists of (feature index: frequency) pairs. The frequency gives the numbers of times the feature has been selected during the corresponding outer LOOCV round. In the third round, for example, the features 530, 496, and 494 are, respectively, selected as the third feature in 63, 31, and 1 out of 95 of the outer LOOCV rounds.

Round	Features
1	1:95
2	2:95
3	530:63 496:31 494:1
4	524:62 446:29 22:2 523:1 365:1
5	365:56 22:26 136:4 221:3 382:2 524:1 270:1 446:1 371:1

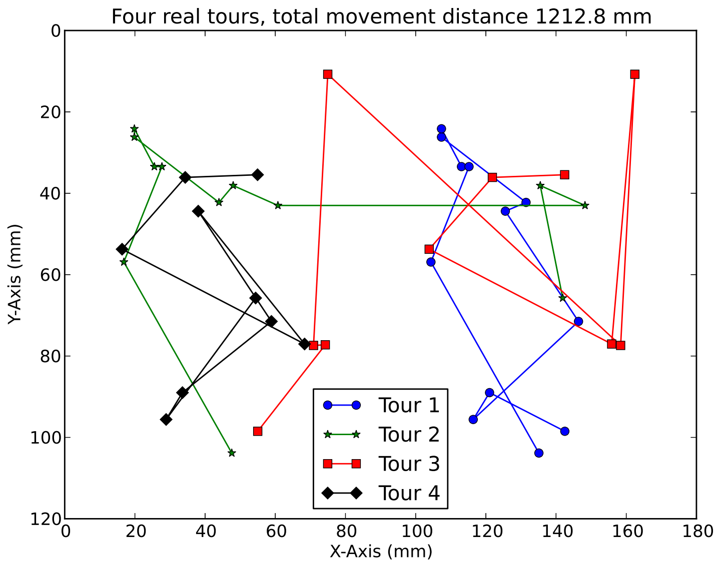

Figure 1. Four placement tours using five different nozzle types for an optimized sample PCB job with 43 component placements.

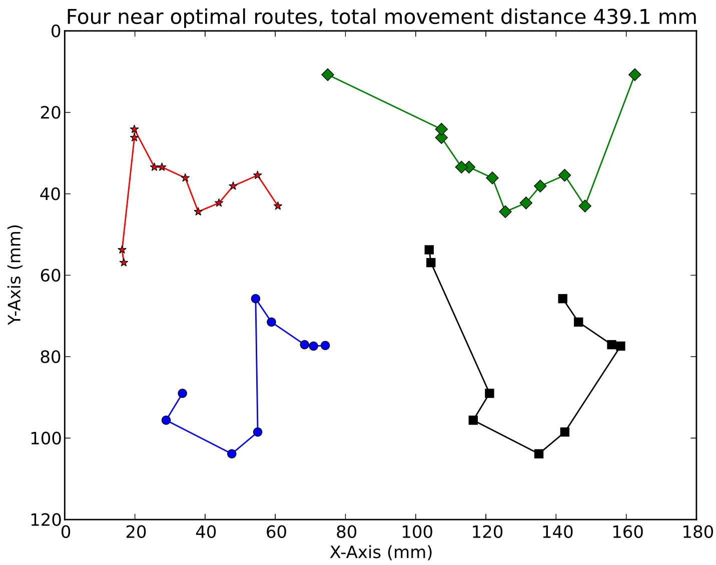

Figure 2. Four near optimal tours for the placement job of Figure 1 when all components use a universal nozzle type.

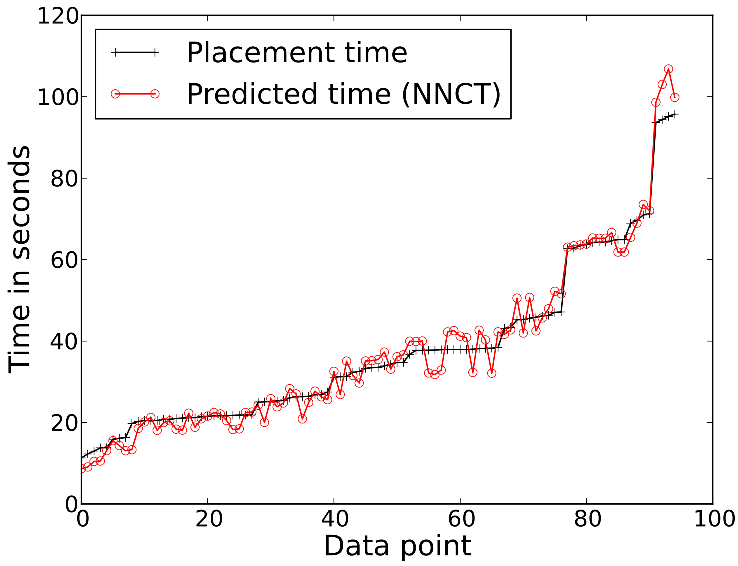

Figure 3. Observed (optimized) ASM Siplace times compared to the NNCT-estimate times.

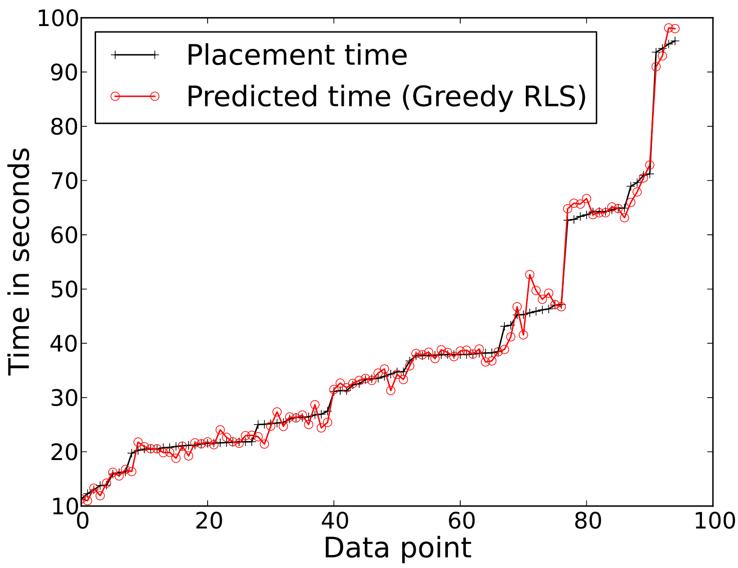

Figure 4. Greedy RLS. The observed and predicted placement times for the 95 PCB assembly jobs. The jobs are in the same order as in Figure 3.

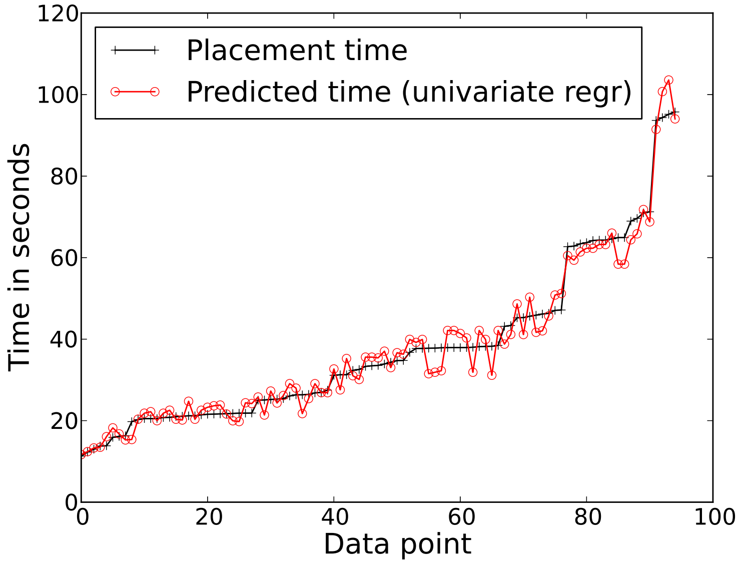

Figure 5. The univariate regression method given in (9). The observed and predicted placement times for the 95 PCB assembly jobs. The jobs are in the same order as in Figure 3.

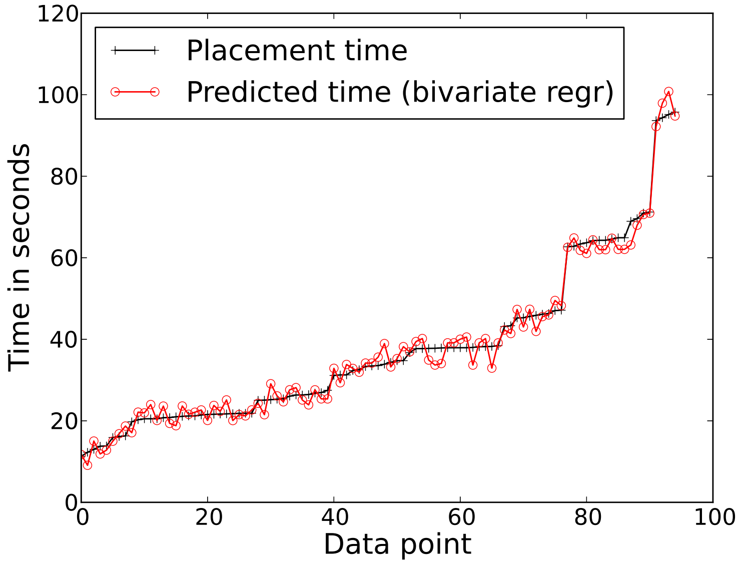

Figure 6. The two-variate regression method given in (9). The observed and predicted placement times for the 95 PCB assembly jobs. The jobs are in the same order as in Figure 3.

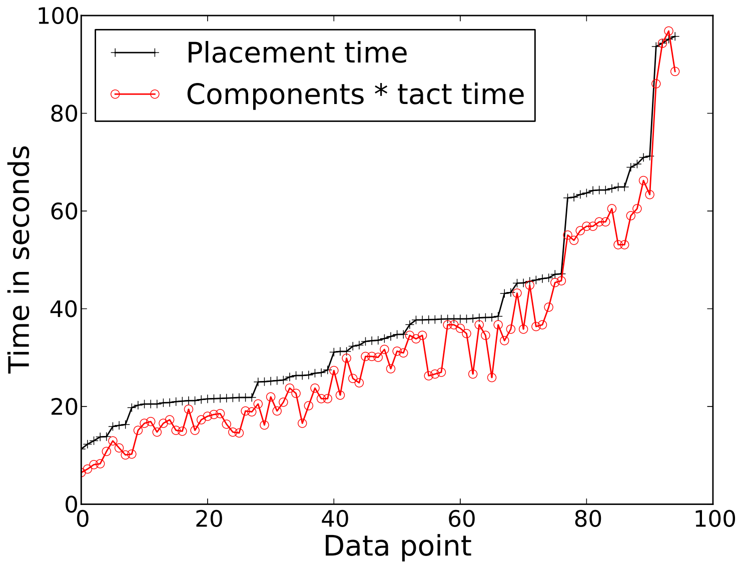

Figure 7. Predicting placement times with tact time. The observed and predicted placement times for the 95 PCB assembly jobs. The jobs are in the same order as in Figure 3.