International Journal of Systems Signal Control and Engineering Application 5 (2-6): 31-46, 2012
ISSN: 1997-5422
© Medwell Journals, 2012

Improving the Throughput of Component Placement Machines by Nozzle Optimisation

Masri Ayob and Mohd Zakree Ahmad Nazri
Data Mining and Optimisation (DMO) Research Group, Center for Artificial Intelligence,
Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600 UKM,
Bangi, Selangor, Malaysia

Abstract: Surface mount device placement machines are designed to place electronic components onto a printed circuit board. In this research, researchers improve the throughput of an economical and medium speed surface mount device placement machine that has four fixed feeder carriers, a fixed printed circuit board table, two vision cameras, a tool bank, a trash bin and a positioning arm head. A head (which is moveable in both X and Y axes simultaneously) is equipped with two pipettes. This research has been adapted from the approach of Magyar (which they applied to a different machine). As in their method, researchers adopt a hierarchical approach. The first stage is to determine the assignment of nozzles to the pipettes that aims to minimise the nozzle change operations and the number of placement groups in order to improve machine throughput. By using the output from the first stage (nozzle assignment) as input to the second stage determines the component pick-and-place schedule. Due to the problem constraints, researchers have to modify the first stage of the Magyar approach and have designed a new approach for sequencing the component pick-and-place operation. Researchers also integrate the Magyar approach with a random descent method in determining the nozzle assignment. Computational results indicate that on average, the approach is superior to Magyar's approach by 4.3% when considering components placed per hour.

Key words: Heuristic, scheduling, component placement sequencing, printed circuit board assembly, nozzle optimisation, Malaysia

INTRODUCTION

Most electronic devices have at least one PCB (Printed Circuit Board) which is a board that contains layers of circuitry, used to connect components. An SMD (Surface Mount Device) placement machine is used to assemble hundreds or even thousands of electronic components onto a PCB. The miniaturisation of component designs and the increasing density of components that can be placed onto a PCB places ever increasing demands on the automation of the printed circuit board assembly process (Moyer and Gupta, 1996a, b) which subsequently has an effect on machine throughput.

Crama *et al.* (2002), Ji and Wan (2001) and McGinnis *et al.* (1992) addressed some optimisation problems which arise in production planning for the assembly of PCBs. These are: Grouping (i.e., assigning PCB types to product families and to machine groups); allocation (i.e., identifying which machine in the assembly line should assemble which components) and arrangement and sequencing (i.e., assigning component feeders to slots on the feeder bank/carrier and sequencing the component pick-and-place operations). These sub problems are tightly intertwined and probably NP-hard (Garey and Johnson, 1979) and most practical instances are difficult to solve to optimality in a reasonable time (Ellis *et al.*, 2001; De Souza and Lijun, 1995). For example, the concurrent movement of many machine parts (such as turret rotation, feeder carrier and PCB table movement) requires a full examination of all feasible combinations of feeder setup and component retrieval sequence in order to determine the best feeder setup and component retrieval sequence for each feasible solution of the component pick-and-place sequence. In addition, there are many other issues that should be considered in optimising these sub problems such as the grouping of components in a sub tour or placement group (i.e., what components should be picked-and-placed together in each tour if there are many pipettes per head), the speed difference between the movement of the PCB table, the feeder carrier and the placement head, the component transportation time, simultaneous pickups, etc. Many researchers have simplified the problem and modelled it as

---

Corresponding Author: Masri Ayob, Data Mining and Optimisation (DMO) Research Group, Center for Artificial Intelligence, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600 UKM, Bangi, Selangor, Malaysia

a travelling salesman problem (Duman and Or, 2004; Jeevan *et al*., 2002; Kumar and Luo, 2003), a vehicle routing problem (Grunow *et al*., 2004) or a quadratic assignment problem (Duman and Or, 2007; Leipala and Nevalainen, 1989; Moyer and Gupta, 1996a). Some researchers have tackled the sub problems independently (by making assumptions about the rest of the sub problems) (Grunow *et al*., 2004) whilst others prefer to solve the problem in an integrated way (Ho and Ji, 2003, 2004, 2006; Shih *et al*., 1996) or by using a hierarchical approach (Magyar *et al*., 1999).

Various types of SMD placement machine such as turret, multi-head, dual-delivery, etc. (Ayob and Kendall, 2008; Bentzen, 2000; Jeevan *et al*., 2002), varies the nature of the problem due to differences in technological characteristics and operational methods of the machines (Crama *et al*., 2002; Moyer and Gupta, 1997). Consequently, many researchers solved the problem as a unique problem since the problem relies heavily on the machine characteristics (Ho and Ji, 2003). This causes difficulties in applying or comparing the various approaches from the literature.

As far as we know, there has been relatively little research that has addressed the importance of nozzle optimisation. Nozzle optimisation in the context of single machine optimisation, involves searching for an effective nozzle assignment and sequencing in order to improve the machine throughput. Some researches that have discussed the importance of nozzle optimisation in the context of single SMD placement machine optimisation, include Ahmadi *et al*. (1988, 1991), Jeevan *et al*. (2002), Magyar *et al*. (1999), Safai (1996), Shih *et al*. (1996) and Tirpak *et al*. (2000). They considered the nozzle optimisation problem together with the problem of sequencing the pick-and-place operation and/or feeder setup.

For example, Shih *et al*. (1996) employed an expert system approach to minimise a multi-station SMD placement machine. They first minimise nozzle changes and then optimise the component pick-and-place operation. They grouped the components in a placement sequence such that components using the same nozzle type can be placed consecutively in order to minimise nozzle changes. Next using the output from the nozzle minimisation stage, Shih *et al*. (1996) used a simple descent search algorithm to optimise the component pick-and-place operation.

Computational results were verified by machine experts and showed an improvement of 5.72% in terms of component placement time which could result in a saving of about 15 working days a year. By considering the importance of optimising the nozzle change operation, Jeevan *et al*. (2002) applied a genetic algorithm (Sastry *et al*., 2005) to minimise the component pick-and-place sequence of the multi-head placement machine. They used the distance of a TSP tour (which represents a total pickup-and-placement distance) as a fitness function. In order to eliminate any unnecessary nozzle changes, they use all the components that can be placed by a certain nozzle before changing the nozzle.

Magyar *et al*. (1999) used a hierarchical problem solving approach to solve the nozzle and sequence component pick-and-place optimisation problems. Their approach significantly improved the assembly cycle time when tested on real industrial problems. On two of the tested PCBs, they achieved assembly cycle time savings of 7.50 and 5.71%.

This study focuses on optimising the nozzle assignment to the placement head and optimising component pick-and-place operations (in a context of single machine and single board type) so as to enhance machine throughput. A nozzle change operation is very time consuming (Crama *et al*., 1990; Jeevan *et al*., 2002; Magyar *et al*., 1999; Safai, 1996; Shih *et al*., 1996). For example, the HP-110 takes about two seconds for a nozzle changeover. Unnecessary nozzle changes can be avoided by optimising nozzle assignment/sequencing when optimising the component pick-and-place operation. This work adapts and extends the nozzle optimisation approach proposed by Magyar *et al*. (1999). The novelty and contribution of this researche is in designing a new nozzle optimisation approach which integrates nozzle layer (Magyar *et al*., 1999) with random descent search in determining a good nozzle assignment.

MATERIALS AND METHODS

Hybrid pick-and-place machine: This research studies a hybrid pick-and-place machine, a new DIMA (DIMA SMT Systems, NL, B.V., Beukelsdijk, 5753 PA Deurne) machine called the HP-110 (Dima SMT Systems, 2003) which is a type of multi-head SMD placement machine. Figure 1 is a photograph of the HP-110 SMD placement machine

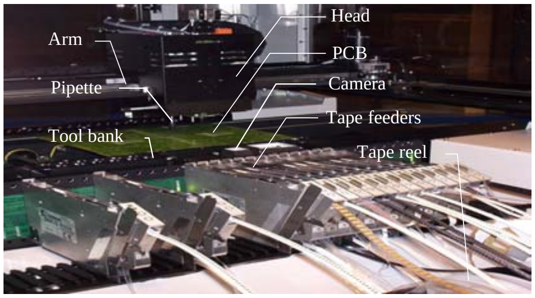

Fig. 1: The HP-110SMD placement machine

(pictured at the DIMA factory). The HP-110 is an economical and medium speed machine that has four fixed feeder carriers (mounted on the four sides of the machine) that hold feeder banks, a fixed PCB table, two vision cameras, a tool bank, a trash bin and a positioning head that is equipped with two pipettes. A feeder bank has several feeder slots where the component feeders are located. These feeders provide the machine with a continuous supply of components. During pick-place operations, the PCB is held in a locked position by the PCB table. The head and arm (sometimes called the robot arm) is movable in the X-Y direction simultaneously. Each pipette holds a nozzle (tool or gripper) that is used to grasp the components. The nozzles are changed automatically, from a tool bank as necessary. The HP-110 has thirteen slots in a tool bank which can hold at most twelve nozzles with one free slot (for use during nozzle change operation where the current nozzle has to be dropped before another is picked up).

In this research, researchers assume that the total number of required nozzles is less than the capacity of tool bank (i.e., <13 in this specific case which includes the nozzles currently in the placement head). Researchers further assume a common pickup priority for all components (no components have to be placed prior to any others) and a common handling time (i.e., the time for pickup, placement and transportation from pickup point to placement point is the same for all components). Usually, there is no pickup or placement priority unless researchers are dealing with small or large sized components that must be placed close to each other or there are multi-level components. In this case, researchers may need to place a smaller component first to avoid interference (Sanchez and Priest, 1991). However, since most of the components on the PCB are small, this problem can easily be solved by assigning larger sized components with a lower priority and treating the board as two separate boards.

Component feeders are available in various types of component packaging: tape, sticks and trays (or waffle) (Ayob and Kendall, 2008). Each packaging type can be associated with many nozzle types and vice-versa. Indeed, one component type can have several types of packaging. This means that each PCB point on the board may be able to accept >1 component packaging type. The component packaging type can be recognised and aligned without using a vision camera (i.e., using mechanical alignment on the fly) by using a small vision camera and/or a large vision camera, depending on the component packaging specification. As the small vision camera is located to the left of the large vision camera then we can have a simultaneous vision and alignment operation (SV) if the left nozzle holds a small vision component and the right nozzle holds a large vision component. That is the two components can be inspected simultaneously which leads to a time saving. It is more economical (in terms of assembly cycle time) to have both mechanically aligned components in a sub tour (MA) rather than having both vision components since the MA sub tour eliminates the time for moving to the camera and performing component recognition and alignment.

The two pipettes on the placement head are fixed at positions such that a Simultaneous Pickup (SP) can occur if the distance between the two pickup points (of the same sub tour) are within a user-defined tolerance (for example when a gap between both pickups is approximately two slots; Appendix B). An SP sub tour can also enhance the throughput of the machine.

The feeder takes a long time (i.e., about 0.5 sec in this case) to transport a component from the component feeder to a pickup point. Therefore researchers should avoid picking up from the Same Component feeder (SC) in a sub tour.

Having four fixed feeder carriers (mounted on the four sides of the machine) also provides a further challenge in optimising the pick-and-place operation of this machine, since a pickup from the Same Feeder bank/carrier (SF) in a sub tour is better (in term of assembly cycle time) than a pickup from Different Feeder banks/carriers (DF). It is apparent from the various situations, conditions and constraints described above that optimising the assembly cycle time of this machine is a challenging scheduling problem.

The operation of the machine starts by concurrently loading a PCB into the machine and reading the PCB and feeder setup information. Then, the head travels above the PCB to check the fiducial marks (these are special points that are located at the corners of the PCB (Magyar *et al*., 1999) to ensure the proper positioning of the PCB. Components are assembled onto the PCB guided by the scheduling and control software that has been installed in the SMD placement machine). Finally, once completed (or partially completed, e.g., due to component runs out or job completion), the PCB is removed from the SMD placement machine and the next PCB is loaded. Before undergoing the solder reflow operation (a soldering process to adhere components on the PCB), the components are secured onto the PCB using adhesive or solder paste (Leu *et al*., 1993).

A sub tour begins with the robot arm moving from the last placement point (or for the first placement sub tour, the movement is from the last fiducial mark) concurrently in the X and Y direction to pickup the appropriate component(s) (one or two (at most)) from the feeder(s). This assumes that the head is already equipped with a correct nozzle. Otherwise, a nozzle change is required and a sub tour begins when the robot arm performs a nozzle change and travels from the tool bank to the pickup points. If this is not a Simultaneous Pickup (SP) then the robot arm needs to move to the second pickup point to pickup the second component after picking up the first component. Next, the robot arm travels concurrently in the X and Y direction and positions itself at the cameras for component recognition and alignment if the component has to be recognised and aligned using camera. If this is not a Simultaneous Vision (SV), the robot has to perform the two component visions sequentially with the added overhead of the robot arm having to position the next pipette/nozzle at the larger vision camera and extra component recognition/alignment time. If a defective component is found, the robot moves to throw the rejected component into the trash bin (located near the camera). Next, the robot arm travels concurrently in the X and Y direction to place the components simultaneously/sequentially (depending on the placement positions) onto the PCB. After completing a sub tour, the robot arm returns to the feeder location to begin another sub tour (if a nozzle change is not required). In the research, researchers assume that there are no defective components. However if there are defective components then researchers assume the robot arm will re-pick the component. Of course, this incurs extra pickup cost.

The assembly cycle time: In this research, researchers aim to minimise the total assembly Cycle Time (CT). That is the total time taken by the machine to assemble all the components onto a Printed Circuit Board (PCB). The CT can be used to evaluate the quality of a schedule because the throughput of the machine can be increased by minimising the CT. The following notations are used to describe the scheduling model of the HP-110 (adopted from Ayob and Kendall, 2004, 2005):

CT = The assembly cycle time to assemble all components

B = The total number of sub tours

λ = The time for picking up a component

θ = The time for placing a component

j = The jth sub tour index where $j \in \{1, 2, \ldots, B\}$

$I(j)$ = The time taken for the robot arm to travel from feeder carrier/slot to PCB point and place the component(s) in the jth sub tour

$P(j)$ = The time taken for the robot arm to travel from PCB point (of the (j-1)th sub tour) to feeder carrier/slot and pick the component(s) in the jth sub tour

$\Phi_0(j)$ = The time taken for the robot arm to move from PCB point (of the (j-1)th sub tour) to pickup the first component in the jth sub tour from a component feeder

$\Phi_1(j)$ = The time taken for the robot arm to move from current feeder slot (pickup point) to the next feeder slot in the jth sub tour

$b_0(j)$ = The time taken for the robot arm to travel from the camera (or pickup point for the mechanical alignment case) to the first PCB point in the jth sub tour

$b_1(j)$ = The time taken for the robot arm to travel from the first PCB point to the second PCB point in the jth sub tour

$C_0(j)$ = The time taken for the robot arm to travel from feeder carrier/slot to position the first pipette above the camera in the jth sub tour

$C_1(j)$ = The time taken for the robot arm to position the next pipette above the camera in the jth sub tour

$\rho(j)$ = A decision variable to indicate either there is a second component for pickup-and-placement ($\rho(j) = 1$) or 0 otherwise

$\tau(j)$ = A decision variable for having one camera vision and one mechanical alignment component in a sub tour where $\gamma(j) = 0$ if true or 1 otherwise

$\eta(j)$ = The number of tool change required to pickup the component (s) in the jth sub tour where $\eta(j) \in \{0,1,2\}$

$\gamma(j)$ = A decision variable of simultaneous vision in the jth sub tour where $\gamma(j) = 0$ if exist simultaneous vision or 1 otherwise

$\omega(j)$ = A decision variable of simultaneous pickup in the jth sub tour where $\omega(j) = 0$ if exist simultaneous pickup or 1 otherwise

$\sigma(j)$ = A decision variable of having two mechanical alignment components in the jth sub tour where $\sigma(j) = 0$ if having two mechanical alignment components or 1 otherwise

μ = The time for the robot arm to move up/down

Ω = The tool changing time

α = The image acquisition and recognition time

$\psi(j)$ = A decision variable either both components are picked up from the same component feeder in a sub tour ($\psi = 1$) or $\psi = 0$ otherwise

ζ = The component feeder transportation time

The assembly cycle time, CT can be calculated as follows:

$$\mathrm{CT}=\sum_{j=1}^{B}\left[P(j)+I(j)\right] \tag{1}$$

Where:

$$\begin{aligned}P(j)=\Phi_0(j)+\lambda+2*u+\rho(j)*\omega(j)*\\ \left[\max\left(*\Phi_1(j),\psi(j)*\zeta\right)+\lambda+2*u\right]+\eta(j)*\Omega\end{aligned} \tag{2}$$

$$\begin{aligned}I(j)=2*u+b_0(j)+\theta+\sigma(j)*\left(C_0(j)+\alpha\right)+\rho(j)*\\ \left[\gamma(j)*\sigma(j)*\tau(j)*\left(C_1(j)+\alpha\right)+b_1(j)+\theta+2*u\right]\end{aligned} \tag{3}$$

CT is a summation of the total time taken by the robot arm to travel from the PCB point to the feeder(s) and picking up the component(s) (i.e., P(j)) and then travelling back to the placement point (on the PCB) and placing the component(s) (i.e., I(j)). This formulation is adopted from Ayob and Kendall (2004). Researchers ignore some constant costs that only happen once for each board and cannot be optimised such as the time of PCB loading, fiducial checking and PCB loading/unloading. Since, the chances of having simultaneous placement is very small due to the nature of the problem, resarchers also ignore simultaneous placements.

In principle, the objective function (Eq. 1) could be applicable to other types of SMD placement machines such as sequential pick-and-place, multi-station, dual-delivery and particularly the multi-head. Unfortunately, due to the differences among various machine specifications and operational methods, the calculation of P(j) and I(j) are machine dependent (Ayob and Kendall, 2004, 2005).

There are many factors involved in determining the CT of many SMD placement machine such as nozzle changes, simultaneous pickup, simultaneous vision, etc. These factors are very machine dependent. Ignoring these factors in solving component pick-and-place sequencing might not be a good strategy. Unfortunately, many researchers are only concerned with minimising the robot travelling time (and/or feeder carrier and PCB table movement) in order to improve the machine throughput (or particularly, the component pick-and-place sequence). Of course, they might be able to produce a good quality solution. However, they may obtain a much better solution if other factors are also considered. In this research, researchers use the average machine operation time given by DIMA to estimate CT as an evaluation for the heuristic performance. DIMA also recommend using average machine operation time, instead of using the actual speed, acceleration and deceleration of the machine for calculating the machine throughput. As the speed of the robot arm is very fast and the component density on the PCB is increasing (i.e., the distance between PCB points tends to be smaller), minimising the robot travelling distance is becoming a less significant factor for improving the throughput of the machine. Indeed, due the acceleration/deceleration rate of the robot arm, the time taken for the robot arm to move shorter or longer distances might be almost the same. In fact, the effect of pickup/placement point distance compared to other factor such as simultaneous vision is relatively small. For example, a sub tour for simultaneous vision with different feeder bank pickup (SV+DF) takes about 2140 ms whilst sub tours for the Same Feeder bank pickup (SF) and Different Feeder bank pickup (DF) takes about 2480 and 2540 ms, respectively. Therefore, it is ineffective to just minimise the robot travelling time in order to improve the machine throughput. As the machine is embedded with control software for accurate movements/operations, exact information about the machine speed, acceleration/deceleration rate, exact location of cameras, trash bin, tool bank, etc., are less important for the optimisation software. The average machine operation time (Table 1) is adequate for guiding the search for a better quality schedule. Moreover including the machine speed, acceleration/deceleration rate, etc., might introduce a more complex formulation for the objective function. As such the exact location of PCB points are not crucially important in determining the component pick-and-place sequence. Of course, if researchers use average machine operation time, the actual machine throughput is different when testing the generated schedule on the real machine. Similarly, if we only use the machine travelling distance (that is solve it as a TSP as many other researchers do (Jeevan *et al.*, 2002; Duman and Or, 2004), the theoretical and actual machine throughput are different. In fact in a real production environment, reserachers may have defective components, component runs out, machine acceleration/deceleration rates, nozzle changes, simultaneous visions, component feeder transportation times, etc. These factors crucially affect machine throughput. Therefore, what is important for the scheduler is the ability to produce a good quality schedule.

Table 1: The average processing time of the HP-110 (given by the machine's manufacturer)

Operations	Time (m sec)	Description(mm)
Pickup (λ)	10	
Placement (θ)	10	
Axis up/down (u)	50	
Move to XY feeder ($\Phi_0(j)$)	350	200
Move to XY next feeder ($\Phi_1(j)$)	290/350$^+$	150
Move XY to camera ($C_0(j)$)	350	200
Move next pipette to camera ($C_1(j)$)	225	45
Image acquisition and recognition (α)	175	
Move to XY place ($b_0(j)$)	300/410*	150
Move to XY next place ($b_1(j)$)	175	10
Tool changing (Ω)	2000	
The component feeder transportation(ζ)	500	

*Pickup both mechanically aligned components in a sub tour; $^+$Pickup from different feeder banks in a sub tour

Sub tour operation types: Based on the machine specification and operational method, the sub tour operation types can be classified into 17 operation types (Table 2) where SP denotes a simultaneous pickup, SV is a simultaneous vision, SF is a same feeder bank pickup, SC is a same component feeder pickup, DF is a different feeder bank pickup, MA is having two mechanical aligned components, MV is having one mechanical and one vision component, M is having only one mechanical aligned in a sub tour and V is one vision component in a sub tour (i.e., for M and V, there is only one component in a sub tour).

Figures in Table 2 are computed based on the average processing time of the HP-110 (Table 1) by using Eq. 1-3 (i.e., the time taken for completing a sub tour). For each sub tour operation type, the throughput of the machine is measured in components per hour (cph) that is represented as E_s in Table 2. This summarises the machine throughput without a nozzle change operation based on one or two component pickup and placement operations in a sub tour. As in Ayob and Kendall (2005), researchers use a weighted parameter, δ_s to represent the effectiveness of the sth sub tour operation type:

$$\delta_s = \frac{E_s}{E_1} \qquad (4)$$

where, E_1 and E_s are the efficiency of the most efficient sub tour operation type (i.e., MA+SP sub tour) and the sth sub tour operation type, respectively. The E_s and δ_s values are shown in Table 2.

Determine-Nozzle-Layer approach: Let nozzle definition describe a relation between component types and nozzles (Table 3 and 4). Table 3 shows an example of a placement list or PCB points whilst Table 4 shows a list of component types (based on the example of Table 3 and Appendix A) with the nozzles that can be used to pick-and-place the various components. Appendix A presents the specification of the component packages which shows the relation between package name, component type and nozzles that can be used to pick-and-place the component package

Table 2: The weighted value of the sub tour operation type, δ_s

Pickup and placement operation type	Time (m sec)	E_s(cph)	δ_s
MA+SP (MA with SP)	1265	5691	1.000
MA+SF (MA with SF)	1665	4324	0.760
MV+SP (MV with SP)	1680	4285	0.753
SV+SP (SV with SP)	1680	4285	0.753
MA+DF (MA with DF)	1725	4173	0.733
MA+SC (MA with SC)	1875	3840	0.675
M (one M component only)	980	3673	0.645
SP (SP only)	2080	3461	0.608
SV+SF (SV with SF)	2080	3461	0.608
MV+SF (MV with SF)	2080	3461	0.608
SV+DF (SV with DF)	2140	3364	0.591
MV+DF (MV with DF)	2140	3364	0.591
SV+SC (SV with SC)	2290	3144	0.552
SF (SF only)	2480	2903	0.510
DF (DF only)	2540	2834	0.498
SC (SC only)	2690	2676	0.470
V (one V only)	1395	2580	0.453

A Nozzle Usage table contains a Minimum Usage$_k$ (i.e., sum of components that can only be picked up/placed by the kth nozzle), Maximum Usage$_k$ (i.e., a total number of components that can be picked up/placed by the kth nozzle) and Substitution Nozzles$_k$ (which lists the substituting nozzles of the kth nozzle along with the number of the substitution). The entries are computed based on the placement list and the nozzle definition table. For example (Table 5), there are 6 components that have to be picked/placed by nozzle 8 and at most 11

Table 3: An example of 30 PCB points with 10 component types

Component ID	Component type	X	Y
1	4	54.63	79.25
2	2	12.50	119.50
3	2	1.50	53.00
4	9	52.13	98.75
5	10	36.75	92.88
6	5	35.00	72.88
7	7	19.50	12.38
8	10	4.63	111.38
9	6	93.75	72.13
10	1	4.63	27.00
11	7	83.63	93.63
12	4	96.13	108.63
13	3	112.13	84.00
14	1	20.13	107.63
15	5	119.75	84.88
16	5	66.25	53.25
17	8	0.50	29.13
18	2	68.25	49.75
19	9	25.13	105.38
20	4	29.13	71.25
21	1	76.38	48.75
22	1	101.75	22.38
23	3	33.25	55.00
24	1	8.88	69.25
25	2	17.38	100.13
26	3	59.00	16.50
27	10	96.63	52.25
28	4	61.75	90.50
29	8	19.88	98.63
30	6	48.50	26.50

Table 4: A nozzle definition (for the example of Table 3)

Component type	Nozzle ID	Component ID
1	1	10, 14, 21, 22, 24
2	2	2, 3, 18, 25
3	4, 8	13, 23, 26
4	16, 8	1, 12, 20, 28
5	1, 2, 4	6, 15, 16
6	8, 16	9, 30
7	4	7, 11
8	32, 64	17, 29
9	8, 64	4, 19
10	2, 4, 64	5, 8, 27

For example, component type 3 uses nozzles 4 and 8. The PCB point IDs of component type 3 are 13, 23 and 26 components can be picked/placed by nozzle 8. Nozzle 4 and 64 can replace nozzle 8 for 3 and 2 component pickups/placements, respectively. The nozzle with a Minimum Usage = 0 will be discarded (Table 5). The algorithm for generating a Nozzle Usage table is shown in Fig. 2 and has been adapted from Magyar *et al.* (1999) with some minor changes to take into account our problem constraints. The difference is that in the problem, each component type can be associated with >2 nozzles. Where as in Magyar *et al.* (1999), each component type is associated to at most two nozzles (primary and secondary nozzle). Step 3 in Fig. 2, iteratively eliminates the unnecessary nozzles in order to minimise the number of nozzles used. Where less nozzles are used, it may reduce the nozzle changes and thus increase the machine throughput.

As addressed by Magyar *et al.* (1999), there is a trade-off between minimising the nozzle changes and the total number of sub tours. It is important to minimise the number of sub tours as additional sub tours might lead to extra cost such as robot travelling, component recognition/alignment, component pickup and/or placements times. Therefore, Magyar *et al.* (1999) proposed a Determine-Nozzle-Layer approach to search for a good nozzle layer. Magyar *et al.* (1999) defined a nozzle layer table as a hidden layer that lists the assignment of nozzle to the pipette in each sub tour (Table 6). A compressed form of the table, Compressed Nozzle Layer table, summarises the nozzle assignment which indicates the sum of sub tours for each assignment (Table 6). For this example, the number of nozzle changes is 3.

Magyar approach: As by Magyar *et al.* (1999), the procedure Determine-Nozzle-Layer begins with the minimal number of allowed nozzle changes. Then Magyar iteratively increase the number of allowed nozzle changes until they obtained the minimum number of sub tours. Finally, they chose the best nozzle layer according to a weighted cost function, F that has the minimum value. F = B+β*CHG where B, β and CHG denote a sum of sub tours, a weighted value of one nozzle change and the number of nozzle changes, respectively. Researchers also use the same cost function to choose the best nozzle layer. For the HP-110, a nozzle changeover takes about 2 m sec whilst one sub tour (Table 2) costs 1928 m sec. Thus, β = 1.04 that is one nozzle change is equivalent to 1.04 sub tours.

Table 5: A Nozzle Usage table (for the example of Table 3)

Nozzle ID	Minimum Usage$_k$	Maximum Usage$_k$	SubstitutionNozzle$_k$/Sum of components			
Before eliminates the unnecessary nozzles						
1	5	8	(2) 3	(4) 3		
2	4	10	(1) 3	(4) 6	(64) 3	
4	2	11	(8) 3	(1) 3	(2) 6	(64) 3
8	0	11	(16) 6	(4) 3	(64) 2	
64	0	7	(32) 2	(2) 3	(4) 3	(8) 2
16	0	6	(8) 6			
32	0	2	(64) 2			
After eliminates the unnecessary nozzles						
8	6	11	(4) 3	(64) 2		
1	5	8	(2) 3	(4) 3		
2	4	10	(1) 3	(4) 6	(64) 3	
4	2	11	(8) 3	(1) 3	(2) 6	(64) 3
64	2	7	(2) 3	(4) 3	(8) 2	

Greedy descent approach: Researchers extend the research proposed by Magyar *et al.* (1999) by integrating their approach with a greedy descent heuristic in order to optimise the nozzle changes (i.e., procedure Determine-Nozzle-Layer-GD in Fig. 3). As in Magyar *et al.* (1999), the procedure Determine-Nozzle-Layer-GD begins with the minimal number of allowed nozzle changes. However before increasing the number of allowed nozzle changes, researchers first perform a greedy descent local search in order to find a good nozzle layer.

In this approach, the iterative procedure in Determine-Nozzle-Layer-GD algorithm (Fig. 3) starts by creating an initial nozzle layer using a procedure Create-One-Nozzle-Layer Magyar *et al.* (1999). To adhere to our problem constraint, researchers have slightly modified the procedure Create-One-Nozzle-Layer (Magyar *et al.* (1999). That is when creating a new row where the Minimum Usage = 0 and Maximum Usage>0 (Magyar *et al.*, 1999), researchers modify the Nozzle Usage table accordingly by deducting the Maximum Usage of the appropriate nozzles (Fig. 4). Based on

Table 6: A Nozzle Layer and Compressed Nozzle Layer (for the example of Table 3)

Sub tour	Pipette 1	Pipette 2
A Nozzle Layer		
1	2	1
2	2	1
3	2	1
4	2	1
5	2	1
6	2	1
7	2	8
8	2	8
9	2	8
10	4	8
11	4	8
12	4	8
13	4	8
14	64	8
15	64	8
Pipette 1	**Pipette**	**Sum of sub tours**
A Compressed Nozzle Layer		
2	1	6
2	8	3
4	8	4
64	8	2

```
Initialisation:  Let NOZZLES = {set of all the nozzles that are available in the Nozzle Definition table} and ?k?
                 NOZZLES is the k$^{th}$ nozzle where k= {1, 2,…Sum Nozz}. Sum Nozz is total number of available
                 nozzles in the Nozzle Definition table. For all nozzles listed in the NOZZLES, set Minimum
                 Usage$_k$=0, Maximum Usage$_k$=0 and Substitute Nozzle$_k$ is empty. Set k=1.
Iteration:       REPEAT
                   • For each component d in the placement list, do:
                     If component d uses nozzle k, then:
                       • Maximum Usage$_k$= Maximum Usage$_k$ +1;
                       • If component d only uses nozzle k, then Minimum Usage$_k$= Minimum Usage$_k$
                         +1;
                       • Else, increase the substitution value for the Substitute Nozzle$_k$ accordingly.
                   • k = k + 1.
                 UNTIL k>Sum Nozz.
Stop:            • If there are some nozzles in the Nozzle Usage table having MinimumUsage$_k$=0, then exclude
                   one of that nozzle (which has MinimumUsage$_k$=0) from the NOZZLES and re-initialise all the
                   parameter values (MinimumUsage$_k$=0, MaximumUsage$_k$=0, Substitute Nozzle$_k$ is empty and
                   k=1). Go to Step 2.
                 • Else, return a Nozzle Usage table.
```

Fig. 2: Procedure Create-Nozzle Usage table

```
Initialisation:  Let the minimal number of nozzle changes, MINCHG=max(0,E-H), where E and H denote the total
                 number of required nozzles and the number of pipettes per head, respectively. Let the minimal number of
                 sub tours, MING=Q/H, where Q denotes the number of available placement points (If Q%H>0, then
                 MING=1 + Q/H). Let the number of nozzle changes, CHG=MINCHG, B denotes a sum of sub tours and
                 the cost function, F=B + ß*CHG, where ß is a weighted value of one nozzle change. Set B=0, F=0 and
                 ß=0. Empty decision table.
Iteration:       If B>MING:
                   REPEAT
                     • Create one nozzle layer using procedure Create-One-Nozzle-Layer-Magyar. Compute B and
                       F. If CHG=MINCHG, set the best cost function, F$_{best}$=F. Store the F$_{best}$, B, CHG and the
                       obtained nozzle layer into a decision table. Based on the obtained nozzle layer, find the
                       sequence order of nozzle change. Let Z as a set of nozzle in a sequence order of nozzle
                       changes and $z_k \in Z$ is the k$^{th}$ nozzle where k={1,2,..CHG+H}.
                     • Set k=1.
                     • Do {
                         • Set m=max{H+1, k+1}. Repeat the following steps until m=CHG+H or B=MING:
                               Swap $z_k$ with $z_m$. Create one nozzle layer using procedure Create-One-Nozzle-
                               Layer-Magyar. Compute B and F.
                               If F< F$_{best}$: set F$_{best}$=F, k=1, accepts the new nozzle layer and update Z, F$_{best}$, B,
                               CHG and the obtained nozzle layer into a decision table.
                               Else, set m=m+1.
                         • k=k+1.
                       }Until k=CHG+H-1 or B=MING.
                     • If B>MING, let CHG=CHG+1.
                   UNTIL B=MING.
Choose:          Choose the best nozzle layer from a decision table with the minimum F.
```

Fig. 3: Procedure Determine-Nozzle-Layer-GD

the obtained nozzle layer generated by the procedure Create-One-Nozzle-Layer (Magyar *et al.*, 1999), researchers then find a sequence of nozzle changes (Z). Using a 2-opt local search operator (Fig. 3), researchers sequentially swap the sequence order of nozzle changes in order to search for the best nozzle layer (while fixing the number of allowed nozzle changes). This is a greedy descent approach which accepts the first improving solution and moves to this solution. For each fixed number of allowed nozzle changes, the local search sequentially visits some neighbours of Z until the minimum number of sub tours is obtained or researchers reach the limit of visited neighbours. Figure 5 shows an example of how the local search creates the neighbour solutions. In this example (Fig. 5), Z = {2, 1, 8, 4, 64} means that the first 2 nozzles to be applied are nozzle 2 and 1, followed by 8, 4 and 64. The Nozzle Usage table and neighbour solutions in Fig. 4 and 5, respectively are generated based on the example in Table 3 and 4.

Optimising component pick-and-place sequencing: Based on the obtained nozzle layer (produced by the procedure Determine-Nozzle-Layer-GD), researchers schedule the component pick-and-place operations (Fig. 6, procedure schedule-component-pick-and-place-sequence). Researchers first create a Nozzle Component which has a list of E nozzles that defines primary and secondary component types. This indicates the component types that can only be picked and placed by a nozzle (primary) and many nozzles (secondary).

NozzleUsage table before create new row:

Nozzle	MinUsage	MaxUsage	Substitute Nozzle/Sum of components
8	1	6	(4)3, (64)2
1	0	3	(2)3, (4)3
2	4	10	(1)3, (4)6, (64)3
4	2	11	(8)3, (1)3, (2)6, (64)3
64	2	7	(2)3, (4)3, (8)2

NozzleUsage table after create one new row that use nozzle ‘1’ and ‘8’:

Nozzle	MinUsage	MaxUsage	Substitute Nozzle/Sum of components
8	0	5	(4)3, (64)2
1	0	2	*(2)2, *(4)2
2	4	* 9	*(1)2, *(4)5, (64)3
4	2	* 10	(8)3, *(1)2, *(2)5, (64)3
64	2	7	(2)3, (4)3, (8)2

Note: * This depends on the Nozzle Definition of the component type that is currently been scheduled for the new row. Since the MinUsage of nozzle ‘1’=0 (before creating new row), there is no component left that can only be picked by nozzle ‘1’. That is, the component left to be picked by nozzle ‘1’ (3 components) can also be picked by nozzle ‘2’ (3 components) and/or nozzle ‘4’ (3 components).
Figures in bold texts show the affected changes of the MaxUsage and substitution nozzle.

Fig. 4: An example of modifying the Nozzle Usage table when creating a new row

Let E=5, H=2 and Z = {2, 1, 8, 4, 64}

The neighbours of x that will be generated are:
Z’ = {8, 1, 2, 4, 64}, {4, 1, 8, 2, 64}, {64, 1, 8, 4, 2}, {2, 8, 1, 4, 64}, {2, 4, 8, 1, 64},
{2, 64, 8, 4, 1}, {2, 1, 4, 8, 64}, {2, 1, 64, 4, 8}, {2, 1, 8, 64, 4};

or if the first neighbour is accepted:

Z’ = {8, 1, 2, 4, 64}, {4, 1, 2, 8, 64}, {64, 1, 2, 4, 8}, {8, 2, 1, 4, 64}, {8, 4, 2, 1, 64},
{8, 64, 2, 4, 1}, {8, 1, 4, 2, 64}, {8, 1, 64, 4, 2}, {8, 1, 2, 64, 4};

or if the second neighbour is accepted:

Z’ = {8, 1, 2, 4, 64}, {4, 1, 8, 2, 64}, {64, 1, 8, 2, 4}, {4, 8, 1, 2, 64}, ...
:

Fig. 5: An example of how the local search creates the neighbour solutions

```
Input:    A CompressedNozzleLayer, NozzleDefinition and ComponentPackages lists.
Output:   A component pick-and-place schedule.
Initialisation:   Create a NozzleComponent list for E nozzles. Let u denotes an index of the row of the
                  CompressedNozzleLayer, whilst MAXLAYER represents the maximum number of rows. Set u=0. Let
                  $Nozz_p(u)$ denotes the nozzle at the $p^{th}$ pipette for the $u^{th}$ row.
Iteration:        REPEAT
                      • Sequentially schedules the sub tours and PCB points, starts with the primary components then
                        with the secondary components that can be picked up by the nozzle, $Nozz_p(u)$.
                      • Choose the appropriate component packages by selecting the component packages, which
                        allows the most efficient sub tour operation type.
                      • Set u=u+1
                  UNTIL u= MAXLAYER.
Improvement:
                  • Try to eliminate the same component feeder pickup (procedure Eliminate-SC).
                  • Reoptimise the assignment of nozzles to pipettes (procedure Optimise-Nozzle-Pipette-Assignment).
```

Fig. 6: Procedure schedule component pick-and-place sequence

Starting from the first row of the nozzle layer, researchers consecutively schedule the sub tours and PCB points by first scheduling the primary component then the secondary component for the selected nozzles (determined by the procedure Determine-Nozzle-Layer-GD). By determining the sequence of nozzle changes before scheduling component pickup-and-placement, we first have to schedule the component pick-and-place operation for the primary components. Otherwise, we may encounter a problem of having insufficient component types to be scheduled because the component type has been scheduled for other nozzles (Table 7). For this example (Table 7), based on the Compressed Nozzle Layer in Table 7, researchers are unable to schedule component ID = 2 (type 4) if researchers do not schedule a primary component first. This is because nozzle 4 has no component left to be scheduled that is all components that can be picked up by nozzle 4 (i.e., component type 7, 3, 10) have been scheduled for nozzles 4 (2 components of type 7), 8 (3 components of type 3) and 2 (3 components of type 10). This problem can be solved by first scheduling the primary component type or by solving the nozzle and the component pick-and-place optimisation problems, simultaneously. Since, a component type can have several packages, researchers choose a package that allows the most efficient sub tour operation type (Table 2). As researchers use average operation times (Table 1) to compute the assembly cycle time, researchers randomly schedule the appropriate PCB points to be placed after scheduling the component packages to be picked up. After scheduling all the components, researchers try to improve the quality of the schedule by eliminating pickups from the same component feeder (procedure Eliminate-SC in Fig. 7) and applying a local

Table 7: An example of component pick-and-place schedule (for the problem in Table 3) that has been generated without first schedule for the primary component type

Pipett	Pipette	Sum of sub tours
Let a Compressed Nozzle Layer as follows:		
8	1	8
8	2	3
4	2	3
64	64	1
Nozzle ID	Primary component	Secondary component
8	6, 4	3, 9
1	1	5
2	2	5, 10
4	7	3, 10
64	8	9
Sub tour	Pipette 1: (Nozzle ID)/ Component ID/Type/Package	Pipette 2: (Nozzle ID) Component ID/Type/Package
1	(8)/20/4/E	(1)/10/1/B
2	(8)/1/4/E	(1)/24/1/B
3	(8)/28/4/E	(1)/21/1/B
4	(8)/12/4/E	(1)/22/1/B
5	(8)/4/9/M	(1)/14 /1/B
6	(8)/30/6/I	(1)/16/5/G
7	(8)/9/6/I	(1)/6/5/G
8	(8)/23/3/D	(1)/15/5/G
9	(8)/19/9/M	(2)/5/10/A
10	(8)/26/3/D	(2)/27/10/A
11	(8)/13/3/D	(2)/8/10/A
12	(4)/7/7 /J	(2)/3/2/C
13	(4)/11/7/J	(2)/18/2/C
14	(4)/NONE	(2)/25/2/C
15	(64)/17/8 /L	(64)/29/8/L

search to re-optimise the assignment of nozzles to pipettes (procedure Optimise-Nozzle-Pipette-Assignment in Fig. 8). The procedure Eliminate-SC attempts to swap (using a two-opt operator) the components from the sub tours that have the same component feeder pick up (i.e. SC, SV+SC or MA+SC) with other sub tours. The swap is restricted to the same nozzle type. That is researchers only swap the components but not the nozzles. Researchers use PrevScore and NewScore to evaluate the effectiveness of the swap operation. NewScore and PrevScore represent the sum of $\boldsymbol{\delta}_s$ for a trial solution (x') and an incumbent solution (x), respectively. The calculation for PrevScore and NewScore only involve the two $\boldsymbol{\delta}_s$ of the sub tours that are swapped (before and after swap operation, respectively). The trial solution is accepted when the NewScore is better than the PrevScore. This is a greedy hill climbing local search which accepts the first improve solution.

```
Initialisation:   Let $\delta_s$ represents the effectiveness of the $s^{th}$ sub tour operation type. NewScore and PrevScore
                  represent the sum of $\delta_s$ for a trial solution and an incumbent solution, respectively. Set NewScore=0
                  and PrevScore=0.Let p as an index of pipette position. Set the sub tour index, u=0, and a decision
                  variable, NewNozzle=0. Let $Nozz_p(u)$ denotes the nozzle at the $p^{th}$ pipette for the $u^{th}$ sub tour.

Iteration:        REPEAT
                      • If the $u^{th}$ sub tour operation type is "SC" or "SV+SC" or "MA+SC":
                        Find the other sub tour, u' that has at least one nozzle of the same type with the
                        $Nozz_p(u)$. If found:
                            • Compute: PrevScore=$\delta_s$ of the $u^{th}$ sub tour + $\delta_s$ of the $u'^{th}$ sub tour.
                            • Swap the appropriate components and PCB points.
                            • Update the sub tour operation type, $\delta_s$ of the $u^{th}$ and $u'^{th}$ sub tours.
                            • Compute: NewScore=$\delta_s$ of the $u^{th}$ sub tour + $\delta_s$ of the $u'^{th}$ sub tour.
                            • If NewScore>PrevScore, then accept the trial solution.
                      • Set u=u+1
                  UNTIL u=B.
```

Fig. 7: Procedure Eliminate-SC

```
Initialisation:   Let $\delta_s$ represents the effectiveness of the $s^{th}$ sub tour operation type. NewScore and PrevScore
                  represent the sum of $\delta_s$ for some trial solutions (x'and x'') and an incumbent solution (x), respectively.
                  Let p as an index of pipette position. Set NewScore=0 and PrevScore=0. Set the sub tour index, u=0
                  and a decision variable, NewNozzle=0. Let $Nozz_p(u)$ denotes the nozzle at the $p^{th}$ pipette for the $u^{th}$ sub
                  tour. Set x'?  x and x''?  x.

Iteration:        REPEAT
                      • If NewNozzle=0 or one of the $Nozz_p(u)$= $Nozz_p(u-1)$:
                          • Compute PrevScore=PrevScore+$\delta_s$.
                          • Create x'' from x' by swapping the nozzle positions of the pipette in the $u^{th}$ sub tour.
                          • Update the sub tour operation type, $\delta_s$ of the $u^{th}$ sub tour.
                          • Compute NewScore=NewScore+$\delta_s$.
                          • NewNozzle= NewNozzle+1;
                          • Set u=u+1 and x'?  x''.
                      • Else
                          • If NewScore> PrevScore
                            Set x?  x'.
                          • Else, x'?  x.
                          • Set NewScore=0, PrevScore=0 and NewNozzle=0.
                  UNTIL u=B.

Choose:           Return the best solution.
```

Fig. 8: Procedure Optimise-Nozzle-Pipette-Assigment

As n tries to re-optimise the assignment of nozzles to pipettes. Again, researchers use δ_s, PrevScore and NewScore to evaluate the effectiveness of the two-opt swap operation but the calculation of PrevScore and NewScore might involve many sub tours. The swap is restricted within a sub tour (Fig. 8). For example, if pipette 1 and 2 hold nozzle ID 8 and 1, respectively then pipette 1 will hold nozzle ID 1 whilst pipette 2 holds nozzle ID 8 after swapping the nozzle positions. Before accepting the trial solution, we swap several sub tours (which have at least one common nozzle/pipette assignment). Similarly, the trial solution is accepted when the NewScore is better than the PrevScore. Then the swap operations continue for the next set of sub tours until all the sub tours have been swapped. For this machine (HP-110), we try all possible nozzle/pipette assignment (in a sub tour) as the machine only has two pipettes. However, this procedure might be computationally expensive if the machine has many pipettes and with a larger dataset. Therefore, a greedy search could be applied.

RESULTS AND DISCUSSION

To demonstrate the performance of the proposed heuristics, researchers perform two experiments (Test A and B) using 30 and 300 datasets, respectively. The datasets contain 10 component types, 14 component packages, 7 nozzles in the tool bank, 2 feeder banks and various number of PCB points (from 30-900 components). The datasets are simulated data which have been designed based on the discussions with DIMA SMT Systems (i.e., the simulated data closely represent the real dataset). Actually, real world datasets are not necessarily useful for evaluating the effectiveness of the scheduling software since they have many parameters which are not useful for the scheduler. Indeed, DIMA also uses simulated datasets to test the performance of their scheduler. Unfortunately, researchers have not been able to test the solution on DIMA machine simulator as it would require some modifications to the simulator which is not possible as researchers do not have access to the source code. In this research, researchers set a user-defined tolerance as 45 mm (user-defined tolerance = nozzle gap). The specification of component packages and feeder setup are shown in Appendix A and B, respectively. In this research researchers use 7 nozzle types. These are 1, 2, 4, 8, 16, 32 and 64. However after creating a Nozzle Usage table using the Magyar *et al.* (1999) approach, researchers have eliminated the unnecessary nozzle types (i.e., 16 and 32). Therefore, the minimum nozzle change, MINCHG is 3. According to Magyar *et al.* (1999), the optimal nozzle layer should have the minimum nozzle change and minimum sub tours.

Researchers perform one run for each heuristic on each dataset. Based on the assembly Cycle Time (CT) of the schedule obtained by each approach, researchers compute the component per hour (cph) to measure the machine throughput of each solution. In this experiment, we report cph for a solution generated using Magyar approach (M_0) which has been derived by Magyar *et al.* (1999) and the approaches (that are Rd (M_1), RdSc (M_2) and RdScLs (M_3)). In order to maintain the originality of the Magyar *et al.* (1999) approach, researchers tried to re-implement their approach as closely as possible. However due to our problem constraints, researchers need to slightly modify their approach. In the Magyar's approach, we use a procedure schedule-component-pick-place-sequence without an improvement step (i.e., researchers ignore step 3 in Fig. 6 and use a Compressed Nozzle Layer produced using the procedure Determine-Nozzle-Layer. Furthermore, researchers have to ignore step 3 in Fig. 6 when reconstructing their approach. All the approaches generate solutions using a procedure schedule component-pick-place-sequence and use a Compressed Nozzle Layer produced by the procedure Determine-Nozzle-Layer-GD. The variation among our approaches: GD uses a procedure schedule-component-pick-place-sequence without an improvement step (Fig. 6), GDSc uses a procedure schedule-component-pick-place-sequence without re-optimise the assignment of nozzle to pipette (Fig. 6) whilst GDScLs uses all steps in procedure Schedule-Component-Pick-Place-Sequence. These approaches are summarised in Table 8.

The relative change in M_1 over M_0, M_2 over M_0 and M_3 over M_0 are denoted as I_1 ($I_1 = (M_1\text{-}M_0)*100/M_0$), I_2 ($I_2 = (M_2\text{-}M_0)*100/M_0$) and I_3 ($I_3 = (M_3\text{-}M_0)*100/M_0$), respectively. Results for Test A are shown in Table 9 and Fig. 9 which can be concluded as follows:

The Magyar approach only obtained 11 (i.e., 36.7%) optimum nozzle layers whilst our approaches obtained 30 (i.e., 100%). This indicates that an injection of a greedy descent search before changing the number of allowed nozzle change is capable of searching for a good nozzle layer which might improve machine throughput.

Based on cph on average GD, GDSc and GDScLs are superior to the Magyar approach in about 2.31, 2.86 and 4.3%, respectively. The best improvement (over the Magyar approach) is obtained by GDSc and GDScLs (i.e., 12.07%) whilst the worst improvement (over the Magyar

Table 8: The variation between Magyar, GD, GDSc and GDScLs methods

Methods	Stage 1: Nozzle optimisation procedure	Stage 2: Component pick-and-place optimisation procedure
Magyar	Determine-Nozzle-Layer (Magyar *et al.*, 1999)	Schedule-Component-Pick-Place-Sequence without an improvement step (ignores step 3 in Fig. 6)
GD	Determine-Nozzle-Layer-GD	Schedule-Component-Pick-Place-Sequence without an improvement step (ignores step 3 in Fig. 6)
GDSc	Determine-Nozzle-Layer-GDd	Schedule-Component-Pick-Place-Sequence without re-optimise the assignment of nozzle to pipette (ignores step 3.2 in Fig. 6)
GDScLs	Determine-Nozzle-Layer-GD	Uses all steps in procedure Schedule-Component-Pick-Place-Sequence

Table 9: Results for Test A on 30 datasets

			Magyar			GD				GDSc				GDScLs			
Test	N	MING	CHG	B	cph	CHG	B	cph	I_1(%)	CHG	B	cph	I_2(%)	CHG	B	cph	I_3 (%)
1	30	15	3	15	2984	3	15	2984	0.00	3	15	2984	0	3	15	3087	3.45
2	60	30	4	30	3006	3	30	3288	9.38	3	30	3297	9.681	3	30	3297	9.68
3	90	45	3	45	3459	3	45	3459	0.00	3	45	3482	0.665	3	45	3482	0.66
4	120	60	4	60	3397	3	60	3383	-0.41	3	60	3394	-0.088	3	60	3394	-0.09
5	150	75	4	75	3339	3	75	3280	-1.77	3	75	3305	-1.018	3	75	3305	-1.02
6	180	90	4	90	3375	3	90	3513	4.09	3	90	3525	4.444	3	90	3525	4.44
7	210	105	4	105	3427	3	105	3488	1.78	3	105	3552	3.648	3	105	3712	8.32
8	240	120	6	120	3358	3	120	3714	10.60	3	120	3741	11.406	3	120	3741	11.41
9	270	135	3	135	3347	3	135	3347	0.00	3	135	3356	0.269	3	135	3521	5.20
10	300	150	3	150	3323	3	150	3323	0.00	3	150	3351	0.843	3	150	3503	5.42
11	330	165	3	165	3501	3	165	3501	0.00	3	165	3508	0.200	3	165	3508	0.20
12	360	180	3	180	3592	3	180	3592	0.00	3	180	3593	0.028	3	180	3593	0.03
13	390	195	6	195	3240	3	195	3593	10.90	3	195	3631	12.068	3	195	3631	12.07
14	420	210	6	210	3500	3	210	3829	9.40	3	210	3840	9.714	3	210	3840	9.71
15	450	225	3	225	3372	3	225	3372	0.00	3	225	3389	0.504	3	225	3418	1.36
16	480	240	6	240	3396	3	240	3507	3.27	3	240	3535	4.093	3	240	3561	4.86
17	510	255	3	256	3461	3	255	3335	-3.64	3	255	3343	-3.409	3	255	3459	-0.06
18	540	270	3	270	3254	3	270	3254	0.00	3	270	3266	0.369	3	270	3347	2.86
19	570	285	3	287	3357	3	285	3255	-3.04	3	285	3290	-1.996	3	285	3445	2.62
20	600	300	3	300	3300	3	300	3300	0.00	3	300	3321	0.636	3	300	3457	4.76
21	630	315	4	317	3430	3	315	3479	1.43	3	315	3490	1.749	3	315	3669	6.97
22	660	330	4	330	3413	3	330	3639	6.62	3	330	3651	6.973	3	330	3651	6.97
23	690	345	3	347	3514	3	345	3670	4.44	3	345	3681	4.752	3	345	3681	4.75
24	720	360	4	360	3509	3	360	3510	0.03	3	360	3510	0.028	3	360	3510	0.03
25	750	375	3	375	3542	3	375	3542	0.00	3	375	3547	0.141	3	375	3615	2.06
26	780	390	5	390	3448	3	390	3465	0.49	3	390	3465	0.493	3	390	3465	0.49
27	810	405	3	405	3432	3	405	3432	0.00	3	405	3449	0.495	3	405	3516	2.45
28	840	420	5	420	3464	3	420	3526	1.79	3	420	3568	3.002	3	420	3568	3.00
29	870	435	6	435	3297	3	435	3620	9.80	3	435	3692	11.981	3	435	3692	11.98
30	900	450	4	450	3423	3	450	3562	4.06	3	450	3569	4.265	3	450	3569	4.27
Average improvement									2.31				2.865				4.30

N is a number of components on a PCB; Minimum nozzle changes, MINCHG = 3

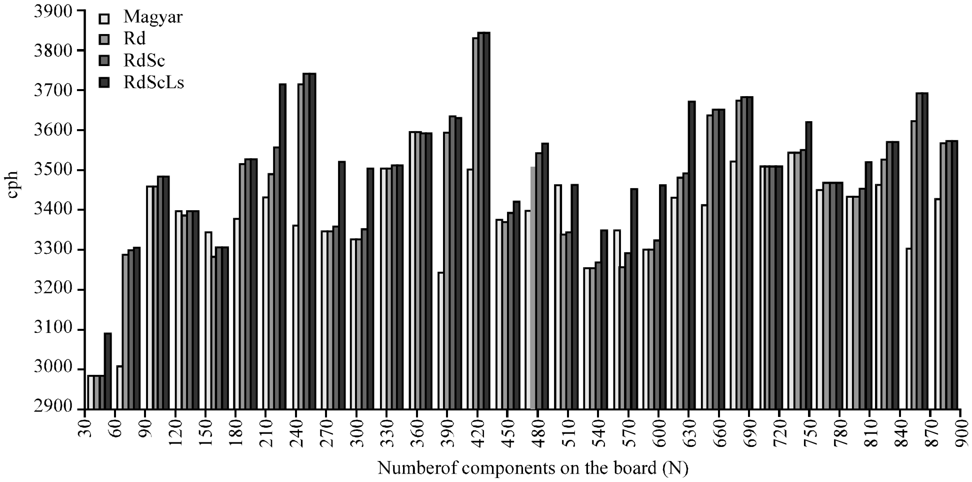

Fig. 9: A comparison of the performance of heuristics based on cph (Test A)

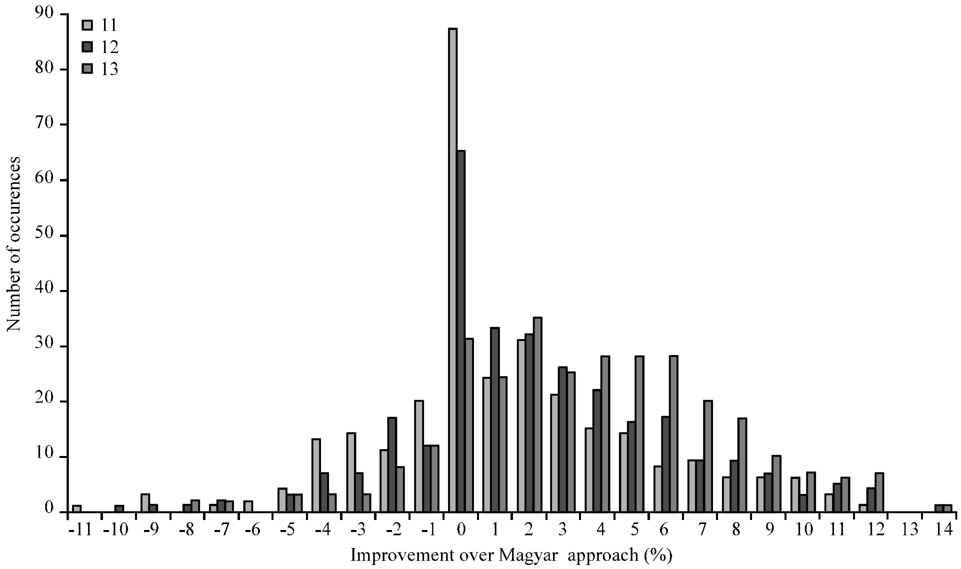

Fig. 10: A frequency distribution of I1, I2 and I3 (Test B)

approach) is -3.64% which is obtained by Rd. For example, when N = 390 (Table 9), the Magyar approach produces a solution with a minimum number of sub tours (210) and 6 nozzle changes whilst the GDSc and GDScLs both produced a solution with minimum sub tours (195) and minimum nozzle changes (3). For this case, the approaches are 12.07% superior to Magyar approach. This shows that less nozzle changes can improve the machine throughput.

In certain cases (such as for N = 150, 510 and 570), the Magyar approach outperformed the approaches, although they have not obtained the best nozzle layers for those problem instances. This might indicate that other factors which determine the quality of the sub tour such as simultaneous pickup, simultaneous vision, etc., are important for improving the machine throughput. However in general, the best nozzle layer which has the optimum number of nozzle changes and sub tours is useful in producing good component pick-and-place sequences.

To further analyse the performance of the heuristics, we perform Test B on 300 datasets (randomly generated). Figure 10 and 11 show the graphical results of the test. Both Fig. 10 and 11 show that in majority of cases, the approaches are better than Magyar (i.e., positive improvement which are the points at the right of 0 on the X-axis in Fig. 10 and the points above the X-axis in Fig. 11).

Based on the experiments on a Pentium 4, 1.5 GHz, 256 MB RAM computer, researcheres obtained a complete schedule in <0.1 sec (for all datasets). Hence, this is a good and fast heuristic.

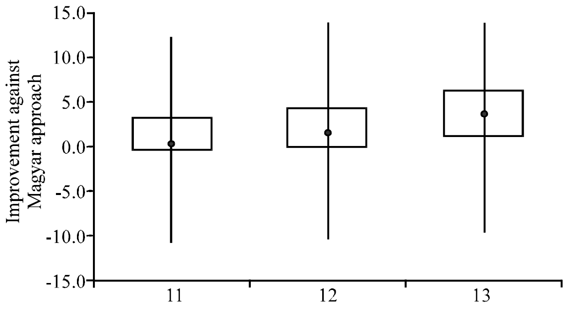

Fig. 11: Boxplot for I_1, I_2 and I_3 (Test B)

CONCLUSION

This reseach has been adapted from Magyar *et al.* (1999). In order to adhere to the constraints of the problem, researches have slightly modified the Magyar *et al.* (1999) approach (which we refer to as the Magyar approach in the rest of this study). Researchers have also proposed an improved Magyar approach that employs a greedy random descent search to generate a good nozzle layer (that has good assignment of nozzles to pipettes with good sequence of nozzle changes and sub tours) and good quality of component pick-and-place schedules for a multi-head machine (DIMA HP-110). Results show that the proposed approach is superior to the Magyar approach in about 4.3% (average result based on component per hour).

Researchers have demonstrated the importance of choosing a proper nozzle sequence in maximising the machine throughput since a nozzle change operation is time consuming. Results also show that there are some significant factors that to be considered to be able to generate good quality schedules for the hybrid pick-and-place machine are as follows (starting with the most significant):

- Minimise the nozzle changes
- Minimise number of sub tours
- Maximise the multi-pickup of mechanical aligned component (MA sub tour)
- Maximise the simultaneous pickup
- Maximise the simultaneous vision pickup and
- Maximise the same feeder bank pickup (pickup both components from the same feeder bank)

Factors (a) and (b) usually conflict with each other. Therefore as by Magyar *et al.* (1999), researchers use a weighted cost function to choose a good nozzles/pipettes assignment with good nozzle sequencing and number of sub tours. According to Magyar *et al.* (1999), optimal nozzle layer has the minimum nozzle changes with minimum number of sub tours. Since, there is a trade-off between minimising nozzle changes and the number of sub tours, a weighted cost function can choose nozzle/pipette assignments with good nozzle changes/sequencing and good number of sub tours. In some cases, researches might obtain minimal nozzle changes with a minimum number of sub tours.

This research was conducted based on our randomly generated datasets (constructed based on the problem description by DIMA machine expert). In order to test the proposed approach on the real-world machine, some modifications might be required to ensure correct communication between the scheduler and other software on the SMD placement machine. The proposed approaches in this research are focused on a constructive heuristic that is machine specific. However, a general solution framework might be applicable in solving other machine types.

ACKNOWLEDGEMENT

This research was sponsored by the FRGS/1/2012/SG05/UKM/02/11.

APPENDICES

Appendix A: The specification of component packages

Package ID	Package name	Component type	Component recognition	Nozzles
1	A	10	9	6
2	B	1	8	1
3	C	2	8	2
4	D	3	2	12
5	E	4	9	24

Appendix A: Continue

Package ID	Package name	Component type	Component recognition	Nozzles
6	F	4	2	8
7	G	5	1	3
9	I	6	2	8
10	J	7	2	4
11	K	8	1	32
12	L	8	8	64
13	M	9	11	72
14	N	10	3	64
15	O	6	2	16

Component recognition-1:SCC only; 2:LCC only; 3:SCC or LCC; 8: Mechanical only; 9: Mechanical or SCC, 10:Mechanical or LCC, 11: Mechanical, SCC or LCC where SCC, LCC and Mechanical are the component recognition/alignment methods that are using small vision camera, large vision camera and mechanical alignment on the fly, respectively. Nozzles-is a combination of Nozzle ID, i.e., the nozzles that can be used for pick-and-place the component package, the following are the examples of the Nozzles coding (using binary coding):

	T_8	T_7	T_6	T_5	T_4	T_3	T_2	T_1	T_0
Nozzles = 4	0	0	0	0	0	0	1	0	0
Nozzles = 6	0	0	0	0	0	0	1	1	0
Nozzles = 19	0	0	0	0	1	0	0	1	1

If Nozzles = 4, the nozzle ID = 4 can be used for the component package
If Nozzles = 6, the nozzle ID = 4 or 2 can be used for the component package
If Nozzles = 19, the nozzle ID = 16, 2 or 1 can be used for the component package

Appendix B: The feeder setup

Feeder bank A:								
Slot ID	0	1	2	3	4	5	6	7
Component package	A			B			C	
Slot ID	8	9	10	11	12	13	14	15
Component package			D		I	F		G
Feeder bank B:								
Slot ID	100	101	102	103	104	105	106	107
Component package	L			M	K		O	
Slot ID	108	109	110	111	112	113	114	115
Component package		J			N		E	

For example, a simultaneous pickup can happen when left and right pipettes/nozzles pickups A and B or B and C, etc

REFERENCES

Ahmadi, J., S. Grotzinger and D. Johnson, 1988. Component allocation and partitioning for a dual delivery placement machine. Oper. Res., 36: 176-191.

Ahmadi, J., S. Grotzinger and D. Johnson, 1991. Emulating concurrency in a circuit card assembly system. Int. J. Flexible Manuf. Syst., 3: 45-70.

Ayob, M. and G. Kendall, 2004. A nozzle selection heuristic to optimise the hybrid pick and place machine. Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems, Volume 2, December 1-3, 2004, Singapore, pp: 1259-1264.

Ayob, M. and G. Kendall, 2005. Component pick-and-place scheduling using a weighted nozzle rank approach. Technical Report NOTTCS-TR-2005-7, School of CSiT, The University of Nottingham, UK.

Ayob, M. and G. Kendall, 2008. A survey of surface mount device placement machine optimisation: Machine classification. Eur. J. Oper. Res., 186: 893-914.

Bentzen, B., 2000. SMD placement. The SMT in FOCUS, November 28, 2000.

Crama, Y., A.W.J. Kolen, A.G. Oerlemans and F.C.R. Spieksma, 1990. Throughput rate optimization in the automated assembly of printed circuit boards. Ann. Oper. Res., 26: 455-480.

Crama, Y., J. Van De Klundert and F.C.R. Spieksma, 2002. Production planning problems in printed circuit board assembly. Discrete Applied Math., 123: 339-361.

De Souza, R. and W. Lijun, 1995. Intelligent optimization of component onsertion in multi-head concurrent operation PCBA machines. J. Intell. Manuf., 6: 235-243.

Dima SMT Systems, 2003. Hybrid P&P HP-110, user manual version 1.2. http://www.dimasmt.nl/.

Duman, E. and I. Or, 2004. Precedence constrained TSP arising in printed circuit board assembly. Int. J. Prod. Res., 42: 67-78.

Duman, E. and I. Or, 2007. The quadratic assignment problem in the context of the printed circuit board assembly process. Comput. Oper. Res., 34: 163-179.

Ellis, K.P., F.J. Vittes and J.E. Kobza, 2001. Optimizing the performance of a surface mount placement machine. IEEE Trans. Electron. Packag. Manuf., 24: 160-170.

Garey, M.R. and D.S. Johnson, 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman, USA.

Grunow, M., H.O. Gunther, M. Schleusener and I.O. Yilmaz, 2004. Operations planning for collect-and-place machines in PCB assembly. Comput. Ind. Eng., 47: 409-429.

Ho, W. and P. Ji, 2003. Component scheduling for chip shooter machines: A hybrid genetic algorithm approach. Comput. Oper. Res., 30: 2175-2189.

Ho, W. and P. Ji, 2004. A hybrid genetic algorithm for component sequencing and feeder arrangement. J. Intell. Manuf., 15: 307-315.

Ho, W. and P. Ji, 2006. A genetic algorithm approach to optimising component placement and retrieval sequence for chip shooter machines. Int. J. Adv. Manuf. Technol., 28: 556-560.

Jeevan, K., A. Parthiban, K.N. Seetharamu, I.A. Azid and G.A. Quadir, 2002. Optimization of PCB component placement using genetic algorithms. J. Electron. Manuf., 11: 69-79.

Ji, P. and Y.F. Wan, 2001. Planning for printed circuit board assembly: The state-of-art review. Int. J. Comput. Appli. Technol., 14: 136-144.

Kumar, R. and Z. Luo, 2003. Optimizing the operation sequence of a chip placement machine using TSP model. IEEE Trans. Electron. Packag. Manuf., 26: 14-21.

Leipala, T. and O. Nevalainen, 1989. Optimization of the movements of a component placement machine. Eur. J. Oper. Res., 38: 167-177.

Leu, M.C., H. Wong and Z. Ji, 1993. Planning of component placement/insertion sequence and feeder setup in PCB assembly using genetic algorithm. J. Electron. Packag., 115: 424-432.

Magyar, G., M. Johnsson and O. Nevalainen, 1999. On solving single machine optimization problems in electronics assembly. J. Electron. Manuf., 9: 249-267.

McGinnis, L.F., J.C. Ammons, M. Carlyle, L. Cranmer and G.W. Depuy *et al.*, 1992. Automated process planning for printed circuit card assembly. IIE Trans., 24: 18-30.

Moyer, L.K. and S.M. Gupta, 1996a. SMT feeder slot assignment for predetermined component placement paths. J. Electron. Manuf., 6: 173-192.

Moyer, L.K. and S.M. Gupta, 1996b. Simultaneous component sequencing and feeder assignment for high speed chip shooter machines. J. Electron. Manuf., 6: 271-305.

Moyer, L.K. and S.M. Gupta, 1997. An efficient assembly sequencing heuristic for printed circuit board configurations. J. Electron. Manuf., 7: 143-160.

Safai, F., 1996. Cycle time improvement for Fuji IP2 pick-and-place machines. Hewlett-Packard J., 47: 80-86.

Sanchez, J.M. and J.W. Priest, 1991. Optimal component-insertion sequenceplanning methodology for the semiautomatic assembly of printed circuit boards J. Intell. Manuf., 2: 177-187.

Sastry, K., D. Goldberg and G. Kendall, 2005. Genetic Algorithms. In: Search Methodologies: Introductory Tutorials in Optimization and Decision Support Methodologies, Burke E.K. and G. Kendall (Eds.). Springer, USA., pp: 97-125.

Shih, W., K. Srihari and J. Adriance, 1996. Expert system based placement sequence identification for surface mount PCB assembly. Int. J. Adv. Manuf. Technol., 11: 413-424.

Tirpak, T.M., P.C. Nelson and A.J. Aswani, 2000. Optimization of revolver head SMT machines using adaptive simulated annealing (ASA). Proceedings of the 26th IEEE/CPMT International Electronics Manufacturing Technology Symposium, October 2-3, 2000, Santa Clara, CA., pp: 214-220.