
Paper:

Andrzejewski, K., Cooper, M., Griffiths, C. & Giannetti, C. (2018). Optimisation process for robotic assembly of electronic components. *The International Journal of Advanced Manufacturing Technology*
http://dx.doi.org/10.1007/s00170-018-2645-y

The International Journal of Advanced Manufacturing Technology
https://doi.org/10.1007/s00170-018-2645-y

ORIGINAL ARTICLE

Optimisation process for robotic assembly of electronic components

K. T. Andrzejewski[1] · M. P. Cooper[1] · C. A. Griffiths[1] · C. Giannetti[1]

Received: 21 February 2018 / Accepted: 26 August 2018

Abstract

Adoption of robots in the manufacturing environment is a way to improve productivity, and the assembly of electronic components has benefited from the adoption of highly dedicated automation equipment. Traditionally, articulated 6-axis robots have not been used in electronic surface mount assembly. However, the need for more flexible production systems that can be used for low to medium production builds means that these robots can be used due to their high degrees of flexibility, excellent repeatability and increasingly lower investment costs. This research investigated the application of an articulated robot with six degrees of freedom to assemble a multi-component printed circuit board (PCB) for an electronic product. A heuristic methodology using a genetic algorithm was used to plan the optimal sequence and identify the best location of the parts to the assembly positions on the PCB. Using the optimised paths, a condition monitoring method for cycle time evaluation was conducted using a KUKA KR16 assembly cell together with four different robot path motions. The genetic algorithm approach together with different assembly position iterations identified an optimisation method for improved production throughput using a non-traditional but highly flexible assembly method. The application of optimised articulated robots for PCB assembly can bridge the gap between manual assembly and the high-throughput automation equipment.

Keywords Sequencing optimisation · Electronics assembly · KUKA robotics · Flexible manufacture · Genetic algorithm

Notations

PTP	Point to point
SPTP	Spline point to point
LIN	Linear
SLIN	Spline linear
GA	Genetic algorithm
SCARA	Selective compliance assembly robot arm
IC	Integrated circuit
SIL	Single in line
PCB	Printed circuit board
DOF	Degree of freedom
KRL	KUKA Robot Language
α (alpha)	Degree of rotation
d	Distance
h	Height

1 Introduction

Electronics manufacturing has evolved over the past years from a labour-intensive activity to a highly automated one. In printed circuit board (PCB) assembly, the electronic components need to be placed into position prior to soldering, and the use of automation provides accuracy, repeatability and efficiency to this process, when compared to manual assembly. Furthermore, the competition faced by electronic component manufacturers causes a need for high-throughput rates for which automated assembly lines are a major asset [1]. Implementation of robotics in assembly of electronics offers some distinct advantages over manual methods due to its reliability and flexibility. Assembly of electronic components is generally limited to two-dimensional horizontal x- and y-axis movements and usually comprise of several sub-systems, including part feeding systems, work holding and pick and place devices. Depending on the application, various types of robots can be used for pick and place operations, e.g., gantry/ Cartesian robot, cylindrical robot, spherical robot, SCARA (Selective Compliance Assembly Robot Arm) and spider robotic arms [2, 3]. Despite its popularity, the articulated robot

---

✉ C. A. Griffiths
c.a.griffiths@swansea.ac.uk

[1] College of Engineering, Swansea University, Swansea, UK

type has been widely neglected for printed circuit board (PCB) assembly due to its relatively higher equipment prices. However, articulated robots with a much larger work envelope when compared to Cartesian and SCARA have the potential to become more popular in the electronics industry due to high degrees of flexibility, excellent repeatability and now also due to lower investment cost and competitive prices. Understanding the costs and benefits of multiple degrees of freedom, especially those that create redundancy, is a fundamental problem in the field of robotics [4]. Redundancy is when the robot has more degrees of freedom than needed for performing the tasks [5]; therefore, a modern automation cell incorporating an articulated 6-axis robot needs has the freedom to not only execute traditional automation activities but also perform specialist assembly operations. In addition to improved redundancy and space control, current technology has resulted in articulated robots being able to work in close proximity to human workers. Collaborative robots allow the possibility for assembly lines with close contact between human and robot, this modern manufacturing development allows for the high performance of robots together with the unique capabilities of people.

The focus of this work is on low-volume assembly of electronic parts. Low-volume production challenges are important as new industrial strategies for high-value twenty-first century products need to consider flexible manufacturing methods. The test part in this research includes a front panel and two printed circuit boards connected through single in-line (SIL) headers. Each of the PCBs consists of numerous parts, such as switches, jack ports, potentiometers and electronic components; and resistors, capacitors, transistors and fuses. This product lends itself to mass production systems that traditionally would have used production lines with either high manual labour costs or high investment specialised automation. However, in this automation research, a non-specialised system is used. In this way, the research will investigate the application of a robotic system that can be flexible in producing multiple part and build options, while meeting the precision, efficiency, reliability and repeatability of a dedicated automation system.

The finished product consists of multiple assembly operations with several possible build routes; consequently, the main contribution of this paper is the application of a heuristic methodology for planning the assembly operation using an articulated KUKA robot with six degrees of freedom in a pick-and-place operation. The introduction of robots into industry seeks to upgrade not only the standards of quality but also productivity, as working time is increased and idle or wasted time is reduced [6]. For this reason, path planning and the improvement of the autonomy is a fundamental issue in robotics [6, 7]. The common criterion for optimisation is make-span minimisation or, in the context of repetitive assembly, cycle time minimisation [1, 8–11]. The preliminary study prior to automation included a manual assembly of the product to identify fundamental issues involved in the process. This approach was necessary to provide a description of the generic steps involved in the assembly and to translate those into the automated environment. It has been concluded that in principle, infinite joint configurations may result in executing the same task [5] and that the production planning process optimisation must determine the following sub-problems:

- The correct allocation of component feeders
- Identify the optimum component placement sequence
- The optimum orientation and work space arrangement

The paper is structured as follows. First, there will be a review of the feasibility of optimising the process using a genetic algorithm. This is followed by a description of the experimental setup including the test part, robot and the workspace arrangement. Then, the methodology adopted is provided with a problem representation and description of the GA used for the feeder assignment and placement sequence. A results section discusses the performance of the GA and the solutions for optimal motion type, orientation of work piece and the objective function for the feeder allocation and location of components. Finally, conclusions are made on the proposed system.

2 Feasibility of the genetic algorithm in process optimisation

The genetic algorithm technique dates back to 1975 and was first introduced by Holland [12]. In the literature, the genetic algorithm has been widely used to solve optimisation problems in many industrial applications such as job shop sequencing and scheduling [13], assembly planning [14], selection of machining parameters in turning operation [15], design of sheet-metal assembly and machining fixtures [16] and tolerance design [17]. The processes of optimisation for the feeder-slot allocation and component sequencing are the two most important factors for improving the efficiency of this assembly operation. In the literature, some of research focused on only one of the two problems, while assuming the solution to the other, is given in advance. Ball and Magazine [18] assumed that the feeder arrangement was fixed and solved the placement sequencing problem. Their solution to the problem was modelled as a rural postman problem with a heuristic approach used to solve it. Ahmadi and Mamer [19] modelled the problem of sequencing the component types for placement and the problem of scheduling the movements between points on the PCB as a collection of interdependent traveling salesman problems. Klomp et al. [20] solved the component allocation problem for a turret type machine. Li et al. [21] considered a pick-and-place machine with a revolving head. The approach to the placement sequencing was modelled as a travelling salesman problem. Then, in the second stage, they employed a genetic algorithm to solve the feeder arrangement problem. Other research work has focused on tackling both problems (the feeder arrangement problem and the placement sequencing problem) by iterating their solutions. Grunow et al. [22] considered a pick-and-place manipulator with a revolving head and solved both problems iteratively. They used simple heuristics to obtain the initial feeder arrangement and then solved the placement sequence as a simple vehicle routing problem.

Finally, many researchers proposed heuristic or metaheuristic methods for solving the feeder arrangement and placement sequencing problems simultaneously. Sohn and Park [23], which considered a turret-type machine, adopted an integrated approach in their solution. They solved the feeder arrangement and placement sequencing problems integrally for a sequential pick-and-place machine. A simple pairwise exchange heuristic was used to solve the feeder arrangement, while the evaluation of each feeder arrangement solution was performed by finding the placement sequence using a modified farthest insertion heuristic. An integrated Integer Programming method was established by Broad et al. [24]. Their model was solved by a binary integer programming package. Deo et al. [25] proposed a genetic algorithm for a complex problem of component sequence optimisation in multiple setups, which were necessary with limited feeder holding capacity. Ho and Ji [26] hybridised the genetic algorithm with the nearest neighbour heuristic, the 2-opt heuristic and an iterated swap procedure. Ellis et al. [27] developed a heuristic for solving the feeder arrangement and the placement sequencing problems simultaneously for a turret-type machine. A construction procedure was used with a set of rules to generate an initial placement sequence and feeder allocation. Their model also contained an improvement procedure to the initial solution. Magyar et al. [28] proposed a hierarchical solution approach to solve the problem of determining the placement sequence and feeder arrangement. Sun et al. [29] studied a dual-gantry collect-and-place machine. They used a genetic algorithm to decide the component allocation between the two revolving. They proposed a greedy heuristic for the work cycle formation and pickup sequencing decisions achieved by maximising the number of simultaneously picked up components for each access of a multi-head module, or equivalently minimising the number of pickups, and balancing the workload between the two gantries. Kulak et al. [30] proposed genetic algorithms for both single-gantry and dual-gantry collect-and-place machines with revolver-type placement heads. The feeder arrangement problem and the placement sequencing problem were solved by a novel genetic algorithm approach, which integrates a clustering algorithm for generating sub-sections of the PCB and grouping the corresponding placement operations. Garcia-Naijera and Brizuela [31] proposed an efficient genetic algorithm to solve the problem of component sequencing with feeder arrangement. Their computational experiments show that the algorithm they developed improves the state-of-the-art result on a benchmark for the problem.

It can be concluded from the above survey that research has been conducted to investigate printed circuit board assembly optimisation problems. However, no GA optimisation research has been performed using a single 6-axis robot, and no theoretical results have been validated in a real-life setting. The need for this is clear as the 6-axis robot allows for a whole production cell that allows for the widest possible autonomy and manufacturing capability. This non-traditional approach is of particular importance when considering low-volume production and prototype builds with a wide range a product variation.

3 Experimental setup

3.1 KUKA robot

The study was conducted using a KUKA KR16 6-axis industrial robot. The KUKA industrial robots are highly accurate with a position repeatability of ± 0.04 mm [31]. This is a precision suitable for the test part assembly. Once the sequence and feeder slot designation had been produced from the genetic algorithm, the path had to be programmed into the KUKA robot. The locations of all the feeders and component positions were saved as way points using the teach method and within the sequence, timers commands were added to start recording at the first point and stop recording at the end. The KUKA robot offers six different types of path motion. Two of these will be disregarded as they are circular motions, which are not relevant to the proposed problem. The motion types available for the electronics assembly are as follows:

- Point to point (PTP)—This motion type involves following the quickest path between two points and is not necessarily a straight path.
- Linear (LIN)—The linear motion type follows a straight path and uses more joints in constant motion to trace the straight path.
- Spline point to point (SPTP)—This is similar to the PTP motion; however, it allows for continuous spline motions where points are estimated and a smoother motion is available.
- Spline linear (SLIN)—As with the SPTP, this motion type uses splines between linear motions.

The effects on the cycle time for each motion type were tested using the same sequence and feeder arrangement. By making this comparison, a solution can be obtained that not only reduces the cycle time through good placement sequencing, but also through direct optimisation of movement using the best motion type.

3.2 Demonstrator test part

The Eurorack Serge filter was chosen as the focus of the study due to its suitable complexity and variety of components. The assembly operation of the printed circuit board can be achieved using a variety of placement technologies; however, the main focus of this paper is to present the procedure of finding the best solution for a 6-axis robot. The Eurorack Serge filter consists of two PCBs (Fig. 1) with 100 components between them that have to be placed and soldered. The first PCB consists mostly of resistors, capacitors and diodes, while the second has larger components such as jack ports and potentiometers. The components vary greatly in size and shape and thus cannot be placed with the same end effector. Twelve component feeders are required, and the product provides a good representation of the PCBs that would be prototyped or produced in small batches.

3.3 Workspace arrangement

The PCBs were positioned on a 58 cm × 58 cm worktable at a height of 80 cm; using the centre of the table as a body frame, the initial distance from the robot axis 1 global frame was 100 cm. With a fixed position for the component feeders, the PCBs are arranged with the long and short edges adjacent to the feeders at an initial distance of 10 cm (Fig. 2). The two arrangements are expected to produce different results due to the total distance between the feeder of a particular component type and the designated place on the board being different for both cases. By means of a simple experiment in which the proposed genetic algorithm is run with both orientations, the best option can be chosen based on the fastest solution. The arrangement of the entire work piece (feeders and PCB) in relation to the base of the KUKA robot can also be optimised. By varying the angle (alpha) (α) by which the PCB and feeders are arranged to the KUKA robot, the optimal orientation can be determined. Similarly, the proximity of the work piece to the base also has an effect which can be determined. Both considerations were investigated using an experimental method of trialing various positions using the KUKA robot. The angle problem was also investigated using a computational method to relate the angle of rotation to the total distance travelled in a cycle.

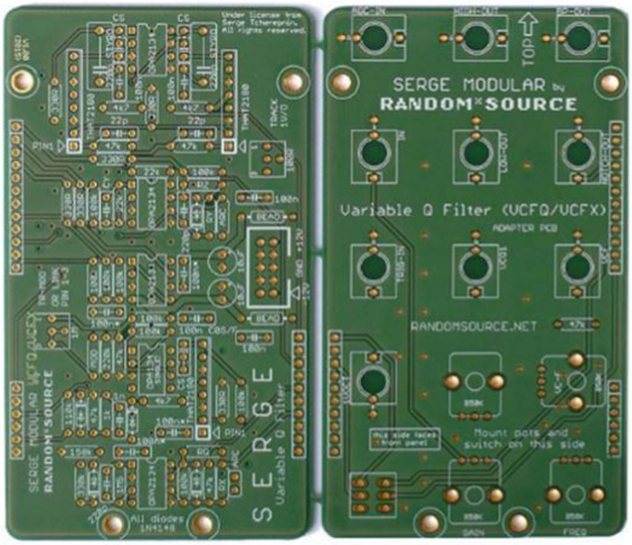

Fig. 1 Eurorack Serge filter PCBs

4 Methodology

4.1 Problem representation

In order to design an efficient assembly system for a particular printed circuit board, two distinct but related problems need to be solved. First is the reduced distance allocation of the component types to the feeder positions. The distance between the individual component pick up location and its destination on the PCB depends on the feeder rack arrangement, the distance between the feeder and PCB and the orientation of the circuit boards on the holding table [24]. The PCB assembly problem consists of placing a number of electronic components of prespecified types at set locations on a PCB. The core decision variables in the feeder allocation problem refer to the allocation of individual component feeders to positions in the magazine [31]. The goal is to minimise the total distance and thus reduce the corresponding cycle time. The next problem is to determine the sequence in which the components are placed into the boards. The assembly operation works as follows: The end effector traverses to the feeder rack, from which it picks up a component and travels to place the component on the PCB. After completing the placement tour, the steps are repeated for a new placement tour [18]. There are, however, some major assumptions and operation specifications of the assembly system, upon which this investigation is based:

- Each component type is setup only once in the feeder magazine for simplification, as this approach eliminates a component retrieval problem
- The robotic arm is equipped with the set of end effectors needed to pick up all types of components one component at a time.
- Each component type can be picked up with a subset of tools, that is, one head with a specific tool can only pickup components from a limited set of component types [20]. It has been concluded that five sub-groups made out of

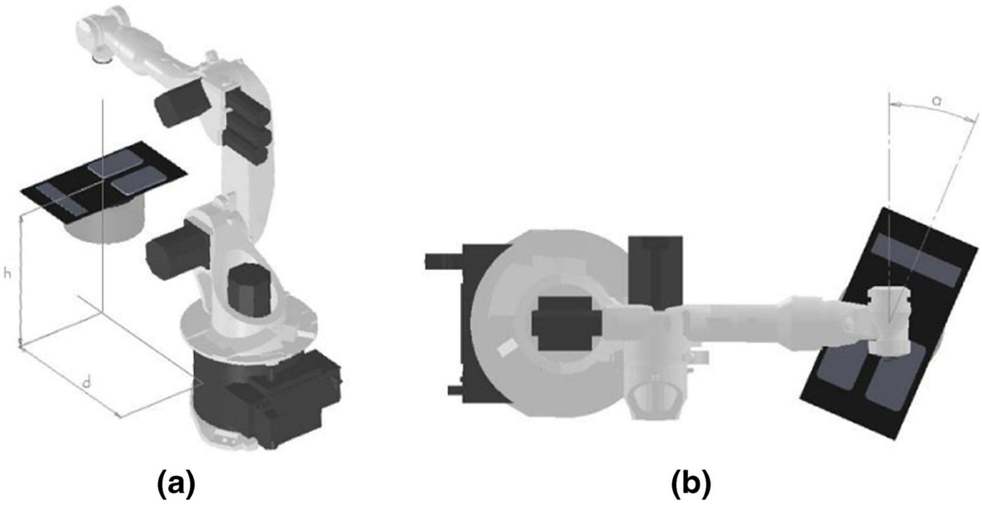

Fig. 2 (**a**) Orientation of the work piece in 3D space in relation to the robot. (**b**) Orientation of assembly process when rotated by an angle of α (alpha)

similar design features of the component types can be distinguished.

- The assembly operation starts with the smallest components due to basic assembly rules established in the manual build.

4.2 The genetic algorithm

Due to the combinatorial nature of the problem studied in this paper, an approach based on the genetic algorithm was developed to solve both the feeder slot assignment and component sequencing problem. In this method, a candidate solution is represented as an individual with a set of properties called chromosomes, and the group of individuals is called a population. The population evolves to the next generation through crossover and mutation with the two-point crossover method adopted, the mutation operates on the offspring created in the crossover step based on set probability [32]. After the reproductive operations, the fitness of the offspring is assessed and compared against the parents. Based upon survival of the fittest rule, the best individuals of the offspring are selected for the next generation. The genetic algorithm runs until termination criteria are satisfied or no improvement is observed during a number of generations. It is the associated cost that determines the fitness of the individual in population. The lower the cost, the higher the fitness is. In the optimisation of the robotic assembly, the cost is often cycle time [1, 8–11] and the associated total distance the robot manipulator travels during an assembly operation [25, 33]. In the model adopted, the best individuals are chosen to minimise the associated cost and provide the shortest path between two points was employed. The cost is evaluated as the sum of all distances between the feeder and component locations on the PCB for the feeder slot assignment problem. In the placement sequence problem, the objective function is a sum of all the distances between each component in the sequence and the feeder location of the next component in the sequence. The objective function for the first problem can be expressed as follows:

$$Min\sum_{i=1}^{n}\sum_{j=1}^{m}\sqrt{\left(x_i-x_j\right)^2+\left(y_i-y_j\right)^2+\left(z_i-z_j\right)^2} \tag{1}$$

where

n	length of individual (number of feeder slots in feeder rack)
m	total number of all components
x_i, y_i, z_i	Cartesian coordinates of the position of *i*th feeder
x_j, y_j, z_j	Cartesian coordinates of the destination of *j*th component

The objective function for the placement sequence problem can be expressed as:

$$Min\left(d_{f,c}+d_{c,f+1}\right) \tag{2}$$

$$d_{f,c}=\sum_{i=1}^{n}\sum_{j=1}^{m}\sqrt{\left(x_i-x_j\right)^2+\left(y_i-y_j\right)^2+\left(z_i-z_j\right)^2} \tag{3}$$

$$d_{c,f}=\sum_{i=1}^{n}\sum_{j=1}^{m}\sqrt{\left(x_i-x_j\right)^2+\left(y_i-y_j\right)^2+\left(z_i-z_j\right)^2} \tag{4}$$

where

n	vector of integers describing sequence of component placement destinations
m	vector of integers describing consequent feeders from which component are picked
x_i, y_i, z_i	Cartesian coordinates of the position of *i*th feeder

x_j, y_j, z_j	Cartesian coordinates of the destination of *j*th component
$d_{f,c}$	sum of distances describing feeder to component path
$d_{c,f}$	sum of distances describing component to feeder path

The developed genetic algorithm (Fig. 3) takes all the inputs from the coordinates of all components and feeders. The algorithm calculation returns the optimal feeder and sequence solutions, and the output is then optimised in accordance with the pre-set performance parameters. Those include initial population size, number of generations, crossover and mutation probabilities, and the type of the objective function (minimum distance the end-effector travels) corresponding to it total assembly time.

4.3 Feeder slot assignment

The core variable in the assembly optimisation refers to the arrangement of feeder slot locations to the component types. As mentioned, the goal is to find suitable allocation of the component types to feeder slots which minimises the total distance the pick and place manipulator travels and thus the corresponding cycle time [28]. The algorithm analyses the neighbourhood relations between the different types of components and the corresponding placement locations on the boards. The idea behind this heuristic approach is to arrange component types, which are characterised by strong neighbourhood relations, adjacent to each other in the component magazine [22]. In the example device, 12 component types were distinguished. Therefore, each gene within the chromosome represents a number corresponding to the feeder for a specific set of components, and the order of numbers within the chromosome corresponds to the feeder location arrangement.

4.4 Placement sequence

After the feeder arrangement has been determined, the next step is to find the sequence of placing the individual components on the circuit boards. The objective again is to minimise the total distance travelled by the placement tool. For this particular problem, each gene was assigned a component ID number. To ease the computational time, several constraints and assumptions were employed. Each component ID was assigned with a tool number, which corresponds to the end effector suitable for the pick-and-place operation. End effector changeover time negatively affects cycle time; therefore, to improve productivity, it is essential to minimise tool changeover for different components. To facilitate this constraint, the algorithm groups the components based on their tool number and searches for optimal sequencing solutions within each group. As the total number of components to be placed on a board is 100, there are actually 105 genes in each chromosome, which drives the number of possible combinations to an extremely large number. The first 100 numbers correspond to the component IDs, with the last 5 corresponding to the tool

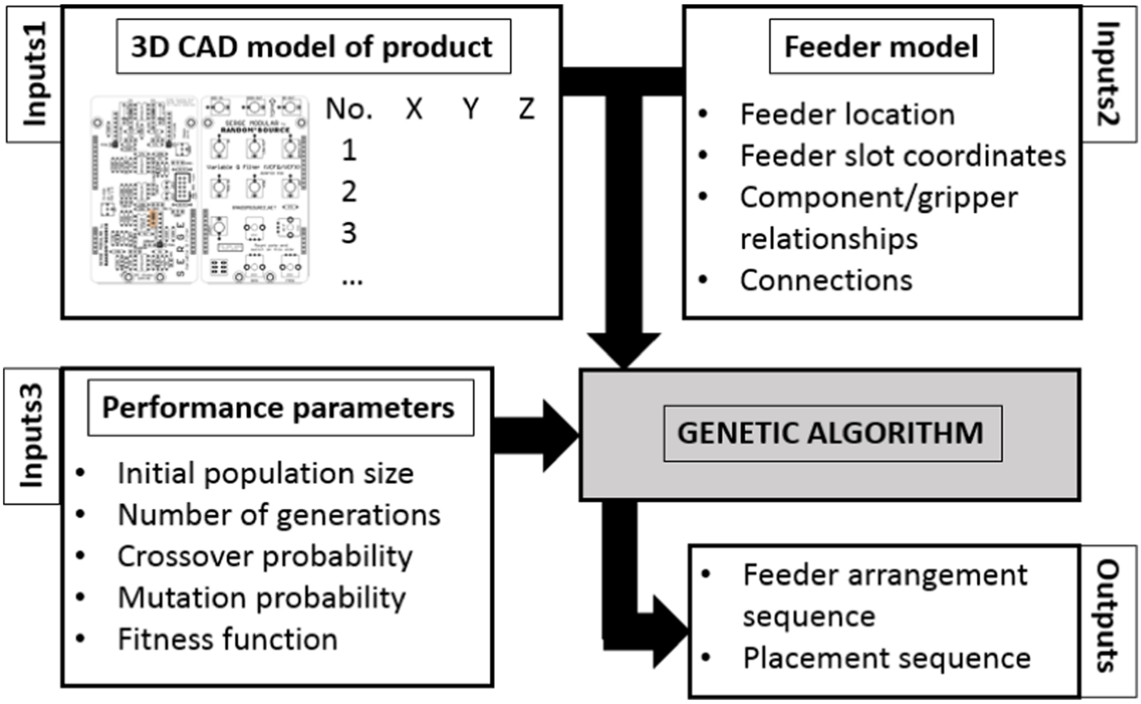

Fig. 3 Genetic algorithm inputs and outputs

sequence. Once the solution is obtained, the heuristic calculates the neighbourhood relationship between marginal components. The suitable tool sequence is defined based on minimising the distances between the last component in each sub-group and location of the magazine from which the first component in the next tool-group is being collected. Figure 4 illustrates the chromosome structure based on the cluster approach described above.

5 Results

5.1 Genetic algorithm reliability

The genetic algorithm has proven to be a reliable tool in finding good quality solutions to two combinatorial problems. The feeder slot assignment problem has been solved, and for the quickest solution, the feeder slots are positioned well to avoid unnecessary travel. The placement order of the components has also been optimised to reduce the total path distance travelled. The genetic algorithm has been used to find the shortest and conversely longest paths. These represent the “best” and “worst” solutions. Between these solutions, there is over a 2-s improvement per cycle for the SLIN motion type, which is commonly used in this sort of operation (Table 1).

Table 1 Comparison of experimentally recorded times for best and worst solutions for all motion types

Motion type	LIN	SPTP	SLIN	PTP
Best cycle time (ms)	107,796	94,284	93,264	56,340
Worst cycle time (ms)	119,604	96,936	95,460	60,108
Difference	11,808	2652	2196	3768

5.1.1 GA results—feeder slot assignment

The results obtained using the genetic algorithm code for the feeder slot assignment problem showed that the initial population indicates the group of the six fittest individuals generated through random permutations. A relatively high distance value was observed. The “distance” gives indication of the value produced by the cost function, and its reciprocal is presented as “fitness.” With each evolution of the population, new feeder slot arrangements of the higher value of fitness are found. The 14th population yielded the chromosome with the highest fitness value and therefore the suitable solution to the feeder allocation problem. Repeatability analysis of the results obtained for the feeder slot arrangement problem is presented in Fig. 5. The upper graph shows the collective result of the evolution of the population shown as fitness against generations. The performance analysis was based on data collected from a batch of 100 runs of the genetic algorithm with population size 200, crossover probability 0.8 and mutation probability 0.6. The algorithm has shown convergence toward the optimal solution with a probability of 80%, and the standard deviation calculated for results of total distance travelled was 0.01425 m which accounts for 0.16% of the total distance.

Parameter sensitivity analysis was carried out on the programmed genetic algorithm using the placement sequence problem. The initial population size for this analysis was set to 200, and the number of generations/iterations was set to 1000 to ensure quality of the results. Figure 6 presents the findings of the crossover probability investigation. The parent solution with high fitness is crossed over more often which increases the algorithms ability to explore a larger space of solutions, and large space search reduces chances of setting for local optimum. It can be concluded that a higher crossover probability improves the quality of the results and produces a better solution. The mutation probability plot, presented in Fig. 7, also shows benefits of applying higher rates. The plot indicates that a search with lower mutation rates, of the data for mutation probability of 0.2 and 0.4, weakens the useful perturbation among the potentially promising genes. In a result, many solutions are never tried out. However, setting the mutation rate too high increases the risk of offspring losing resemblance to their parent chromosomes, resulting in rapid convergence and significant loss of the algorithm’s ability to learn from the history search (data points for mutation probability of 0.6 and 0.8).

On the basis of the examinations that were performed on the initial population size, it can be concluded that for the feeder slot assignment, the genetic algorithm remains unresponsive to increase in the population size. However, for the placement sequencing problem, the initial population size was proven to be a significant factor in finding a quality solution. In Fig. 8, it can be seen that the fitness of the population drops rapidly at the first 100 iterations. When the population size is small (100), the algorithm finds poor quality chromosomes during the later stage. On the other hand, when population size is set to 200, the algorithm generates good offspring

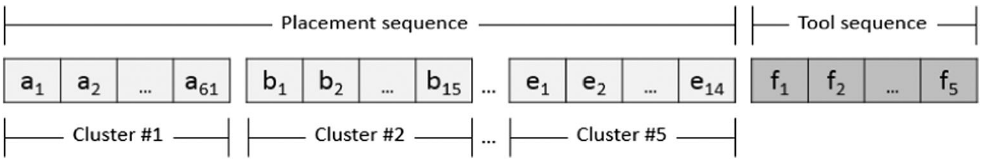

Fig. 4 Placement sequence problem—chromosome structure based on cluster approach

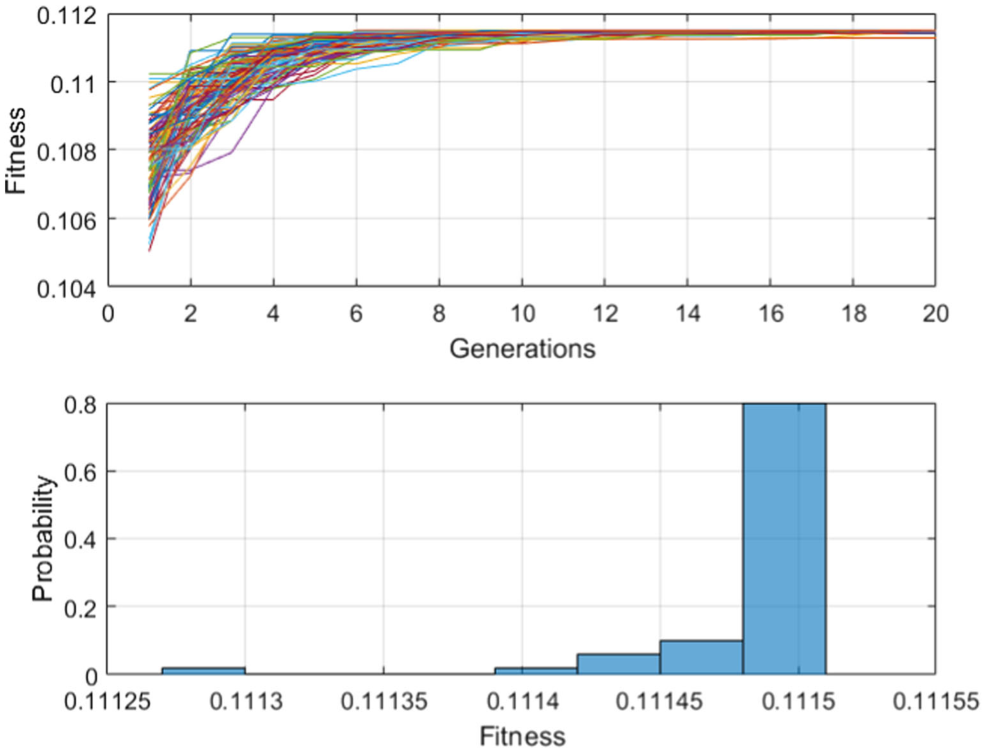

Fig. 5 Repeatability analysis: fitness vs generations plot (top); probability vs fitness histogram (bottom)

quickly from comparably highly fit parents. When the algorithm finds the best solution, the curve reaches the plateau, indicated by a horizontal stretch in a learning curve, as the best solution is already the optimal one.

5.2 Optimal motion type

Using the solutions created with the genetic algorithm, a time difference between the best and worst has been found. Depending on the motion type (LIN, PTP, SLIN, SPTP), the improvement has been found as between 2 and 11 s (Table 1). The motion type has a large effect on the speed of the path followed. As can be seen from the results, the PTP motion type is far quicker than the others; however, while slowest, the LIN motion type offers the best improvement between the best and worst. This is however negated by the quicker motion types. There are drawbacks to the faster motion types; at faster speeds (particularly PTP motion), vibrations occur in the robot structure. This could cause wear in the robot and reduce repetition accuracy within the process. The nature of the assembly process requires high levels of control and accuracy, and it

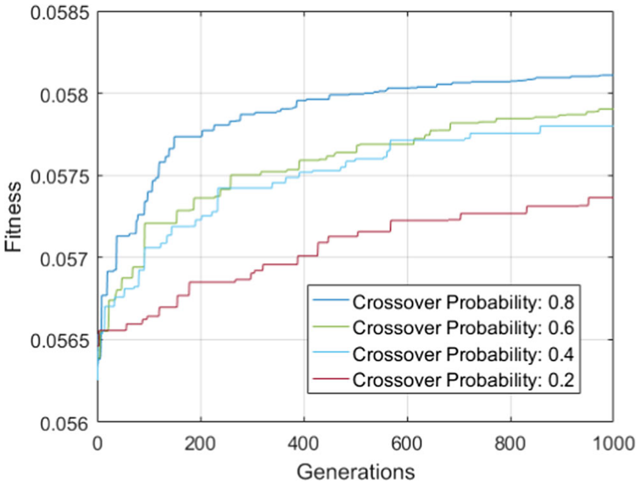

Fig. 6 Parameter sensitivity: effect of crossover probability on algorithm performance

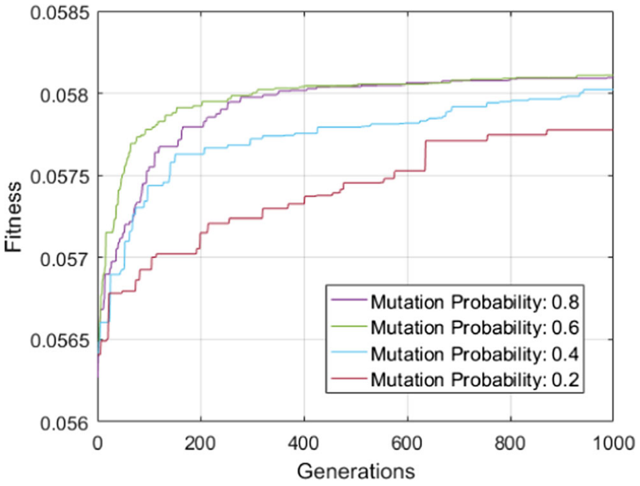

Fig. 7 Parameter sensitivity: effect of mutation probability on algorithm performance

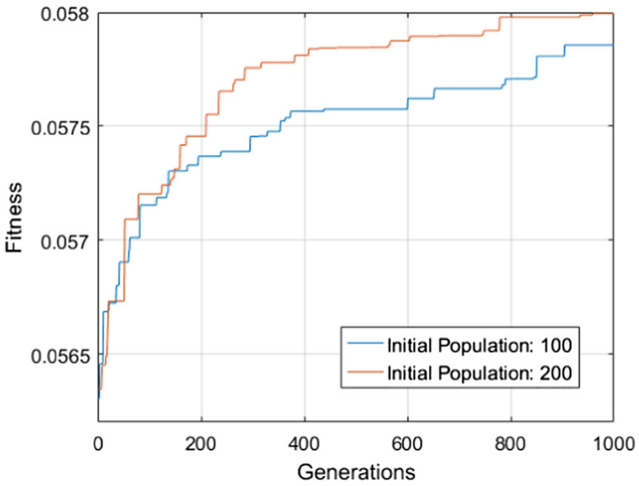

Fig. 8 Parameter sensitivity: effect of population size on solution quality

would be potentially damaging to the robot to run at full speed with PTP motion. Because of this, the PTP motion type was retested at lower speeds to reduce vibration, and, at 50% velocity, the time was 79,956 ms, which is still quicker than the next best motion type (SLIN). The SPTP motion type performed worse than expected, and it is theorised that this is due to the increasingly complex calculations performed by the robot and a notable “thinking” time was observed between movements.

5.3 Orientation of work piece

During the experiments, it was found that some paths, while longer, showed a reduced assembly time. Upon further investigation, it was found that cycle time is not exclusively affected by path length. The orientation of the work piece to the KUKA co-ordinate system affected the time it takes to perform a motion between two points. This was investigated for angles between 0° and 90°, as other angles could simply be inferred from these results. By rotating the angle at which the feeder and PCBs are arranged within the workspace, the X and Y components of motion can be reduced. Between two points, the distance travelled is the same; however, it takes less time to perform (Fig. 9). In the figure, it can be seen that by rotating the entire process, the Y component of motion in particular is reduced and, for most components, this translates to a shorter projection and faster movement time. Variation in assembly time was seen for angles between 0° and 90° rotations. Angles closer to 45° showed improvement in time which is due to this being the shortest projection (Table 2) (Fig. 10). These results were found by assigning a midpoint around which the whole layout was rotated. It can be seen that the numerical results of projected distance against angle results closely match the experimentally tested time against angle results. This shows that by shortening the projected distance, the time increases, which confirms the beneficial effect of rotating the assembly process by an angle of α close to 45°.

Further to this, the distance (*d*) from the KUKA base at which the assembly process was performed was investigated as this influenced the assembly cycle time. The entire assembly operation was translated to various locations within the workspace. The closer to the base the operation was performed, the faster the assembly time (Table 3). This is likely due to a reduction of moment within the robot arm allowing for faster acceleration and deceleration and a higher travel speed. Therefore, the optimised operation should be performed as close to the base as possible in the *d* direction,

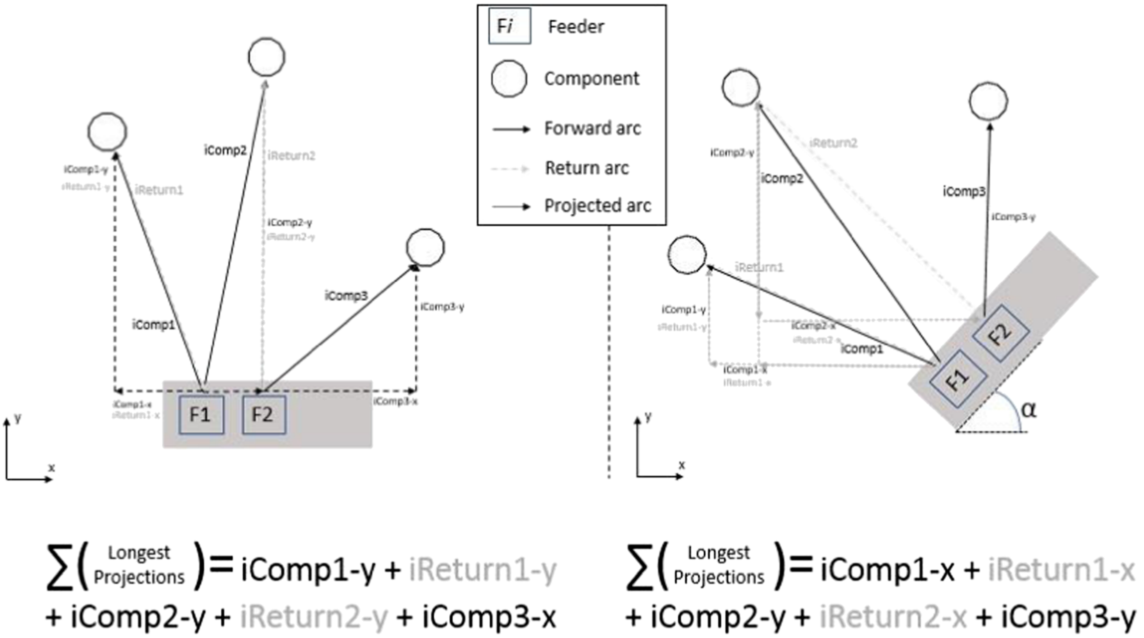

Fig. 9 Components of motion between feeders and component placement points

Table 2 Experimentally recorded angle of rotation and cycle time

Angle (°)	Time (ms)
0	92,448
10	92,520
20	92,004
30	90,936
40	89,724
50	89,280
60	89,988
70	91,404
80	92,436
90	93,072

without reaching the axis limits (Fig. 2). The height (*h*) at which the assembly is performed showed slight variation but not enough to conclude that anything other than the available work table should be used.

5.4 Objective function for feeder allocation and location of components

During the experiments, it was found that some components were less affected by the feeder assignment. This was due to an unanticipated but logical phenomena occurring. It was found that the time it takes for a diagonal line to be traced by the robot is dictated by the longest X or Y component of the motion, as both happen simultaneously. Therefore, changing the objective function to minimise projection rather than linear distance improves the solution and speeds up the process by shortening the dictating X or Y component. This led to the discovery that certain areas of the PCB are not affected by feeder placement, as the distance from the feeder is greater than the gap between feeders 1 and 12. This is demonstrated by timing the end effector as it travels to one such point from every feeder where it can be observed and the time is identical. This led to a section of the PCB that is not important in feeder placement and theoretically could be prioritised below the components closer to the feeders (Fig. 11). The shaded zone shows the components that can have any feeder allocation and still take the same time for the end effector to move to them. This is demonstrated by comparing the time in which the end effector can move between a labelled component and each feeder (F1–12) (Table 4). The motion times for component JK5 (Fig. 11) show variation; therefore, it is affected by feeder allocation and is outside of the shaded zone. Component JK8 shows small variation and is on the edge of the zone and can be assigned any feeder apart from 12. The motion times for component JK9 show no variation and the component is inside the zone; therefore, any feeder could be assigned to this part.

6 Conclusions

This research investigated the application of an articulated 6-axis robot to assemble a multi-component PCB for an electronic product. The approach considers the need for a highly flexible automation system for medium- and low-volume

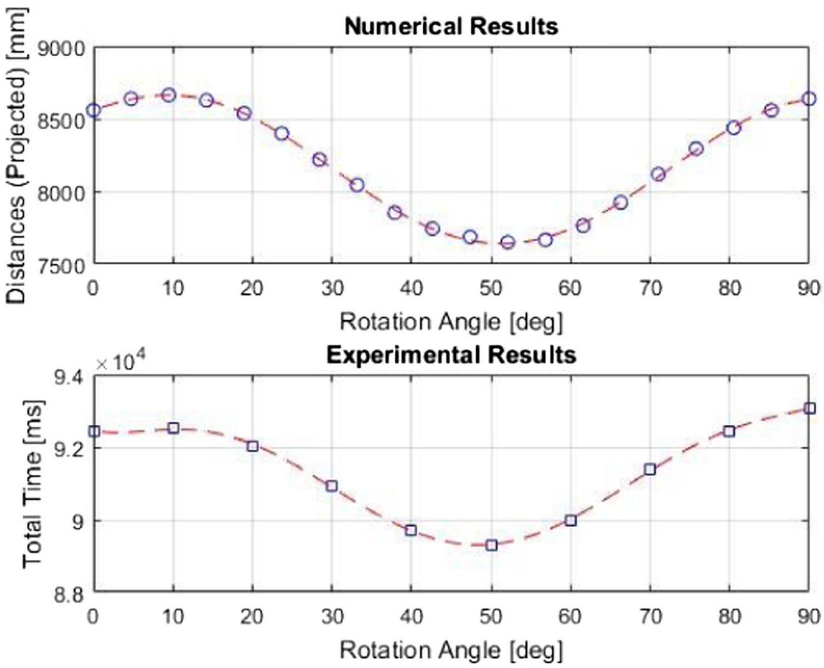

Fig. 10 Plot comparing experimental and numerical results of rotation angle effect

Table 3 Experimentally recorded translation and cycle time results

Direction	Translation (mm)	Time (ms)
D	− 300	87,540
D	0	92,448
D	250	98,844
H	100	92,616
H	− 200	92,952

manufacture. This non-traditional assembly robot for electronic assembly can be used to bridge the gap between time-consuming and costly manual builds and high-volume production lines using dedicated high investment automation equipment. To increase the potential of using a highly flexible robot with six degrees of freedom, a heuristic methodology using a genetic algorithm for planning the assembly operation was used. The investigation utilised a KUKA KR16 robotic assembly cell to validate the optimised programs and used condition monitoring to identify which of the path planning programs yielded the best results. The method was successfully implemented for determining the optimal location of feeders and parts in relation to the robot and the best sequence of robot movements in order to maximise production throughput. The main findings are as follows:

- The genetic algorithm showed an 80% probability of convergence toward the optimal solution and identified solutions to the combinatorial problems in this investigation. The feeder slot assignment problem was solved to find the best allocation of the various component types based on their position on the circuit board. The order sequence for component placement has also been optimised. Both solutions result in a significant reduction the total distance travelled by the KUKA robot leading to an important improvement to the assembly cycle.
- When considering the relationship between the KUKA robot global position and the possible angles at which the feeder and PCBs are arranged within the workspace, the *x*- and *y*-axis components of motion can be reduced. By arranging the PCB and feeders at a 45° angle to the robot primary axis, the assembly operation can be performed with a reduction in cycle time. Also by moving the work piece closer to the robot base, the cycle time is further reduced. This is due to a reduction in the moment required to move the arm and improved acceleration control within the robot system.

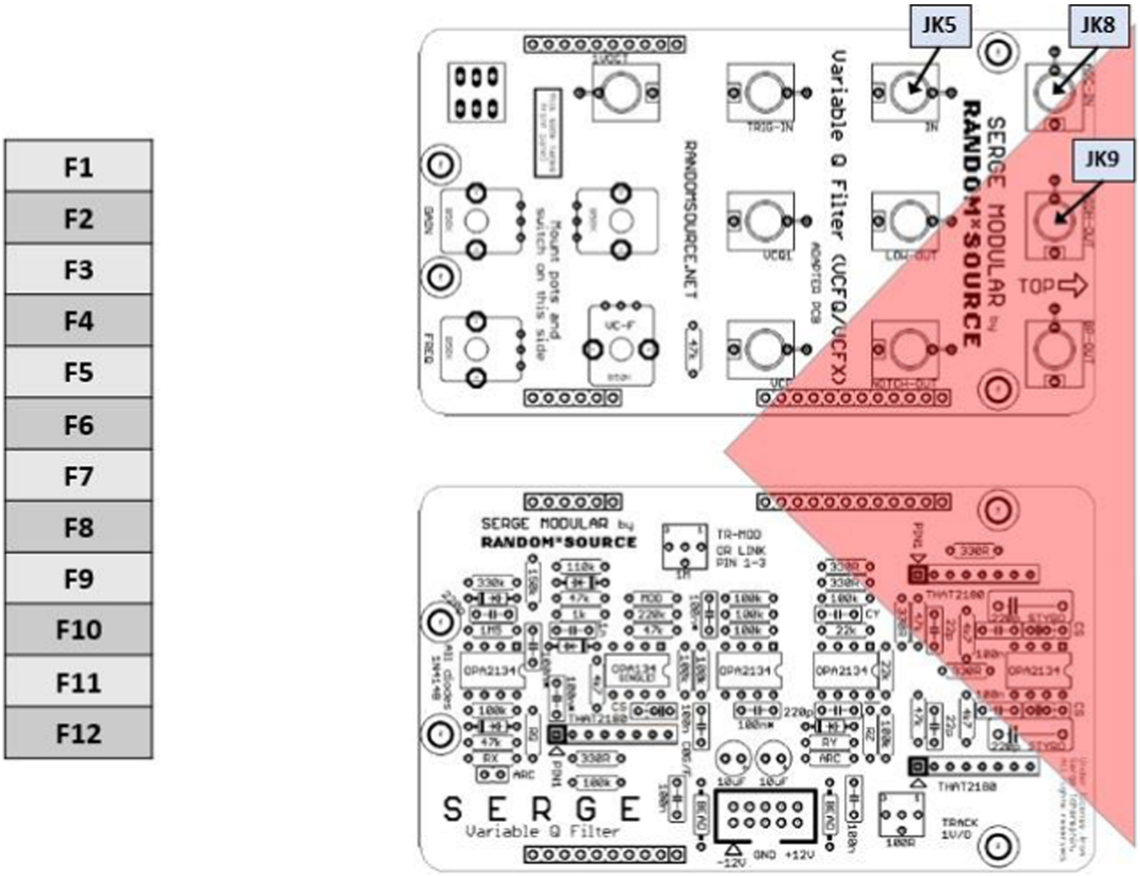

Fig. 11 Zone on PCB unaffected by feeder allocation and location of components JK5, JK8 and JK9

Table 4 Component to feeder motion time in milliseconds

Component	Time (ms)											
	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12
JK5	516	516	504	504	504	504	504	504	504	516	516	528
JK8	528	528	528	528	528	528	528	528	528	528	528	540
JK9	528	528	528	528	528	528	528	528	528	528	528	528

- It was found that some components on the PCB were unaffected by the feeder positioning assignment. This is due to them being in a position where the projected distance of the robot primary world axis is not reduced by changing the feeder position. This led to an area of the PCB that did not need to be prioritised in the feeder assignment problem. However, by changing the objective function to accommodate this finding, the total cycle time was not reduced. For the number of components tested, a change was not observed, but on a larger scale, changing the objective function in such a manner could provide improvements to cycle time.

- The experimental results were tested using the KUKA robot to validate the identified best solution. Using condition monitoring, the four different robot motion types were compared. Point-to-point motion provided the fastest assembly time; however, this did increase the vibration within the kinematic structure when compared to other path motions. Therefore, to avoid wear within the robot and unwanted vibrations, a reduction in operation velocity would be necessary when using this motion type.
- Flexible manufacturing is important in modern industry, and current 6-axis robot systems lend themselves to a wide range of automation activities. They exceed Cartesian and SCARA robots in manipulation capability and lend themselves to automation activities beyond basic pick and place. This research has shown that genetic algorithm methodologies can be used to solve a combination of problems and provides the basis for increased use of 6-axis robots in PCB assembly. It also showed that the relationship between the robot motion types and the position of the robot to the PCB assembly area needs to be considered when trying to solve the optimum sequence order.

Funding information This work was supported by the Advanced Sustainable Manufacturing Technologies (ASTUTE 2022) project, which is partly funded from the EU's European Regional Development Fund through the Welsh European Funding Office, in enabling the research upon which this paper is based. Further information on ASTUTE can be found at www.astutewales.com.

References

1. Crama Y, Flippo OE, van de Klundert J, Spieksma F (1997) The assembly of printed circuit boards: a case with multiple machines and multiple board types. Eur J Oper Res 98:457–472
2. Crama Y, van de Klundert J, Spieksma F (2002) Production planning problems in printed circuit board assembly. Discret Appl Math 123:339–361
3. Moghaddam M, Nof SY (2016) Parallelism of pick-and-place operations by multi-gripper robotic arms. Robot Comput Integr Manuf 42:135–146
4. Milutinović D, Dejan JR (2013) Redundancy in robot manipulators and multi-robot systems. Springer-Verlag, Berlin. https://doi.org/10.1007/978-3-642-33971-4
5. Erdos G, Kovacs A, Vancza J (2016) Optimized joint motion planning for redundant industrial robots. CIRP Ann Manuf Technol 65:451–454
6. Rubio F, Llopis-Albert C, Valero a F, Suñera JL (2016) Industrial robot efficient trajectory generation without collision through the evolution of the optimal trajectory. Robot Auton Syst 86:106–112
7. Fu B, Chen L, Zhou Y, Zheng D, Wei Z, Dai J, Pan H (2018) An improved A* algorithm for the industrial robot path planning with high success rate and short length. Robot Auton Syst 106:26–37
8. Wang C, Ho L-S, J. Cannon D (1998) Heuristics for assembly sequencing and relative magazine assignment for robotic assembly. Comput Ind Eng 34(2):423–431
9. Mukund Nilakantan J, Huang GQ, Ponnambalam SG (2015) An investigation on minimizing cycle time and total energy consumption in robotic assembly line systems. J Clean Prod 90:311–325
10. Ji P, Wan YF (2008) Minimizing cycle time for PCB assembly lines: an integer programming model and a branchand-bound approach. Inf Manag Sci 19(2):237–243
11. Kodek DM, Krisper M (2007) Optimal algorithm for minimizing production cycle time of a printed circuit board assembly line. Int J Prod Res 42(23):5031–5048
12. Holland JH (1975) Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. University of Michigan Press, Ann Arbor
13. Wang W, Brunn P (2000) An effective genetic algorithm for job shop scheduling. Proc Inst Mech Eng B J Eng Manuf 214(4):293–300

14. Lu C, Wong YS, Fuh JYH (2006) An enhanced assembly planning approach using a multi-objective genetic algorithm. Proc Inst Mech Eng B J Eng Manuf 220(2):255–272
15. Yildiz AR, Ozturk F (2006) Hybrid enhanced genetic algorithm to select optimal machining parameters in turning operation. Proc Inst Mech Eng B J Eng Manuf 220(12):2041–2053
16. Liao YG (2003) A genetic algorithm-based fixture locating positions and clamping schemes optimization. Proc Inst Mech Eng B J Eng Manuf 217(8):1075–1083
17. Sing PK, Jain SC, Jain PK (2005) Comparative study of genetic algorithm and simulated annealing for optimal tolerance design formulated with discrete and continuous variables. Proc Inst Mech Eng B J Eng Manuf 219(10):735–758
18. Ball MO, Magazine MJ (1988) Sequencing of insertions in printed circuit board assembly. Oper Res 36(2):192–201
19. Ahmadi RH, Mamer JW (1999) Routing heuristics for automated pick and place machines. Eur J Oper Res 117:533–552
20. Klomp C, van de Klundert J, Spieksma F, Voogt S (2000) The feeder rack assignment problem in PCB assembly: a case study. Int J Prod Econ 64:399–407
21. Li S, Hu C, Tian F (2008) Enhancing optimal feeder assignment of the multi-head surface mounting machine using genetic algorithms. Appl Soft Comput 8:522–529
22. Grunow M, Günther HO, Schleusener M, Yilmaz IO (2004) Operations planning for collect-and-place machines in PCB assembly. Comput Ind Eng 47:409–429
23. Sohn J, Park S (1996) Efficient operation of a surface mounting machine with a multihead turret. Int J Prod Res 34(4): 1131–1143
24. Broad K, Mason A, Ronnqvist M, Frater M (1996) Optimal robotic component placement. J Oper Res Soc 47:1343–1354
25. Deo S, Javadpour R, Knapp GM (2002) Multiple setup PCB assembly planning using genetic algorithms. Comput Ind Eng 42:1–16
26. Ho W, Ji P PCB assembly line assignment: a genetic algorithm approach. J Manuf Technol Manag 16(6):682–692
27. Ellis KP, Vittes FJ, Kobza JE (2001) Optimizing the performance of a surface mount placement machine. IEEE Trans Electron Packag Manuf 24(3):160–170
28. Magyar G, Johnsson M, Nevalainen O (1999) On solving single machine optimization problems in electronics assembly. J Electron Manuf 9(4):249–267
29. Suna D, Lee T, Kim KH (2005) Component allocation and feeder arrangement for a dual-gantry multi-head surface mounting placement tool. Int J Prod Econ 95:245–264
30. Kulak O, Yilmaz IO, Günther HO (2007) PCB assembly scheduling for collect-and-place machines using genetic algorithms. Int J Prod Res 45(17):3949–3969
31. Garcia-Naijera A, Brizuela CA (2005) PCB assembly: an efficient genetic algorithm for slot assignment and component pick and place sequence problems. 2005 IEEE Congress on Evol Comput 2:1485–1492
32. Hong DS, Cho HS (1999) A genetic-algorithm-based approach to the generation of robotic assembly sequences. Control Eng Pract 7: 151–159
33. Taha Z, Tahriri F (2010) Optimizing the robot traveling time (ORTT) in robot assembly line balancing problem (RALBP), The 11th Asia Pacific Industrial Engineering and Management Systems Conference