A Genetic Algorithm Approach to Optimising Component Placement and Retrieval Sequence for Chip Shooter Machines

W. Ho* and P. Ji

Department of Industrial and Systems Engineering

The Hong Kong Polytechnic University

Hung Hom, Kowloon, Hong Kong

E-mail: mfrsch25@inet.polyu.edu.hk, Phone: (852)27666631, Fax: (852)23625267

A chip shooter machine in printed circuit board (PCB) assembly has three movable mechanisms: a X-Y table carrying a PCB, a feeder carrier with several feeders holding components, and a rotary turret with multiple assembly heads to pick up and place components. In order to get the minimal placement or assembly time for a PCB on the machine, all the components on the board should be placed in a perfect sequence, and the components should be set up on a right feeder, or feeders since two feeders can hold the same type of components, and additionally, the assembly head should retrieve or pick up a component from a right feeder. The entire problem is very complicated, and this paper presents a genetic algorithm approach to tackle it.

Keywords: Chip shooter machines; Component retrieval; Component sequencing; Feeder arrangement; Genetic algorithms

* Corresponding author.

1. Introduction

Nowadays, the highly competitive electronic industry has forced the PCB manufacturers to optimise the placement process, which is the most time consuming one among the PCB assembly processes, for achieving a higher throughput rate. In order to optimise the placement process, several optimisation problems should be considered [1]. They include the assignment of components to machines, that is, the component allocation problem, the assignment of component types to feeders, that is, the feeder arrangement problem, and the sequencing of component placements, that is, the component sequencing problem. In this paper, the component sequencing and the feeder arrangement problems are considered for a chip shooter machine. Except these two problems, an additional optimisation problem is also taken into consideration. It is due to the fact that some components of same types may be in hundreds or even more. The movement of the feeder carrier can be reduced if two feeders are arranged for holding such component types with frequent use. As a consequence, the placement or assembly time can be minimised. In case some component types are assigned to two feeders, it is then necessary to determine which feeder a component should be retrieved from, that is, the component retrieval problem. So, the problem we face in this paper is to solve the component sequencing, the feeder arrangement, and the component retrieval problems for the chip shooter machine, with the objective of minimising the total placement time.

Since the placement time is also dependent on a feeder to hold which types of components besides the pick and placement sequence in the chip shooter machines, hence the three sub-problems should be simultaneously studied and solved. Actually, the individual sub-problems are already very hard and complex [2]. Moreover, the problem sizes in the real cases prohibit the effective use of deterministic approaches. So, a genetic algorithm is developed to solve the combined problem of the three sub-problems in order to obtain good solutions in reasonable amount of time.

The chip shooter machine is the most commonly used due to its high speed. The machine, as shown in Fig. 1, consists of a movable X-Y table carrying a PCB, a movable feeder carrier with several feeders holding components, and a rotary turret with multiple assembly heads (usually 10 or 12). Each assembly head has several (normally 5) nozzles with different sizes. A large nozzle is used to pick up and place large components. The operation sequence of the chip shooter machine is that: As the first board of a batch enters the machine, the first nozzle of the turret picks up a component from a feeder. Then the turret indexes one step and the next nozzle picks up the second component. After that the turret indexes again to pick up the next component, and so on. In the same moment, the PCB is moved to the placement location waiting for the first component to be placed on the board. When the sixth component is being picked up if the turret has 10 heads, the first component is being placed on the board at the same time. These operations continue such as the turret indexes one step, the feeder carrier moves to the location containing the next pick-up component, and the X-Y table moves to the next placement location. In the assembly of the last five components, there is no need to pick up components for the board being assembled. However, the nozzles of the turret can pick up the first five components for the next board to be assembled if necessary. For the first few components assembled in a batch of PCBs, there are only pick-up movements and no placement movement. For the last few components of the same batch, there are only placement movements and no pick-up movement. Therefore, if the PCB's quantity in a batch is very large, these boundary effects can be neglected [3].

2. Literature Review

There were a number of researches concerning the optimisation of the component sequencing problem, the feeder arrangement problem, and the component retrieval problem for the chip shooter machines. But, many of them focused on the individual problems for the machine.

Some researchers [4,5] studied the component sequencing problem only, whereas some researchers [6-8] worked on the feeder arrangement problem alone. Similarly, Crama et al. [9] also investigated the component retrieval problem merely for a chip shooter machine.

Besides, many researchers solved the problems separately for the chip shooter machines. Bard et al. [10] used an iterative two-step approach to determine the component placement sequence, the feeder arrangement, and the component retrieval plan. Although three problems were considered, they were solved separately. Sohn and Park [11] formulated the component sequencing and the feeder arrangement problems as a nonlinear programming model while using the one-head case as an approximation to the multi-head case. A heuristic approach was developed to solve the problems sequentially. Yeo et al. [12] developed a rule-based frame system to generate the feeder arrangement first, followed by the component placement sequence for a chip shooter machine. Crama et al. [2] proposed a solution procedure based on a hierarchical decomposition of the planning problems for a chip shooter machine. They solved the feeder arrangement problem heuristically first, and then solved the problems of component sequencing and the component retrieval using the constructive heuristics and the local search methods. Ellis et al. [13] developed a heuristic approach to determine the component placement sequence first, and then the feeder arrangement for a chip shooter machine.

Leu et al. [3], Moyer and Gupta [14], and Ho and Ji [15] studied and solved the component sequencing and the feeder arrangement problems simultaneously for a chip shooter machine. But, none of them have mentioned about the component retrieval problem. Therefore, in this paper, a hybrid genetic algorithm is presented to solve the combined problem of the three sub-problems simultaneously for the machine.

3. Hybrid Genetic Algorithm

The success of Genetic Algorithms (GAs) in solving a wide variety of complex optimisation problems [16] and the advantages of GAs such as simplicity, easy operation, and great flexibility have prompted us to apply GA to solve the combined problem. As mentioned in Section 1, the problem we face in this paper can be regarded as the combination of three hard optimisation problems. So, a simple GA may not perform well in this situation. The GA developed in this paper is therefore hybridised with several heuristics in order to improve the solution further. However, it was found that none of the researchers have studied the three sub-problems for the chip shooter machine using a hybrid GA (HGA). The flowchart of the HGA for the combined problem is illustrated in Fig. 2. The algorithm is so called because three heuristics are incorporated in it. First, the nearest neighbor heuristic (NNH) is applied for generating the first link, which is the sequence of component placements, in the initial population. Second, a heuristic called the iterated swap procedure (ISP) [15] is applied to improve the first link. Third, the 2-opt local search heuristic is adopted to improve the second link, which is the feeder arrangement. Actually, the principle of the ISP is very similar to that of the 2-opt local search heuristic, except that some instead of all 2 swaps are examined. The computational effort will be high if the 2-opt local search is performed on the first link because the number of components is quite large, normally several hundreds.

Besides, the roulette wheel selection operation is adopted in the HGA to select chromosomes to perform the genetic operations, which include one crossover and two mutations. They are the modified order crossover [15], the heuristic mutation [16], and the inversion mutation [16]. Generally, GAs developed especially for the PCB assembly problems are not limited to using one crossover and one mutation. Due to the complexity of the problems, several genetic operations will be adopted [3].

In the following sub-sections, the way of representing a chromosome for the combined problem of the three sub-problems for a chip shooter machine is explained first, followed by the discussion of the component retrieval plan. After that, the fitness function for calculating the total placement time of a chromosome is presented.

3.1 Encoding

A chromosome is illustrated in Fig. 3 with the two-link representation, in which the number inside the bracket in link 1 represents the component type. For example, if the sequence of placements starts with component 3, then component 1, component 2, component 5, component 4, and finally component 6, its sequence can be represented as (3 1 2 5 4 6), that is the first link in a chromosome, as shown in Fig. 3.

For example, the six components are from three component types. If type 2 is assigned to feeder 1, whereas types 3 and 1 are assigned to feeders 2 and 3, respectively, then the feeder arrangement can be represented as (2 3 1). Sometimes, the number of feeders available is more than that of component types required. Component types can therefore be assigned to more than one feeder. Nevertheless, it was observed that no more than two feeders with the same component type could be used on an assembly line [2]. In the other words, at most a particular component type can be stored in two feeders at the same time. Not surprisingly, component types with greater use should have the priority to be assigned to the surplus feeders. In the above example, it is assumed that one surplus feeder is available. One component type can be stored in that feeder as well. Since component type 2 is the most frequently used, it is assigned to the surplus feeder (i.e., feeder 4). The feeder arrangement is now encoded as (2 3 1 2), that is the second link in a chromosome, as illustrated in Fig. 3.

3.2 Retrieval Plan

When a component type is assigned to two feeders, a retrieval plan must be set up so as to determine which feeder a component should be retrieved from. The retrieval plan adopted in this paper is similar to the NNH. For the first component, if its type is stored in two feeders, then select a feeder randomly. But if the types of the remaining components are stored in two feeders, then select the feeders as close as possible to the previous ones so that the movements of the feeder carrier are minimised. For example, if a component type is stored in two feeders, say $v1$ and $v2$, and the previous feeder to be visited is u , then select $v1$ if $|x_{v1} - x_u| < |x_{v2} - x_u|$, or select $v2$ provided that $|x_{v1} - x_u| > |x_{v2} - x_u|$. In case $|x_{v1} - x_u| = |x_{v2} - x_u|$, then select a feeder randomly.

Based on the retrieval plan, the component retrieval problem for the above example can be solved. First, based on the random selection, it is supposed that component type 2 is retrieved from feeder 1 for the first component (i.e., component 3). Since the next component (i.e., component 1) is of type 2 again, it is undoubtedly that the component should be retrieved from feeder 1 rather than feeder 4. For the placement of component 4, type 2 should be retrieved from feeder 4 in this time. It is due to the fact that the distance for the feeder carrier to travel from the previous feeder (i.e., feeder 3) to feeder 4 is shorter than that for it to travel from feeder 3 to feeder 1 (i.e., $|x_1 - x_3| > |x_4 - x_3|$).

The operation sequence of the chip shooter machine is that: feeder 1 is moved to the pick-up location, and a nozzle picks up the first component of type 2 in feeder 1, and the turret rotates one step. After that, feeder 1 remains stationary in the pick-up location, and the second nozzle picks up a component of type 2 again. The feeder carrier moves and the turret rotates again to pick up the third component of type 3 from feeder 2 and the fourth component of type 1 from feeder 3. After the fifth component of type 2 is picked up from feeder 4, the PCB moves so that component number 3 is located at the placement location. When the sixth component of

type 1 is being picked up from feeder 3 if the turret has 10 heads, the first component is being placed at the same time. The sequence of component placements, c , will be (3 1 2 5 4 6), while the sequence of the feeders to be visited by the pick-up nozzle, f , will be (1 1 2 3 4 3).

3.3 Evaluation

The objective of the combined problem is to minimise the total placement time for assembling all electronic components on a PCB using a chip shooter machine. So certainly, the fitness function should be the total placement time, which is the summation of all dominating times of components. It is because three mechanisms of the chip shooter machine move at different speeds, such as the traveling time of the PCB or the X-Y table, the traveling time of the feeder carrier, and the indexing time of the turret. The longest one among the three is the dominating time needed in the assembly of the component. Let T_i be the time needed for the placement of component i and $eval(X_h)$ be the total placement time or the fitness function for chromosome X_h in the combined problem, then

$$T_i = \max(t_1(c(i-1), c(i)), t_2(f(i+g), f(i+g+1)), t_3)$$

$$eval(X_h) = \sum_{i=1}^n T_i$$

where

$t_1(a, b)$ is the traveling time of the X-Y table from component a to component b .

$$t_1(a, b) = \max\left(\frac{|x_b - x_a|}{V_x}, \frac{|y_b - y_a|}{V_y}\right) \text{ for Chebyshev metric, and}$$

V_x and V_y are the speeds of the X-Y table in the x and y directions, respectively.

$t_2(u, v)$ is the traveling time of the feeder carrier from feeder u to feeder v .

$$t_2(u, v) = \frac{|x_v - x_u|}{V_f}, \text{ and } V_f \text{ is the speed of the feeder carrier.}$$

t_3 is the indexing time of the turret.

n is the number of components.

$c(i)$ is the location in the board for the i th component.

$f(i)$ is the feeder location for the i th component type.

g is the number of components in the gap between the pick-up component and the placement component in the turret, and normally, $g = 5$ or 6 .

In the above expression, $c(i - 1) = c(n)$ when $i = 1$. When $f(l)$ has $l > n$, where $l = i + g$ or $i + g + 1$, $f(l)$ is replaced by $f(l - n)$, which represents the initial g components of the next board in the batch.

4. Performance Analysis

The performance of the HGA is evaluated by use of the PCB example in Leu et al. [3]. The example has 50 components with 10 different component types. Two tests are conducted in order to examine the effectiveness and the efficiency of the HGA. First, the 50-component problem is solved using the HGA in which only 10 feeders are available, which means that each component type can only be stored in exactly one feeder. Second, the 50-component problem is solved again using the HGA in which three surplus feeders are available. In this case, three types of components can be assigned to two feeders. According to the 50-component problem, component types 4, 9, and 10 are the most frequently used, these three types of components can therefore be stored in two feeders. The results of the HGAs are also compared with the solution of the simple GA adopted by Leu et al. [3]. They assumed that the number of feeders available was equal to that of component types required.

The parameters of the HGAs for the problem are preset as: population size = 25, iteration number = 1000, crossover rate = 0.4, and mutation rate = 0.2. Fig. 4 shows the best placement time at each iteration for the HGA to solve the 50-component problem with 3 surplus feeders, whereas the comparison between the HGAs' results and solution of the simple GA [3] is

summarised in Table 1. Two conclusions can be drawn. First, it is found that the performance of the HGAs is superior to that of the simple GA. The HGAs can obtain a better solution using a smaller population size as well as fewer iterations. Second, the solution is even better if the number of feeders available is set more than that of component types. Since the three most frequently used component types are assigned to more than one feeder, they can be retrieved from a closer feeder according to the retrieval plan in Section 3.2. So, the traveling time of the feeder carrier can be shortened. As a result, the total placement time can be minimised too.

5. Conclusions

A hybrid genetic algorithm was applied successfully to the combined problem of the component sequencing, the feeder arrangement, and the component retrieval sub-problems for a chip shooter machine. A two-link representation together with the retrieval plan was used to solve the problem. The algorithm was so called because three heuristics were incorporated to improve the solution. They included the nearest neighbor heuristic, the 2-opt local search heuristic, and a heuristic called the iterated swap procedure. Finally, it was shown that the performance of the HGAs was superior to that of the simple GA in terms of the total placement time. Furthermore, the solution was even better when component types could be stored in more than one feeder.

Acknowledgements

The authors wish to acknowledge The Hong Kong Polytechnic University for the financial support of the project (No. G-W031).

References

- [1] M. O. Ball and M. J. Magazine, "Sequencing of insertions in printed circuit board assembly", *Operations Research*, 36(2), pp. 192-201, 1988.
- [2] Y. Crama, O. E. Flippo, J. V. D. Klundert and F. C. R. Spieksma, "The assembly of printed circuit boards: a case with multiple machines and multiple board types", *European Journal of Operational Research*, 98, pp. 457-472, 1997.
- [3] M. C. Leu, H. Wong and Z. Ji, "Planning of component placement/insertion sequence and feeder setup in PCB assembly using genetic algorithm", *Journal of Electronic Packaging*, 115, pp. 424-432, 1993.
- [4] R. D. Souza and L. J. Wu, "Intelligent optimization of component onsertion in multi-head concurrent operation PCBA machines", *Journal of Intelligent Manufacturing*, 6, pp. 235-243, 1995.
- [5] L. K. Moyer and S. M. Gupta, "An efficient assembly sequencing heuristic for printed circuit board configurations", *Journal of Electronics Manufacturing*, 7(2), pp.143-160, 1997.
- [6] L. K. Moyer and S. M. Gupta, "SMT feeder slot assignment for predetermined component placement paths", *Journal of Electronics Manufacturing*, 6(3), pp. 173-192, 1996.
- [7] A. Dikos, P. C. Nelson, T. M. Tirpak and W. Wang, "Optimization of high-mix printed circuit card assembly using genetic algorithms", *Annals of Operations Research*, 75, pp. 303-324, 1997.
- [8] C. Klomp, J. V. D. Klundert, F. C. R. Spieksma and S. Voogt, "The feeder rack assignment problem in PCB assembly: a case study", *International Journal of Production Economics*, 64, pp. 399-407, 2000.

- [9] Y. Crama, O. E. Flippo, J. V. D. Klundert and F. C. R. Spieksma, "The component retrieval problem in printed circuit board assembly", *International Journal of Flexible Manufacturing Systems*, 8, pp. 287-312, 1996.
- [10] J. F. Bard, R. W. Clayton and T. A. Feo, "Machine setup and component placement in printed circuit board assembly", *International Journal of Flexible Manufacturing Systems*, 6, pp. 5-31, 1994.
- [11] J. Sohn and S. Park, "Efficient operation of a surface mounting machine with a multihead turret", *International Journal of Production Research*, 34(4), pp. 1131-1143, 1996.
- [12] S. H. Yeo, C. W. Low and K. H. Yong, "A rule-based frame system for concurrent assembly machines", *International Journal of Advanced Manufacturing Technology*, 12, pp. 370-376, 1996.
- [13] K. P. Ellis, F. J. Vites and J. E. Kobza, "Optimizing the performance of a surface mount placement machine", *IEEE Transactions on Electronics Packaging Manufacturing*, 24(3), pp. 160-170, 2001.
- [14] L. K. Moyer and S. M. Gupta, "Simultaneous component sequencing and feeder assignment for high speed chip shooter machines", *Journal of Electronics Manufacturing*, 6(4), pp. 271-305, 1996.
- [15] W. Ho and P. Ji, "Component scheduling for chip shooter machines: a hybrid genetic algorithm approach", *Computers & Operations Research*, 30, pp. 2175-2189, 2003.
- [16] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Design*, Wiley, New York, 1997.

Fig. 1. The schematic diagram of the chip shooter machine.

Fig. 2. The flowchart of the HGA.

Fig. 3. The two-link representation for a chromosome.

Fig. 4. The best placement time at each iteration.

Fig. 1. The schematic diagram of the chip shooter machine.

Fig. 2. The flowchart of the HGA.

Fig. 3. The two-link representation for a chromosome.

Fig. 4. The best placement time at each iteration.

Table 1. A comparison of the experimental results.

	HGA	HGA	Leu et al. [3]
Number of surplus feeders	3	N/A	N/A
Population size	25	25	100
Iteration number	685	323	about 1,750
Final best solution (seconds)	25	26	about 51.5

Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:

Ho, W;Ji, P

Title:

A genetic algorithm approach to optimising component placement and retrieval sequence for chip shooter machines

Date:

2006-03-01

Citation:

Ho, W. & Ji, P. (2006). A genetic algorithm approach to optimising component placement and retrieval sequence for chip shooter machines. INTERNATIONAL JOURNAL OF ADVANCED MANUFACTURING TECHNOLOGY, 28 (5-6), pp.556-560. <https://doi.org/10.1007/s00170-004-2390-2>.

Persistent Link:

<http://hdl.handle.net/11343/118692>