Optimization of PCB component placements for the collect-and-place machines

William Ho[1,*], Ping Ji[2], and Prasanta Kumar Dey[3]
[1, 3]Operations and Information Management Group
Aston Business School, Aston University
Birmingham B4 7ET, United Kingdom
[1]E-mail: w.ho@aston.ac.uk; Tel: +44 (0) 121 2043342

[2]Department of Industrial and Systems Engineering,
The Hong Kong Polytechnic University,
Hung Hom, Kowloon, Hong Kong

Abstract

This paper presents two hybrid genetic algorithms (HGAs) to optimize the component placement operation for the collect-and-place machines in printed circuit board (PCB) assembly. The component placement problem is to optimize (i) the assignment of components to a movable revolver head or assembly tour, (ii) the sequence of component placements on a stationary PCB in each tour, and (iii) the arrangement of component types to stationary feeders simultaneously. The objective of the problem is to minimize the total traveling time spent by the revolver head for assembling all components on the PCB. The major difference between the HGAs is that the initial solutions are generated randomly in HGA1. The Clarke and Wright saving method, the nearest neighbor heuristic, and the neighborhood frequency heuristic are incorporated into HGA2 for the initialization procedure. A computational study is carried out to compare the algorithms with different population sizes. It is proved that the performance of HGA2 is superior to HGA1 in terms of the total assembly time.

Keywords: Printed circuit board manufacturing; Collect-and-place machines; Genetic algorithms; Component grouping; Component sequencing; Feeder arrangement.

[*] Corresponding author

1. Introduction

The printed circuit boards (PCBs) have been extensively applied in numerous contemporary electronic products such as personal computers, mobile phones, and audio-video equipment. This has placed an unprecedented demand for PCBs. Besides, customers' changing needs such as for smaller product size and more functions are forcing the surface mount technology (SMT) to replace the plated-through-hole technology. The PCB assembly process in the SMT environment consists of five major operations. First of all, solder paste is applied to where the electronic components will be placed on. Typically, it is applied in a screen print fashion. Then, it is followed by the placement operation. A high-speed placement machine (e.g., the chip shooter machine) is used to mount small components such as chip resistors on the PCB first. A flexible placement machine (e.g., the pick-and-place machine) is then adopted to mount large components such as integrated circuits on the PCB. After all the components have been assembled, the PCB is inspected whether there is any missing component or not. Subsequently, the PCB is conveyed through an oven, which causes the solder paste to reflow and form the solder joints. Finally, the PCB must be cleaned in order to remove the contaminants exposed during fabrication and assembly.

Among the assembly operations, the component placement one is generally the most time-consuming (Souza and Wu, 1995; Ong and Khoo, 1999; Ong and Tan, 2002). In addition, it is frequently a bottleneck of an assembly line (Dikos et al., 1997; Csaszar et al., 2000; Ong and Tan, 2002), and determines the line's cycle time (Wilhelm and Tarmy, 2003). Evidently, the throughput rate of a PCB manufacturing company or the company's competitiveness can be increased if the component placement process is optimized. So, the focus of this paper is confined to the optimization of the component placement process. The objective is to minimize the placement time spent for assembling all electronic components to a board using an SMT placement machine.

In PCB assembly, there are several types of SMT placement machines. Each type of machines possesses its own peculiarities as well as operation. Among them, the three most popular types are the pick-and-place, chip shooter, and collect-and-place (CAP) machines. In the pick-and-place machine, components of the same type are stored in a single stationary feeder, whereas the PCB is secured on a fixed working table. During the placement operation, the assembly head travels to pick up one component at a time from a feeder, and then places it on the stationary board. The pick-and-place machine can achieve high accuracy. Moreover, it is suitable for operating with large components such as integrated circuits.

The second type of machine is the chip shooter machine. It possesses an 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. Each assembly head has several nozzles of different sizes. A large nozzle is used to pick up and place large components. The major advantage of the chip shooter machine is its high speed because the pickup and placement operations are performed concurrently. However, it is only preferable for operating with small components such as chip resistors.

In the last two decades, the component placement problem has been studied thoroughly for both pick-and-place machine (Ball and Magazine, 1988; Ji et al., 1992; Foulds and Hamacher, 1993; Leu et al., 1993; Francis et al., 1994; Kumar and Li, 1995; Broad et al., 1996; Magyar et al., 1999; Ong and Khoo, 1999; Ho and Ji, 2004; Ho and Ji, 2005), and chip shooter machine (Leu et al., 1993; Bard et al., 1994; Souza and Wu, 1995; Moyer and Gupta, 1996a; Moyer and Gupta, 1996b; Sohn and Park, 1996; Yeo et al., 1996; Crama et al., 1997; Dikos et al., 1997; Moyer and Gupta, 1997; Klomp et al., 2000; Ellis et al., 2001; Ong and Tan, 2002; Ho and Ji, 2003; Wilhelm and Tarmy, 2003; Ho and Ji, 2006). Due to this reason, this paper studies the component placement problem for the CAP machine.

The CAP machine, as shown in Fig. 1, combines the advantages of the pick-and-place machine and those of the chip shooter machine. It can assure high placement accuracy and also achieve high placement speed. Furthermore, it is suitable for operating with a wide variety of components ranging from chip resistors to integrated circuits. This type of machines has multiple stationary feeders holding components, a stationary working table securing a PCB, and two or even more revolver heads with several vacuum nozzles (normally, 12). Each feeder of the machine can only hold components of the same type. In this paper, three basic assumptions are made. First of all, the CAP machine being studied consists of one revolver head in which there are 12 nozzles. Second, the head is always fully loaded with 12 components except for the final assembly tour. Third, each component type can only be assigned to a single feeder. The operation sequence of the CAP machine is that: The revolver head moves from its original location to a stationary feeder carrying components at the beginning. The first nozzle of the head picks up a component at a time, and then rotates one step. The second nozzle of the head picks up another component from the previous feeder if the next component is the same type with the previous one or moves to another feeder to pick up the next component if it is different from the previous one. The head repeats this pick-up operation until the head is fully loaded. After that, the head travels to the PCB, and places the components on it sequentially. This operation procedure is regarded as an assembly tour in which the placement sequence is the same as the pick-up sequence. In addition, the revolver head performs stepwise rotational movements only in one or forward direction while reverse rotation is not allowed. Another assembly tour is carried out until all components are placed on the board.

In order to minimize the assembly time for the CAP machine, three interrelated problems should be solved simultaneously. They include the assignment of components to assembly tours or which set of components is placed by the same assembly tour (i.e., the component grouping problem), the determination of component placement sequence (i.e., the component sequencing problem), and the arrangement of component types to feeders (i.e., the feeder arrangement problem). Since the traveling distance or assembly time is dependent on all these three problems, they are studied and solved in this paper.

This paper is organized as follows. Section 2 reviews the relevant literature concerning the component placement problem for the CAP machine. Section 3 describes two hybrid genetic algorithms for optimizing the component placement problem. Section 4 compares the performance of the algorithms, and studies the effect of population sizes on the solutions' quality. Finally, Section 5 concludes the paper.

2. Literature Review

To our best knowledge, there are only three publications in the international literature studying the component placement problem for the CAP machine. Altinkemer et al. (2000) formulated a mathematical model for the component placement problem with a single revolver head. In their approach, it was assumed that the revolver head can only pick up components of the same type from a stationary feeder and place them one by one at their predefined positions on a stationary board in a tour. The capacity of the revolver head, however, may exceed the number of components of the same type. On this occasion, extra tours are incurred. The total distance traveled by the revolver head or the placement time for assembling the components on the board is, therefore, longer. Kazaz and Altinkemer (2003) extended the mathematical model formulated in Altinkemer et al. (2000) by allowing components of the same type to be stored in multiple feeders.

Grunow et al. (2004) proposed a three-stage heuristic approach to solve the component placement problem separately for the CAP machine with one revolver head. At the first stage, component types were assigned to feeders based on the neighborhood relations between the different types of components and the corresponding placement locations on the PCB. At the second stage, the component placement sequence was determined based on the Clarke and Wright saving method. Finally, the 2-opt local search heuristic was applied to improve the solutions. It was, however, proved that the efficiency of component placement process in the SMT environment is also dependent on a feeder to hold which types of components besides the pick and placement sequence. If the arrangement of components to feeders is not done carefully, even if the pick and placement sequencing is optimally solved, it can cause significant deterioration in the machines' performance (Altinkemer et al., 2000).

Based on the inadequacies of the above approaches, this paper develops two hybrid genetic algorithms to solve the component grouping, component sequencing, and feeder arrangement problems simultaneously rather than separately as Grunow et al. (2004) did. In addition, it is valid that the revolver head can pick up components of the different types instead of the same type only as Altinkemer et al. (2000) and Kazaz and Altinkemer (2003) proposed.

3. Hybrid Genetic Algorithms

Genetic algorithm (GA), developed by John Holland in the 1960s, is a stochastic optimization technique. Similar to other heuristic methods like simulated annealing (SA) and tabu search (TS), GA can avoid getting trapped in a local optimum by the aid of one of the genetic operations called mutation. The basic idea of GA is to maintain a population of candidate solutions that evolves under a selective pressure. Hence, it can be viewed as a class of local search based on a solution-generation mechanism operating on attributes of a set of solutions rather than attributes of a single solution by the move-generation mechanism of the local search methods, like SA and TS (Osman and Kelly, 1996). In the recent years, it has been successfully applied to a wide variety of hard optimization problems such as the traveling salesman and quadratic assignment problems (Goldberg, 1989; Gen and Cheng, 1997). The success is critical due to GA's simplicity, easy operation, and great flexibility. These are the major reasons why GA is selected as an optimization tool in this paper.

GA starts with an initial set of random solutions, called population. Each solution in the population is called a chromosome, which represents a point in the search space. The chromosomes evolve through successive iterations, called generations. During each generation, the chromosomes are evaluated using some measures of fitness. The fitter the chromosomes, the higher the probabilities of being selected to perform the genetic operations: crossover and mutation. In the crossover phase, the GA attempts to exchange portions of two parents (i.e., two chromosomes in the population) to generate an offspring. The crossover operation speeds up the process to reach better solutions. In the mutation phase, the mutation operation maintains the diversity in the population to avoid being trapped in a local optimum. A new generation is formed by selecting some parents and some offspring according to their fitness values, and by rejecting others to keep the population size constant. After the predetermined number of generations is performed, the algorithm converges to the best chromosome, which hopefully represents the optimal solution or may be a sub-optimal solution to the problem.

As mentioned earlier in Section 1, the problem being studied can be regarded as an integration of three hard optimization problems: the component grouping, component sequencing, and feeder arrangement problems. An individual problem is already complex, and difficult to solve (Crama et al., 1997). So, a simple GA may not perform well in this situation. The GA developed in this paper is, therefore, hybridized with several heuristics in order to improve the solution further. Nevertheless, it is found that none of the researchers have solved the problems simultaneously for the CAP machine using a hybrid GA (HGA).

The flowchart of the two HGAs for the component placement problem is shown in Fig. 2. The difference between them is that HGA1 only hybridizes an improved heuristic called the iterated swap procedure (ISP). Besides the ISP, HGA2 also hybridizes the Clarke and Wright saving method, the nearest neighbor heuristic, and the neighborhood frequency heuristic for the initialization procedure. The procedure of the HGAs is described as follows: After the GA parameters such as the iteration number, the population size, the crossover rate, and the mutation rate, have been set, the HGA generates the initial chromosomes or solutions to the problem. Each chromosome contains two links. The first link represents the sequence of component placements, whereas the second link represents the arrangement of component types to feeders. In HGA1, both links are generated randomly. In HGA2, conversely, the Clarke and Wright saving method is used to divide a set of components into groups. Each group of components is picked up and placed in the same assembly tour. Then, the nearest neighbor heuristic (NNH) is used to determine the sequence of component placements in each group. After the component grouping and component sequencing problems are solved, the first link can be generated. Based on the first link, the second link can be generated using the neighborhood frequency heuristic. After generating a population of initial chromosomes, the ISP is performed on the two links. Each chromosome is then measured by an evaluation function. The roulette wheel selection operation is adopted to select some chromosomes for the genetic operations including the order crossover, the heuristic mutation, and the inversion mutation. After a new chromosome or offspring is produced, all its two links are improved by the ISP. The fitness of the offspring will be measured and may become a member of the population if it possesses a relatively good quality. These steps form an iteration, and then the roulette wheel selection is performed again to start the next iteration. The HGAs will not stop unless the predetermined number of iterations is conducted.

3.1. Encoding

The chromosome to the component placement problem is encoded by the two-link representation as illustrated in Fig. 3. Link 1 represents the sequence of component placements, whereas Link 2 denotes the assignment of component types to feeders.

Consider a PCB with ten components (numbered as 1, 2, …, 10) to be assembled by the CAP machine in which the revolver head consists of five nozzles. Since the head can only pick up and place five components in each tour, the ten components are divided into two groups. For example, the components are grouped as:

- First group: components 1, 2, 4, 5, and 10;
- Second group: components 3, 6, 7, 8, and 9.

And, the sequence of component placements is as follows:

- First group: components 1, 2, 5, 10, and 4;
- Second group: components 9, 7, 3, 6, and 8.

Link 1 can be represented as (1 2 5 10 4 9 7 3 6 8). The number inside the bracket in Link 1 represents the component type. In this case, component 1 of type 4 is placed in advance in the first assembly tour, whereas component 9 of type 1 is placed first in the second assembly tour. Besides, Link 2 shows the assignment of component types to feeders. If component types 3, 4, 1, and 2 are stored in feeders 1, 2, 3, and 4, respectively, Link 2 can be represented as (3 4 1 2). Therefore, the head has to visit feeder 2 or f_2 initially to pick up a component of type 4 for component 1 or c_1. The entire assembly sequence of the head is shown in the following:

- First tour (pick-up): original location $\rightarrow f_2 \rightarrow f_1 \rightarrow f_2 \rightarrow f_3 \rightarrow f_4$;
  First tour (placement): $f_4 \rightarrow c_1 \rightarrow c_2 \rightarrow c_5 \rightarrow c_{10} \rightarrow c_4$;
- Second tour (pick-up): $c_4 \rightarrow f_3 \rightarrow f_2 \rightarrow f_3 \rightarrow f_1 \rightarrow f_4$;
  Second tour (placement): $f_4 \rightarrow c_9 \rightarrow c_7 \rightarrow c_3 \rightarrow c_6 \rightarrow c_8$.

3.2. Initialization

Three heuristic methods are hybridized in HGA2 to generate initial chromosomes. First, the Clarke and Wright (1964) saving method is used to solve the component grouping problem. The saving regarded in this paper is the traveling time of the revolver head for assembling the components. The method is to construct a saving matrix, $S(a, b)$, for every two components first: $S(a, b) = T(0, a) + T(0, b) - T(a, b)$, where $T(a, b)$ is the traveling time for moving from point a to point b, and 0 refers to the original location of the revolver head. Then, the components with larger saving value are grouped in the same assembly tour.

The NNH (Reinelt, 1994) is used to solve the component sequencing problem. The principle of the NNH is to start with the first component randomly, then to select the next component as close as possible to the previous one from those unselected components to form the placement sequence until all components are selected.

The neighborhood frequency heuristic is used to solve the feeder arrangement problem. Based on the sequence of component placements, the heuristic calculates the neighborhood relationships or frequencies between every two component types. Then, the component types with higher frequencies are assigned to the closer feeders.

3.3. Improvement

The 2-opt local search heuristic is generally used to improve the solutions to the hard optimization problems (Grunow et al., 2004). It is found that, however, this heuristic increases the computational time since every two swaps are examined. If a new solution generated is better than the original one or parent in terms of quality, it will replace and become the parent. All two swaps are examined again until there is no further improvement in the parent. In order to increase efficiency, the ISP is used to improve the two links of each initial solution as well as each offspring generated by the three genetic operators. Actually, the principle of the ISP, as illustrated in Fig. 4, is very similar to that of the 2-opt local search heuristic, except that some instead of all two swaps are examined. The procedure of the ISP is as follows:

Step 1: Select two genes randomly from a link of a parent.

Step 2: Exchange the positions of the two genes to form an offspring.

Step 3: Swap the neighbors of the two genes to form four more offspring.

Step 4: Evaluate all offspring and find the best one.

Step 5: If the best offspring is better than the parent, replace the parent with the best offspring and go back to Step 1. Otherwise, stop.

It is noticed that the ISP may interchange the components within the same assembly tour (intra-tour improvement), or swap a component from one component group to another (inter-tour improvement). The type of improvements performed is dependent on the selection of components in Step 1.

3.4. Evaluation

For the CAP machine, the fitness function used is the summation of all the pick-up times shown in equation (1), and placement times shown in equation (2) of the assembly tours. When the revolver head moves from a feeder/component’s location to another, the head rotates one step. The indexing time of the revolver head is also taken into consideration in such occasions. Let *eval*(X_h) be the fitness function for chromosome X_h in the component placement problem.

$$eval(X_h) = T\left\{c\left[N_{cc(t-1)}\right], f(a_t)\right\} + \sum_{i=a_t}^{N_{cc(1)}-1} T\left[f(i), f(i+1)\right] \qquad (1)$$

$$+T\left\{f\left[N_{cc(t)}\right], c(a_t)\right\} + \sum_{i=a_t}^{N_{cc(1)}-1} T\left[c(i), c(i+1)\right] \qquad (2)$$

for $t = 1, 2, \ldots, N_t$.

where

$T(a, b)$ is the longest traveling time for moving from point a to point b;

$$T(a,b) = \begin{cases} \max\left(\dfrac{|x_b - x_a|}{V_x}, \dfrac{|y_b - y_a|}{V_y}, t_r\right) & \text{if the head moves from a feeder to the other one} \\ & \text{or moves from a PCB's location to the other one,} \\ \max\left(\dfrac{|x_b - x_a|}{V_x}, \dfrac{|y_b - y_a|}{V_y}\right) & \text{otherwise.} \end{cases}$$

V_x and V_y are the speed of the revolver head in x and y directions, respectively;

t_r is the time for the revolver head to index one step;

$c(i)$ is the location on the PCB for the ith component;

$f(i)$ is the location of feeder for the ith component;

n is the number of components on a PCB;

N_n is the number of nozzles in a revolver head;

N_t is the number of assembly tours required for assembling all components;

$N_t = \text{ceil}(n / N_n)$;

t is the tour number;

$t = 1, 2, \ldots, N_t$;

$N_{c(t)}$ is the number of components collected and placed in tour t;

$N_{c(t)} = \min\,[N_n, N_{cr(t-1)}]$;

$N_{cr(t)}$ is the number of components remained after tour t;

$N_{cr(t)} = N_{cr(t-1)} - N_{c(t)}$;

$N_{cc(t)}$ is the cumulative number of components after tour t;

$N_{cc(t)} = N_{cc(t-1)} + N_{c(t)}$;

a_t is the first component in tour t;

$a_t = N_n\,(t - 1) + 1$;

$c(0)$ refers to the original location of the revolver head.

3.5. Selection

The roulette wheel method (Goldberg, 1989) is adopted to choose some chromosomes to undergo genetic operations. The approach is based on an observation that a roulette wheel has a section allocated for each chromosome in the population, and the size of each section is proportional to the chromosome's fitness. The fitter the chromosome, the higher the probability of being selected. Although one chromosome has the highest fitness, there is no guarantee it will be selected. The only thing that is certain is that on the average a chromosome will be chosen with the probability proportional to its fitness.

3.6. Genetic operations

The genetic operators used in the HGAs are one crossover and two mutations. Generally, GAs developed especially for the PCB assembly problems are not limited to using one crossover and one mutation (Leu et al., 1993). The two links in a chromosome are required to perform these genetic operations. Similar to that in the ISP, intra-tour and inter-tour moves will be occurred using the following genetic operations.

3.6.1. The order crossover

The crossover operator adopted in the HGAs is the classical order crossover operator (Gen and Cheng, 1997), and two offspring will be generated at each time. The procedure of the order crossover operation is listed as follows:

Step 1: Select a substring from the first parent randomly.

Step 2: Produce a protochild by copying the substring into the corresponding positions in the protochild.

Step 3: Delete those genes in the substring from the second parent. The resulted genes form a sequence.

Step 4: Place the genes into the unfilled positions of the protochild from the left to the right according to the resulted sequence of genes in Step 3 to produce an offspring.

Step 5: Repeat Steps 1 to 4 to produce another offspring by exchanging the two parents.

3.6.2. The heuristic mutation

A heuristic mutation (Gen and Cheng, 1997) is designed with the neighborhood technique to produce a better offspring. A set of chromosomes transformed from a parent by exchanging some genes is regarded as the neighborhood. Only the best one in the neighborhood is used as the offspring produced by the mutation. Herein, the original heuristic mutation is modified in order to promote diversity of the population. The modification is that all neighbors generated are used as the offspring. The procedure of the heuristic mutation operation is listed as follows:

Step 1: Pick up three genes in a parent at random.

Step 2: Generate neighbors for all possible permutations of the selected genes, and all neighbors generated are regarded as the offspring.

In Step 1, only three genes are selected since two genes have only one variation (one offspring) while more than three genes will generate too many offspring and it will take a very long time for computation.

3.6.3. The inversion mutation

The inversion operator (Gen and Cheng, 1997) selects a substring from a parent and flips it to form an offspring. The inversion operator, however, operates with one chromosome only. It is very similar to the heuristic mutation and thus lacks interchange of the characteristics between chromosomes. So, the inversion operator is a mutation operation, which is used to increase the diversity of the population rather than to enhance the quality of the population.

4. Result Analysis

In this section, two studies concerning the performance of the HGAs are carried out. First, the effect of the population size on the performance of HGAs is studied. Second, a computational study is made to compare the HGA1 and HGA2.

The performance of the HGAs is evaluated using a PCB example. The example has 50 components with 10 different component types. There is no solid instruction in selecting the crossover rate and mutation rate. A higher crossover rate allows exploration of more solution space and reduces the chances of setting for a false optimum; but if this rate is too high, it results in the wastage of a lot of computation time in exploring unpromising regions of the solution space. On the other hand, if the mutation rate is too low, many genes that would have been useful are never tried out. But if it is too high, there will be much random perturbation, the offspring will lose their resemblance to their parents, and the algorithm will lose the ability to learn from the history of search (Gen and Cheng, 1997). Here, a relatively high mutation rate is applied so that more new genes can be introduced into the population. The parameters of the HGA for the problem are: population size = 5 or 25, iteration number = 500, crossover rate = 0.4, and mutation rate = 0.2.

4.1. Effect of population size

Population size plays a vital role since it determines the efficiency and the effectiveness of the HGA. Larger population size means that more offspring is produced at each iteration. The more the offspring or search points, the higher the chance of finding the optimal solution or a better solution. Nevertheless, larger population size spends longer times for computation and takes more computer storage space. In order to identify the effect of the population size, HGA1 and HGA2 are run with the 50-component problem using two different population sizes: 5 and 25. Fig. 5 and Fig. 6 show the effect of population size on the performance of HGA1 and HGA2, respectively.

It can be seen from Fig. 5 and Fig. 6 that the curves representing the population size of 25 are lower than those representing the population size of 5. This phenomenon shows that the HGAs with larger population size can obtain better initial and final solutions. This may be due to the fact that more offspring are produced at each iteration. If the population size is 25, there are totally 40 offspring produced at each iteration. However, there are only 8 offspring generated at each iteration provided that the population size is 5. It can therefore be proved that the quality of the HGAs' solution is highly related to the population size since it determines the number of chromosomes generated.

4.2. HGA1 vs. HGA2

The result of comparison between HGA1 and HGA2 is summarized in Table 1, Fig. 7, and Fig. 8. It is found that the performance of HGA2 is superior to that of HGA1 in terms of the solutions' quality. First, the best initial solutions generated by HGA2 are much better than those generated by HGA1. With population size of 5, the best chromosomes in the initial population obtained by HGA1 and HGA2 are 107.75 seconds and 76.75 seconds, respectively. The percentage of improvement is about 29%. With population size of 25, the best initial chromosome obtained by HGA2 (73.6667 seconds) improves 31% when compared with that obtained by HGA1 (106 seconds). These significant improvements are due to the application of three heuristics, that is, the Clarke and Wright saving method, the nearest neighbor heuristic, and the neighborhood frequency heuristic in the initialization procedure. Second, and the most important, HGA2 generates better final solutions than HGA1: 64.6667 vs. 67.4167 seconds (population size = 5), and 60 vs. 60.1667 seconds (population size = 25). It is, therefore, suggested that HGA2 instead of HGA1 should be adopted to optimize the component placement problem for the CAP machine.

5. Conclusions

This paper developed two HGAs to solve the component grouping, component sequencing, and feeder arrangement problems simultaneously for the CAP machine with one revolver head, with the objective of minimizing the total placement time. It was proved that the algorithm hybridized with the Clarke and Wright saving method, the nearest neighbor heuristic, and the neighborhood frequency heuristic for the initialization procedure, and an improved heuristic called the iterated swap procedure can generate better initial and final solutions than the algorithm hybridized with the iterated swap procedure only.

In the intermediate future, there are three possible areas related to the component placement problem for the CAP machine can be investigated. First of all, a mathematical model should be formulated in order to obtain the global optimal solutions to the problem. Once it is obtained, the effectiveness of the proposed algorithms can be known. Second, in the component placement problem, the number of feeders supplied is exactly the same as the number of component types required. In the real-world situations, the number of feeders available may be greater. Components of the same type can, therefore, be stored in more than one feeder. In this case, the feeder arrangement problem no longer belongs to the category of 1-to-1 assignment problem. Also, it is necessary to determine which feeder a component should be retrieved from since more than one feeder hold the same type of components, that is, the component retrieval problem. Third, the component placement problem for the CAP machine with two or even more revolver heads can also be studied. The HGAs developed in this paper can be easily modified to tackle this problem.

Acknowledgements

The authors wish to acknowledge The Hong Kong Polytechnic University for the financial support of the project (No. A-PG05).

References

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

Ball, M. O. and Magazine, M. J. (1988), "Sequencing of insertions in printed circuit board assembly", *Operations Research*, Vol. 36, pp. 192-201.

Bard, J. F., Clayton, R. W. and Feo, T. A. (1994), "Machine setup and component placement in printed circuit board assembly", *International Journal of Flexible Manufacturing Systems*, Vol. 6, pp. 5-31.

Broad, K., Mason, A., Rönnqvist, M. and Frater, M. (1996), "Optimal robotic component placement", *Journal of Operational Research Society*, Vol. 47, pp. 1343-1354.

Clarke, G. and Wright, J. W. (1964), "Scheduling of vehicles from a central depot to a number of delivery points", *Operations Research*, Vol. 12, pp. 568-581.

Crama, Y., Flippo, O. E., Klundert, J. V. D. and Spieksma, F. C. R. (1997), "The assembly of printed circuit boards: a case with multiple machines and multiple board types", *European Journal of Operational Research*, Vol. 98, pp. 457-472.

Csaszar, P., Tirpak, T. M. and Nelson, P. C. (2000), "Optimization of a high-speed placement machine using tabu search algorithms", *Annals of Operations Research*, Vol. 96, pp. 125-147.

Dikos, A., Nelson, P. C., Tirpak, T. M. and Wang, W. (1997), "Optimization of high-mix printed circuit card assembly using genetic algorithms", *Annals of Operations Research*, Vol. 75, pp. 303-324.

Ellis, K. P., Vittes, F. J. and Kobza, J. E. (2001), "Optimizing the performance of a surface mount placement machine", *IEEE Transactions on Electronics Packaging Manufacturing*, Vol. 24, pp. 160-170.

Foulds, L. R. and Hamacher, H. W. (1993), "Optimal bin location and sequencing in printed circuit board assembly", *European Journal of Operational Research*, Vol. 66, pp. 279-290.

Francis, R. L., Hamacher, H. W., Lee, C. Y. and Yeralan, S. (1994), "Finding placement sequences and bin locations for cartesian robots", *IIE Transactions*, Vol. 26, pp. 47-59.

Gen, M. and Cheng, R. (1997), *Genetic Algorithms and Engineering Design*, Wiley, New York.

Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, New York.

Grunow, M., Günther, H. O., Schleusener, M. and Yilmaz, I. O. (2004), "Operations planning for collect-and-place machines in PCB assembly", *Computers & Industrial Engineering*, Vol. 47, pp. 409-429.

Ho, W. and Ji, P. (2003), "Component scheduling for chip shooter machines: a hybrid genetic algorithm approach", *Computers & Operations Research*, Vol. 30, pp. 2175-2189.

Ho, W. and Ji, P. (2004), "A hybrid genetic algorithm for component sequencing and feeder arrangement", *Journal of Intelligent Manufacturing*, Vol. 15, pp. 307-315.

Ho, W. and Ji, P. (2005), "A genetic algorithm to optimise the component placement process in PCB assembly", *International Journal of Advanced Manufacturing Technology*, Vol. 26, pp. 1397-1401.

Ho, W. and Ji, P. (2006), "A genetic algorithm approach to optimising component placement and retrieval sequence for chip shooter machines", *International Journal of Advanced Manufacturing Technology*, Vol. 28, pp. 556-560.

Ji, Z., Leu, M. C. and Wong, H. (1992), "Application of linear assignment model for planning of robotic printed circuit board assembly", *Journal of Electronic Packaging*, Vol. 114, pp. 455-460.

Kazaz, B. and Altinkemer, K. (2003), "Optimization of multi-feeder (depot) printed circuit board manufacturing with error guarantees", *European Journal of Operational Research*, Vol. 150, pp. 370-394.

Klomp, C., Klundert, J. V. D., Spieksma, F. C. R. and Voogt, S. (2000), "The feeder rack assignment problem in PCB assembly: a case study", *International Journal of Production Economics*, Vol. 64, pp. 399-407.

Kumar, R. and Li, H. (1995), "Integer programming approach to printed circuit board assembly time optimization", *IEEE Transactions on Components, Packaging, and Manufacturing Technology – Part B*, Vol. 18, pp. 720-727.

Leu, M. C., Wong, H. and Ji, Z. (1993), "Planning of component placement/insertion sequence and feeder setup in PCB assembly using genetic algorithm", *Journal of Electronic Packaging*, Vol. 115, pp. 424-432.

Magyar, G., Johnsson, M. and Nevalainen, O. (1999), "On solving single machine optimization problems in electronics assembly", *Journal of Electronics Manufacturing*, Vol. 9, pp. 249-267.

Moyer, L. K. and Gupta, S. M. (1996a), "SMT feeder slot assignment for predetermined component placement paths", *Journal of Electronics Manufacturing*, Vol. 6, pp. 173-192.

Moyer, L. K. and Gupta, S.M. (1996b), "Simultaneous component sequencing and feeder assignment for high speed chip shooter machines", *Journal of Electronics Manufacturing*, Vol. 6, pp. 271-305.

Moyer, L. K. and Gupta, S. M. (1997), "An efficient assembly sequencing heuristic for printed circuit board configurations", *Journal of Electronics Manufacturing*, Vol. 7, pp. 143-160.

Ong, N. S. and Khoo, L. P. (1999), "Genetic algorithm approach in PCB assembly", *Integrated Manufacturing Systems*, Vol. 10, pp. 256-265.

Ong, N. S. and Tan, W. C. (2002), "Sequence placement planning for high-speed PCB assembly machine", *Integrated Manufacturing Systems*, Vol. 13, pp. 35-46.

Osman, I. H. and Kelly, J. P. (1996), *Meta-heuristics: Theory & Applications*, Kluwer Academic Publishers, Boston.

Reinelt, G. (1994), *The Traveling Salesman: Computational Solutions for TSP Applications*, Springer-Verlag, New York.

SIEMENS (2005), http://www.siplace.com.

Sohn, J. and Park, S. (1996), "Efficient operation of a surface mounting machine with a multihead turret", *International Journal of Production Research*, Vol. 34, pp. 1131-1143.

Souza, R. D. and Wu, L. J. (1995), "Intelligent optimization of component onsertion in multi-head concurrent operation PCBA machines", *Journal of Intelligent Manufacturing*, Vol. 6, pp. 235-243.

Wilhelm, W. E. and Tarmy, P. K. (2003), "Circuit card assembly on tandem turret-type placement machines", *IIE Transactions*, Vol. 35, pp. 627-645.

Yeo, S. H., Low, C. W. and Yong, K. H. (1996), "A rule-based frame system for concurrent assembly machines", *International Journal of Advanced Manufacturing Technology*, Vol. 12, pp. 370-376.

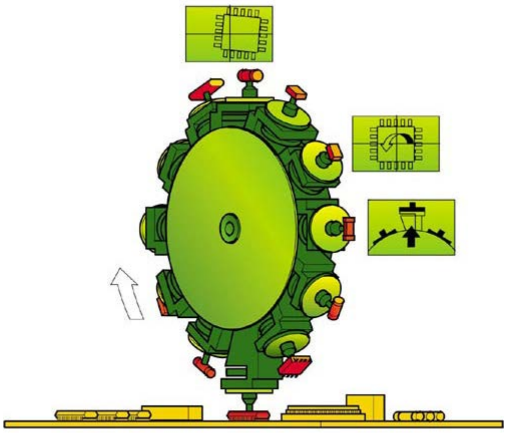

Fig. 1. The schematic diagram of the collect-and-place machine (SIEMENS, 2005).

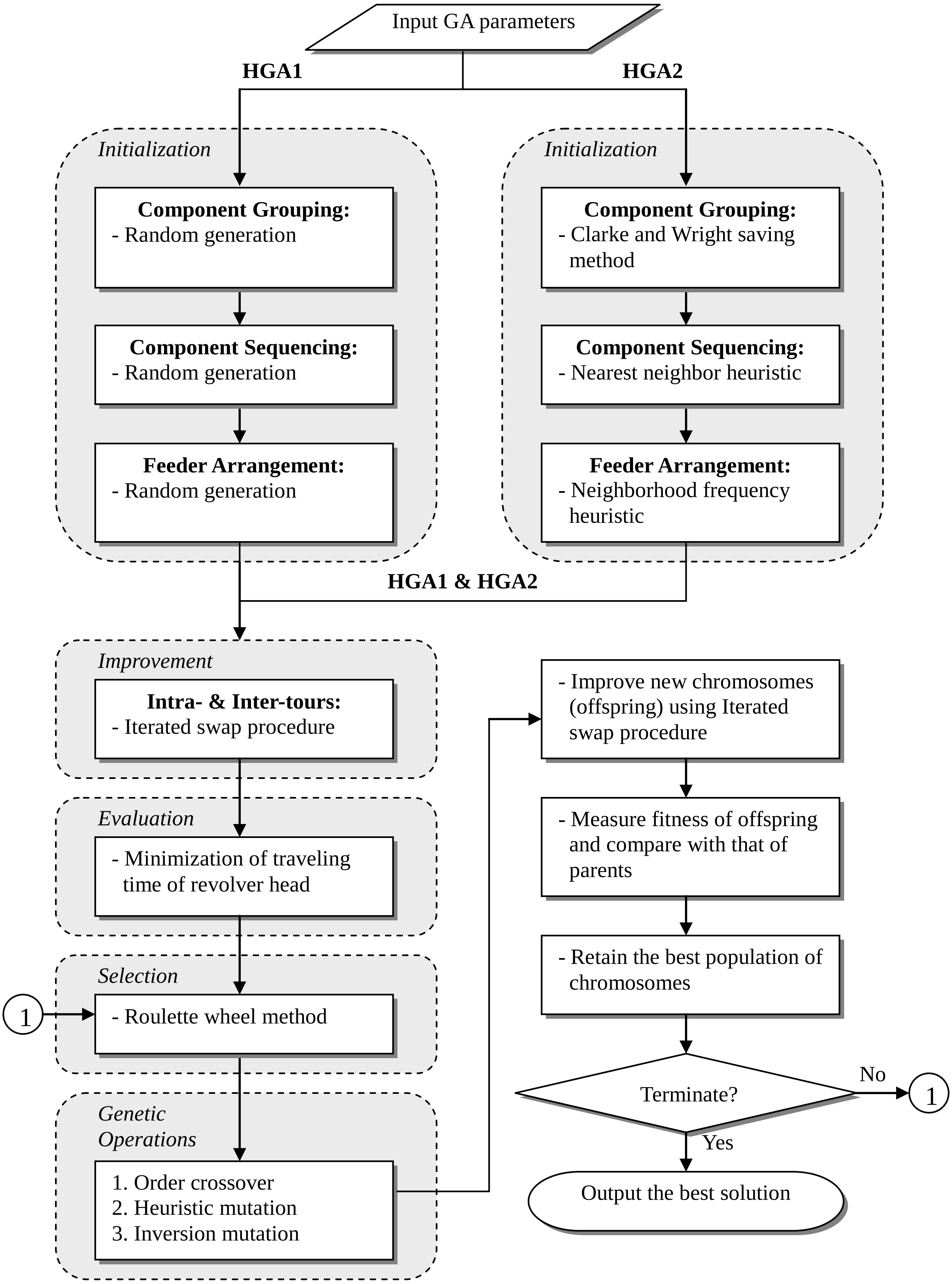

Fig. 2. The flowchart of the HGAs.

Assembly Sequence	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Component Number	1 (4)	2 (3)	5 (4)	10 (1)	4 (2)	9 (1)	7 (4)	3 (1)	6 (3)	8 (2)

Link 1

Component Type	3	4	1	2
Feeder	**1**	**2**	**3**	**4**

Link 2

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

Select 2 genes randomly

Parent:	1	2	$\boxed{5}$	10	4	9	7	$\boxed{3}$	6	8
Offspring 1:	1	2	**3**	10	4	9	7	**5**	6	8
Offspring 2:	1	**3**	**2**	10	4	9	7	5	6	8
Offspring 3:	1	2	**10**	**3**	4	9	7	5	6	8
Offspring 4:	1	2	3	10	4	9	**5**	**7**	6	8
Offspring 5:	1	2	3	10	4	9	7	**6**	**5**	8

Fig. 4. The iterated swap procedure.

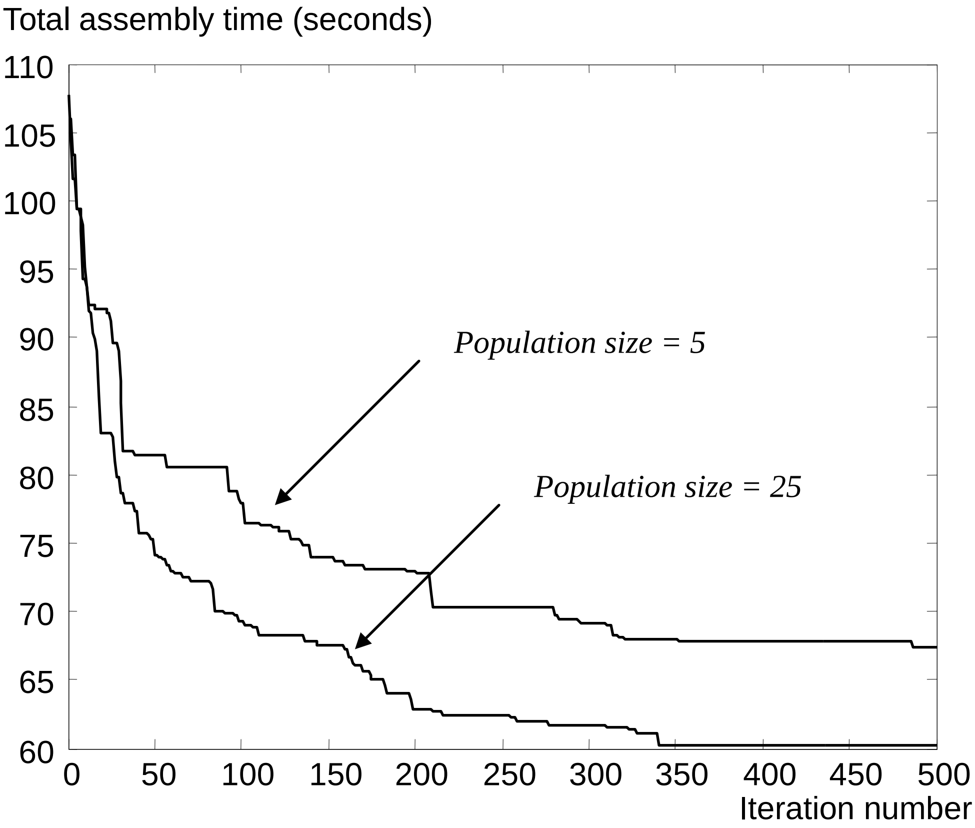

Fig. 5. The effect of population size on the performance of HGA1.

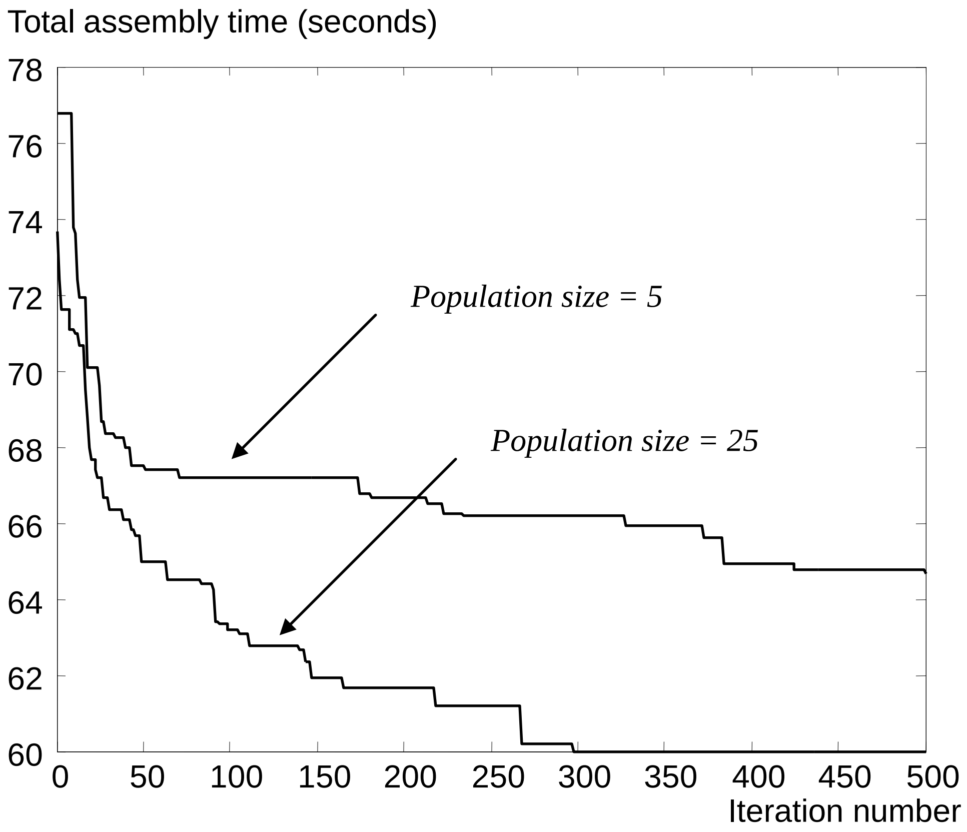

Fig. 6 The effect of population size on the performance of HGA2.

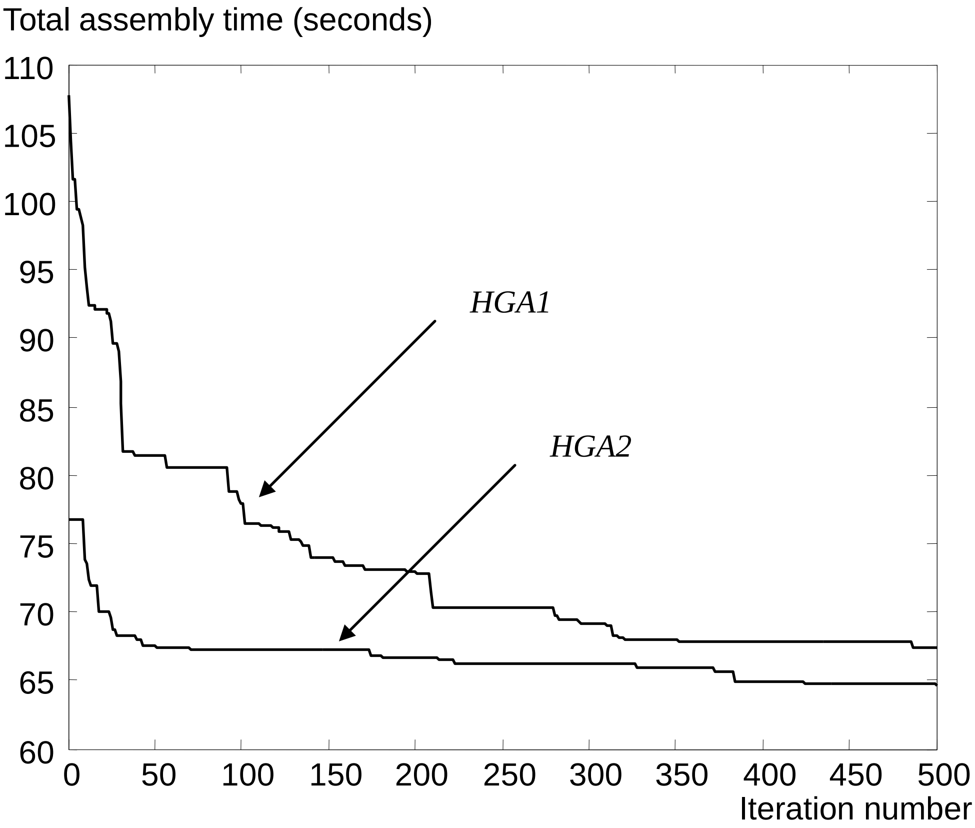

Fig. 7. Comparison of HGA1 and HGA2 with population size of 5.

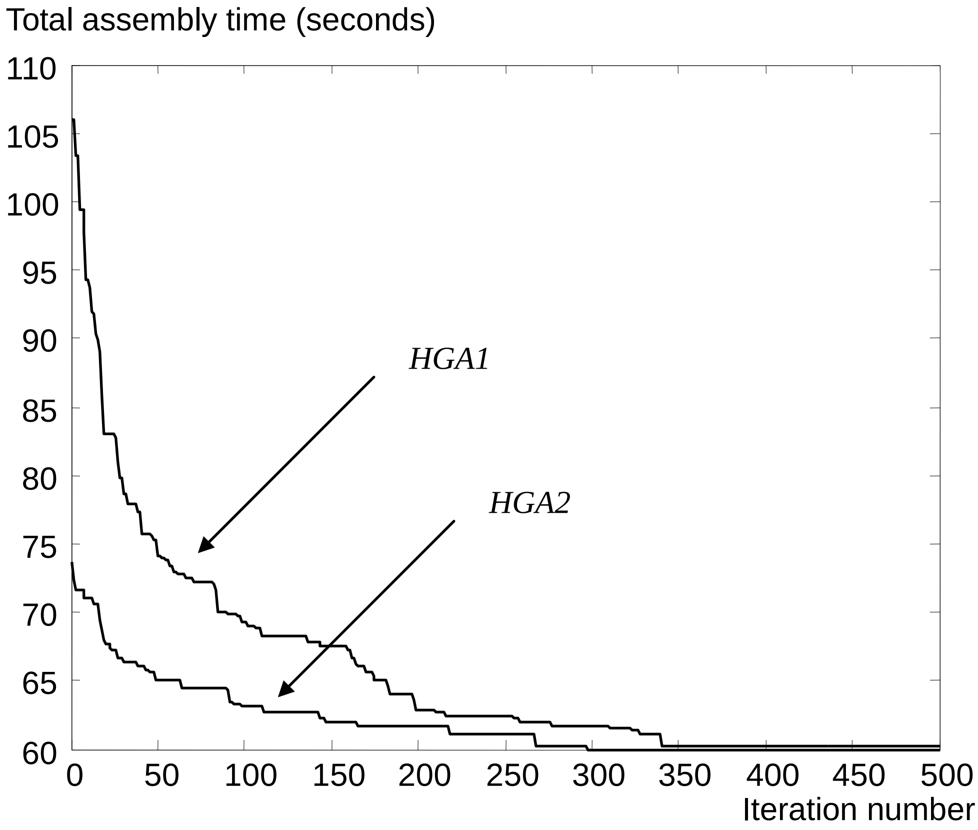

Fig. 8. Comparison of HGA1 and HGA2 with population size of 25.

Table 1

A comparison of the experimental results

	HGA1		**HGA2**	
Population size	5	25	5	25
Best one in the initial population (seconds)	107.7500	106.0000	76.7500	73.6667
Iteration number	487	340	500	298
Final best solution (seconds)	67.4167	60.1667	64.6667	60.0000

Author/s:
Ho, W; Ji, P; Dey, PK

Title:
Optimization of PCB component placements for the collect-and-place machines

Date:
2008-06-01

Citation:
Ho, W., Ji, P. & Dey, P. K. (2008). Optimization of PCB component placements for the collect-and-place machines. INTERNATIONAL JOURNAL OF ADVANCED MANUFACTURING TECHNOLOGY, 37 (7-8), pp.828-836. https://doi.org/10.1007/s00170-007-1014-z.

Persistent Link:
http://hdl.handle.net/11343/118685

File Description:
Accepted version