

## Heuristic Algorithms for the Path Planning Problem in Three-Dimensional Automated Optical Inspection (AOI) Machines

Geun-Cheol Lee<sup>1</sup>, SungKuk Choi<sup>2</sup>, DeokHwa Hong<sup>3</sup> and Seong-Hoon Choi<sup>4\*</sup>

<sup>1</sup>Konkuk University, College of Business Administration, South Korea

<sup>2</sup>InterClone Corporation, South Korea

<sup>3</sup>Koh Young Technology, R&D/Vision Engineer Group, South Korea

<sup>4</sup>Sangmyung University, Department of Management Engineering, South Korea

\*shchoi@smu.ac.kr

### Abstract

*In this study, the path planning problem in 3D automated optical inspection (AOI) machines is considered. In the problem, the primary objective is finding a path that minimizes the total working time among all possible paths for the camera. In 3D AOI machines, a very precise level of image processing has to be performed when inspecting the mounted components on printed circuit boards (PCBs). Therefore, both the routing time of the camera path and the image processing time have to be considered when calculating the total working time of the camera in the AOI machines. In this paper, we propose efficient heuristic algorithms that can find the paths that have the shortest total working times. The performances of the proposed methods primarily depend on the NEH-based procedure, which is one of our proposed improvement procedures. We introduce several construction and improvement procedures, and select the best combination as our proposed approach. The results of computational experiments conducted on real data to evaluate the performance of our proposed approach show that our proposed methods improve on the original path planning performances. In addition, tests done on randomly generated data indicate that our proposed approach is superior to other benchmark approaches, including metaheuristic algorithms, in terms of computing time and performance measures.*

**Keywords:** *automated optical inspection, path planning, heuristics, simulated annealing*

### 1. Introduction

Printed circuit boards (PCBs) have formed the core of most electronic products in the past several decades. While the sizes of PCBs have decreased significantly, the role of PCBs in electronics has not diminished. Cutting-edge technologies, such as surface mounting technology (SMT), have facilitated the miniaturization of PCBs. Along with the newest technologies, PCB assembly-line systems have been improved with automated process machines. One of the machines in the spotlight is the automated optical inspection (AOI) machine, which has replaced the traditional manual visual inspection of PCBs [1]. Inspection by machine or robot vision is a current trend of technology in other inspection fields [2, 3].

Using AOI machines, inspections to ensure that every component on a PCB is properly mounted are conducted quickly. However, as the number of components mounted on a PCB increases, the total working time needed to inspect all the components on the board rises as well. Traditionally, the total inspection time mainly depended on the path length of the inspection camera. Therefore, to reduce the total inspection time, the length of the camera route that covers all the components should be minimized. However, images acquired using

traditional 2D AOI machines are not precise enough to detect defects accurately. To deal with these vulnerabilities of 2D AOI machines, in 3D AOI machines, a more precise level of image processing is performed when the mounted components on PCBs are being inspected. Consequently, to plan the path of the inspection camera, both the length of the path and the image processing time that occurs after each camera shot have to be considered. Figure 1 shows a typical 3D AOI machine [4].



**Figure 1. A picture of a typical 3D AOI machine**

The path planning problem, which affects AOI machines as well as component mounting machines, has been studied by many researchers. In a recent survey paper on vision robotics, the path planning problem was introduced as one of the productivity improvement issues in robotic manipulations [5]. Recently, Wu *et al.*, [1] proposed a path planning method based on the Hopfield net algorithm, whereas most other studies on AOI path planning problems proposed metaheuristics, such as genetic algorithms and ant colony algorithms [6, 7, 8, 9]. They primarily consider traveling salesman problem (TSP)-based methods to construct a shortest route. Further, Park *et al.*, [8, 9] considered combined problems in which sequencing as well as clustering problems needed to be solved. Along with the TSP, which is already known as a non-tractable problem [10]; additional issues such as clustering make the problem even more difficult, which has led many researchers to use metaheuristics as their only tool. Different types of the path planning problems can be found in other studies on robot manipulations [11, 12].

In this study, we consider the path planning problem in 3D AOI machines. This differentiates it from the existing studies mentioned above, all of which focus on 2D AOI machines. The remainder of this paper is structured as follows. In Section 2, we describe the problem being considered in detail. In Section 3, we present our proposed efficient heuristic methods for the problem. To evaluate the performance of our proposed methods, we compare them with other methods, including metaheuristic methods, in terms of computational experimental results; these results are presented in Section 4. We conclude this paper with a summary and discussions in Section 5.

## **2. Problem Description**

### **2.1. Path planning problem**

A typical AOI machine maneuvers its camera to capture stills, and processes the images of the components mounted on a PCB. Unfortunately, the size of each image

captured by the camera is not large enough to cover the entire PCB. In fact, the area acquired by the camera, called the field of view (FOV), is very small compared to the size of the PCB. Thus, several hundred FOVs may be needed to inspect the entire PCB. Inspection of a PCB is completed when all the FOVs in the PCBs are shot by the camera and the images processed on the AOI machine. Consequently, the sequence in which the FOVs to be taken are shot has to be decided in order to minimize the total working time for PCB inspection on the 3D AOI machine. In this study, we assume that FOVs are given for PCBs.

The objective function of the problem is minimization of the total working time of the inspection, which includes the move time and the image processing time. In classic path planning problems, the move time is the major factor comprising the total working time because 2D image processing is not as sophisticated as 3D image processing. Thus, for 2D AOI path planning, TSP-based algorithms were mainly used to find the shortest route that covers all the given FOVs. However, in 3D AOI machines, advanced image processing techniques are applied, which force us to consider the time to process the acquired images, as well as the move time. That is, the traditional TSP-based algorithms are not enough for the 3D AOI path planning problem.

As mentioned earlier, using 3D measurements, very precise images of components can be acquired; however, this requires more processing power than 2D AOI machines. To resolve this problem, 3D AOI machines have multiple CPU processors that can process 3D images simultaneously. Thus, multiple FOVs acquired by the camera can be processed at the same time even while the camera is moving or capturing images of other FOVs. In this paper, we consider a 3D AOI machine that has eight parallel processors, which means that it can simultaneously process the images of eight FOVs.

Although minimization of the total working time is the primary objective of the problem, from a practical point of view, the total time spent moving also needs to be minimized. A path with minimized total working time can have unreasonable movements, such as the revisiting of certain areas repeatedly; therefore, a path that has both minimized total working time as well as reasonable shape needs to be found. In this paper, we use the objective function as a weighted equation that considers both factors, *i.e.*, (total working time) +  $\alpha$  × (total move time). Using parameter  $\alpha$ , we find a path as short as the shortest route with a slight sacrifice in the total working time.

## 2.2. Solution representation

Because there are multiple processors to conduct the image processing, it may be thought that the specific processor to be used for the image processing of a specific FOV must be decided. However, if we simply decide on the sequence of processing, then each FOV can be automatically assigned to the earliest available processor. Furthermore, two operations for each FOV, *i.e.*, shooting and image processing, have the same sequence. Consequently, the permutation sequence of the given FOVs is the only thing that we need to decide for the problem, which can be represented with a single vector.

For example, in a scenario where there are five FOVs for a PCB, (1, 4, 3, 2, 5) may be the vector representation of a permutation sequence, which means that FOV #1 was shot first, then FOV #4 second, and so on. In addition, FOV #1 is processed first, then FOV #4, and so on. That is, the shooting sequence and the image processing sequence are the same, which leads us to use the permutation sequence to represent the solution to the problem under consideration.

### 3. Proposed Methods

In our proposed methods, final solutions (sequence of FOVs) are obtained in two steps: initial solutions are obtained by a couple of construction procedures, and the solutions obtained are further improved using several improvement procedures. Finally, we select several of the best combinations of the construction and improvement procedures.

#### 3.1. Construction procedures

We think of the first way of constructing a solution as random, and it can be a benchmark for any other method. In this paper, this method is abbreviated as RAND. Focusing on the image processing time is the second way to construct a solution. As mentioned earlier, 3D AOI machines characteristically have larger image processing times and multiple CPUs to process the image. Therefore, it is better to process the FOV with the largest image processing time first. Thus, in the second construction procedure, the FOV sequence is constructed in such a way that the FOVs are aligned in descending order of their image processing times. Henceforth, this procedure is abbreviated as IPAO.

#### 3.2. Improvement procedures

Once a solution (FOV sequence) is given, it can be improved upon by modifying a part of the solution. We propose three improvement procedures to enhance the constructed solution obtained by either RAND or IPAO.

##### NEH-based approach

The NEH method introduced by Nawaz *et al.*, [13] is known as one of the most efficient and effective heuristics in scheduling areas, resulting in the paper being cited more than 1,000 times by other researchers. The method was originally devised for flowshop scheduling problems; however, it has since been modified for many other extended problems, such as job shop and hybrid flowshop scheduling.

Using the NEH, FOV sequences given by RAND or IPAO can be significantly improved. The details of the improvement procedure are given below. In the procedure,  $S$  and  $U$  denote the set of FOVs for which sequences are decided and not decided, respectively.

Step 0: Let  $U$  be a given FOV sequence.

Step 1: If  $U$  is empty, go to Step 4; otherwise, select the first FOV in  $U$ .

Step 2: Put the selected FOV at every available position in  $S$  and evaluate the corresponding objective function, i.e., (total working time) +  $\alpha$  × (total move time).

Step 3: Fix the position of the selected FOV in  $S$  where the best objective function value is returned. Delete the selected FOV in  $U$  and go to Step 1.

Step 4:  $S$  is the final FOV sequence.

In Step 2, when the objective function value of sequence  $S$  is estimated, the undecided part of  $S$  is replaced with the corresponding part of  $U$ . Through this

procedure, a given FOV sequence can be improved upon in terms of the objective function. This procedure is denoted as NEH for now.

#### Pair-wise interchange (PI)

The simplest way to modify a given sequence is to exchange the positions of two FOVs in the sequence. The result of the interchange is different according to the pair selected. In this study, we use the following procedure to select the pair and improve the given sequence. (In the procedure,  $M$  is the parameter that designates the maximum number of iterations.)

- Step 0: Iteration := 0. The given FOV sequence is the current sequence.
- Step 1: Select an FOV arbitrarily from among the FOVs loaded on the CPU with the greatest load using the current sequence.
- Step 2: Select an FOV arbitrarily from those loaded on CPUs apart from the one on which the first FOV selected was loaded.
- Step 3: Change the positions of the two selected FOVs and evaluate the modified sequence. If the objective value is improved, the modified sequence is the current sequence.
- Step 4: If Iteration =  $M$ , then the current sequence is the final FOV sequence—STOP; otherwise, Iteration := Iteration + 1—go to Step 1.

#### NEH insertion (NI)

Traditional insertion is an improvement procedure in which a job from a given sequence is selected and placed arbitrarily in another position in the given sequence. In our study, we use insertion based on the NEH approach. That is, we arbitrarily select an FOV from a given sequence and find the new position that returns the smallest objective function value. The procedure is then repeated according to the pre-specified maximum iteration number.

### **3.3. Proposed combinations**

As outlined above, we propose two construction methods and three improvement methods for the problem. Further, the improvement methods can be applied multiple times, which results in a large number of possible combinations that approach the solution to the problem. For example, “RAND→NEH→PI(100)→NI(100)” means that the initial solution is obtained with RAND, and then improved by the application of NEH, PI(100), and NI(100) sequentially, where the numbers in the parenthesis next to PI and NI are the maximum numbers of iterations.

After preliminary computational experiments, we selected the following combinations considering performance and computing time: IPAO→NEH→PI(1000)→NI(100). Of the two construction procedures, IPAO gave the best performance. Further, with the same improvement procedure, the performances of RAND and IPAO differed significantly. Among the improvement procedures, NEH is the core process that primarily determines the performance of the overall procedure.

## 4. Computational Experiments

To evaluate the performance of the proposed methods, we conducted a series of computational experiments. We implemented a simulated annealing (SA) method [14], one of the most well-known metaheuristic methods and known to be relatively faster than other metaheuristics, as a benchmark method. The proposed methods as well as the SA method were implemented in the C language, and all the tests conducted on a 3.20-GHz i5 CPU PC with 3 GB of RAM.

### 4.1. Test results for real data

The problem considered in this study is a real problem that arises in a Korean AOI machine manufacturing company. The company is focused on reducing the total working time needed to inspect various PCBs from various customers. Originally, the company used a TSP-based method to plan the FOV path. We tested several sets of real data to compare their current method with our proposed methods. Table 1 shows the results of the tests conducted on the real data. The numbers displayed in the table are the total working times for the corresponding methods. Confidentiality issues in the company restricted the number of problems that could be tested to four.

**Table 1. Test results for real data**

Data	The original method	The proposed methods			
		RAND	IPAO	RAND→NEH	IPAO→NEH
RD01	18.3*	19.6	19.2	17.8	16.9
RD02	25.6	29.5	22.8	26.8	20.9
RD03	26.1	28.7	22.5	25.3	19.8
RD04	38.4	48.7	46.7	37.1	35.4

\* Total working times (seconds)

As expected, RAND did not show any superiority over the other methods. In addition, IPAO and RAND→NEH did not dominate the current method. However, IPAO→NEH outperformed the current method as well as the other proposed methods. We presume that more enhanced solutions can be found by applying other improvement procedures several times. However, no more procedures were applied to the solutions because the tested problems were rather small in size, *i.e.*, less than 100 FOVs in each PCB. The computation times of all the proposed methods in this test were less than 0.1 s, which is primarily due to the small number of FOVs in the tested data.

### 4.2. Test results for generated data

Because the test conducted on the real data was very limited, we generated test problems that had various FOV sizes. The test problems generated arbitrarily based on the real FOV data. We considered five levels of FOV sizes—10, 20, 50, 100, and 200—and each instance had 60 replications. That is, a total of 300 problem instances were generated and tested. The overall test results obtained are summarized in Table 2.

First, the selected combination—IPAO→NEH→PI(1000)→NI(100)—as a matter of course always improved the solution by IPAO→NEH. The extent of the improvement increased with the size of the FOV. To aid in the evaluation of the performance of our proposed methods, test results for a metaheuristic, *i.e.*, SA in this study, are also shown in the table. SA performed better than our proposed methods on the small problems. However, the

overall performances of the selected combination across the problem sizes were better than that of SA considering both the weighted objective function values and the computation times. Even in the largest problem size, the selected combination gave the best performances over SA within 10 s.

**Table 2. Test results for generated data by the number of FOVs**

	IPAO→NEH			IPAO→NEH→PI(1000)→NI(100)			SA		
	0.5 <sup>#</sup>	1	2	0.5	1	2	0.5	1	2
10	13.1 <sup>*</sup>	13.2	13.5	12.7	12.8	13.1	12.7	12.8	12.9
	<0.01 <sup>**</sup>	<0.01	<0.01	0.01	0.01	0.01	1.94	1.70	1.64
20	19.9	20.0	20.5	19.1	19.2	19.6	18.7	19.0	19.2
	<0.01	<0.01	<0.01	0.04	0.04	0.04	0.12	0.13	0.12
50	41.8	41.8	42.4	40.3	40.1	40.3	40.5	40.5	41.0
	0.01	0.01	0.01	0.31	0.30	0.30	0.73	0.72	0.71
100	77.9	77.9	77.7	75.5	75.4	75.5	77.4	77.7	77.7
	0.04	0.04	0.04	1.60	1.59	1.60	3.04	2.99	2.96
200	149.5	148.7	148.9	145.9	145.6	145.9	152.4	152.5	152.6
	0.29	0.29	0.30	9.29	9.34	9.32	12.47	12.35	12.21

\* Average objective function values, i.e., total working times +  $\alpha$  × total move times (seconds)

\*\* Computation times (seconds)

# Value of parameter  $\alpha$

## 5. Conclusion

In this paper, we proposed NEH-based heuristic methods for the path planning problem in 3D AOI machines. Unlike traditional 2D AOI machines, the inspection times in 3D AOI machines are larger because of the very precise image-processing of components. In our study, we transformed the path planning problem into a scheduling problem, for which the permutation sequence of the FOVs had to be determined. Most of the existing methods used to solve the problem, such as genetic algorithms and ant colony algorithms, are metaheuristic. We argue that our proposed methods are so efficient that the solutions obtained are much faster than those obtained by metaheuristic methods. Further, the quality of the solutions obtained by our proposed methods is as good as that of a metaheuristic, in terms of the total working time. The results of a series of computational experiments and comparisons with other heuristics and a metaheuristic method indicate that our proposed methods are superior.

Our study needs to be extended to integrated problems that combine both sequencing and clustering of FOVs. In this study, FOVs were pre-given, so the clustering problem was not considered. However, in the real world, clustering is one of the major issues in AOI machine the path planning decisions. An integrated approach based on genetic algorithms was proposed by Park and Kim [8]; however, more efficient integrated algorithms are needed for path planning in 3D AOI machines. The algorithms proposed in this paper can form the basis for such integrated algorithms.

## Acknowledgements

This study was supported by a grant from InterClone Corporation and Koh Young Technology in 2012.

## References

- [1] H. Wu, Y. Kuang, G. Ouyang, H. Xie and X. Zhang, "Optimize position and path planning of automated optical inspection", *Journal of Computational Information Systems*, vol. 8, no. 7, (2012), pp. 2957-2963.
- [2] M. H. Dehrouyeh, M. Omid, H. Ahmadi, S. S. Mohtasebi and M. Jamza, "Grading and Quality Inspection of Defected Eggs Using Machine Vision", *International Journal of Advanced Science and Technology*, vol. 17 (2010), pp. 23-30.
- [3] Teutsch, C. D. Berndt, M. Hubner and N. Bachfischer, "Optical Inspection of Laser Markings in the Production Process", *International Journal of Advanced Science and Technology*, vol. 21, (2010), pp. 31-40.
- [4] Zenith 3D AOI system, <http://www.kohyoung.co.kr/>.
- [5] S. Chen, Y. Li and N. M. Kwok, "Active vision in robotic systems: a survey of recent developments", *The International Journal of Robotics Research*, vol. 30, no. 11, (2011), pp. 1343-1377.
- [6] B. Ren and L. Cheng, "SMT automatic optical inspection path planning based on MDSPSO algorithm", *Proceedings of the International Conference on Computational Intelligence and Natural Computing (CINC'09)*, vol. 2, (2009), pp. 134-137.
- [7] B. Luo and Y. Zhang, "SMT automatic optical inspection path planning based on ant colony algorithm", *Chinese Journal of Scientific Instrument*, vol. S2, (2006), pp.1251-1252, 1278.
- [8] T. -H. Park, H. -J. Kim and N. Kim, "Path planning of automated optical inspection machines for PCB assembly systems", *International Journal of Control, Automation, and Systems*, vol. 4, no. 1, (2006), pp. 96-104.
- [9] T. -H. Park, H. -J. Kim, "Path planning of automatic optical inspection machines for PCB assembly systems", *Proceedings 2005 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, (2005) June 27-30, Espoo, Finland, pp. 249-254.
- [10] M. Bellmore and G. Nemhauser, "The traveling-salesman problem: a survey", *Operation Research*, vol. 16, (1968), pp. 538-558.
- [11] G. Wang, L. Guo, H. Duan, L. Liu and H. Wang, "A Modified Firefly Algorithm for UCAV Path Planning", *International Journal of Hybrid Information Technology*, vol. 5, no. 3, (2012), pp. 123-144.
- [12] Q. Song and L. Liu, "Mobile Robot Path Planning Based on Dynamic Fuzzy Artificial Potential Field Method", *International Journal of Hybrid Information Technology*, vol. 5, no. 4, (2012), pp. 85-94.
- [13] M. Nawaz, E. Enscore and I. Ham, "A heuristic algorithm for the  $m$ -machine,  $n$ -job flow shop sequence problem", *OMEGA-The International Journal of Management Science*, vol. 11, no. 1, (1983), pp. 91-95.
- [14] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, "Optimization by Simulated Annealing", *Science*, vol. 220, no. 4598, (1983), pp. 671-680.