A MATHEMATICAL PROGRAMMING MODEL FOR OPTIMAL CORRECTION OF JIG POSITION IN PCB INSPECTIONS AND ITS NATURE-INSPIRED SOLUTION ALGORITHMS

Hongwei Wu and Hideki Katagiri

Received November 12 2016; revised November 25 2016

Abstract. Pin probe inspection methods have been widely used in printed circuit board electrical inspection. Due to the miniaturization of electronic devices, the positioning of inspection jig (called probe jig) is very important for precisely conducting pattern tests of wiring on PCBs. This article newly presents a mathematical programming approach to the optimal position correction of a probe jig. The optimal position correction problem is formulated based on nonlinear programming problem. Several nature-inspired algorithms, such as firefly algorithm (FA), bat algorithm (BA), cuckoo search (CS) and flower pollination algorithm (FPA), are developed for obtaining optimal position correction. The position correction of a probe jig is attained through interactive processes between an operator and the system. The performances of the integrated bio-inspired algorithms are compared. It is shown that the FPA is best among four nature-inspired algorithms for correcting a probe jig position in PCB inspection.

1 Introduction PCBs are used in almost all electronic products. A printed circuit board (PCB) mechanically supports and electrically connects electronic components using conductive tracks, pads and other features etched from copper sheets laminated onto a non-conductive substrate. There are many of previous studies on PCB manufacturing [1, 4, 5].

The PCB inspection is a very important process in PCB manufacturing in order to enhance the reliability of PCBs. In production process of PCBs, various wiring patterns are etched on PCBs. When a certain problem happens in forming wiring patterns, defect generation may arise and often make PCBs work irregularly. Hence, PCB inspections must be done through pattern check of wiring.

In this paper, we focus on the task of setting probe jigs which is considered as one of the most burdensome tasks in PCB inspections, because the position of a probe jig is corrected bit by bit manually in the field; it takes skilled workers several hours to complete the task, which lowers the effectiveness of PCB inspections. Hence, an automatic setting method has been needed in order to streamline PCB inspection.

In order to decrease the setup time of a probe jig on an electric inspection machine, we propose optimization techniques based on nonlinear programming and nature-inspired metaheuristic algorithms. Recently, nature-inspired metaheuristic algorithms, such as firefly algorithm (FA) [7], cuckoo search (CS) [10], bat algorithm (BA) [8], flower pollination algorithm (FPA) [9], have drawn widely attention from many researchers. The objective of this article is to develop a probe jig correction method in which the problem to be solved is formulated as a nonlinear programming problem and solution methods based on FA, CS, BA and FPA are developed. The performances of the four nature-inspired metaheuristics are compared.

2010 *Mathematics Subject Classification.* Primary 90B30, 90C11; Secondary 90B25, 90B10.

Key words and phrases. Printed circuit board (PCB), mathematical programming, bio-inspired metaheuristics, electrical inspection, wiring pattern test.

In the previous studies on PCB manufacturing, there were some studies on optimization approaches to PCB assembly [3, 6]. On the other hand, optimization techniques for PCB inspections have not been sufficiently discussed so far. This article tackles one of the most challenging problems in PCB manufacturing.

This paper is organized as follows: In Section 2, we review the existing method of correcting probe jig positions and discuss inefficiencies of the existing method. Section 3 proposes a computational method for seeking (approximate) optimal amounts of correction with respect to the position of a probe jig on the basis of nonlinear mathematical programming techniques. Solution methods based on nature-inspired algorithms are proposed. In Section 4, we conduct numerical experiments and apply the proposed method to the field of PCB inspection. The experimental results show that FA-based solution method is best among four algorithms. Finally, Section 5 summarizes this paper and discuss future works.

2 Electrical wiring pattern test in PCB inspections

2.1 Probe jig in electrical PCB inspection In production process of PCBs, various wiring patterns are etched on PCBs. When a certain problem happens in forming wiring patterns, PCBs may include some defects such as open (disconnection) defects and short defects.

For electrical pattern test of wiring, a test jig, called a *probe jig*, is used. Probe jigs have many of very small pins. On the other hand, wirings on PCBs have bulged parts, called *contact pads*. The diameter of contact pads is about 100 $\sim$ 300 μm.

Electric wiring pattern tests are done by pressing a probe jig onto PCB sheets and carrying electric currents through pins into contact pads of wirings, as shown in Figure 1.

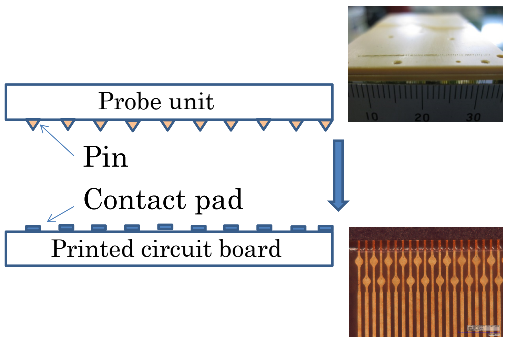

Figure 1: Wiring pattern test with a probe jig

To exactly conduct the test, each pin must hit the corresponding contact pad. In other words, a probe jig needs to be set in a proper position for exact pattern tests. If the position of a probe jig is not proper, then a part of the pins on the probe jig does not hit the corresponding wirings on PCBs, which causes inspection errors and constitutes a temporary suspension of the PCB production line. Since the size of contact pads is very small (100 μm $\sim$ 300 μm), setting a probe jig in a proper position is considered as a burdensome task.

2.2 Existing probe jig position correction method In order to avoid inspection errors, operators need to correct the position of a probe jig by rotating and translating the probe jig, as shown in Figure 2, so that all the pins can hit the corresponding contact pads.

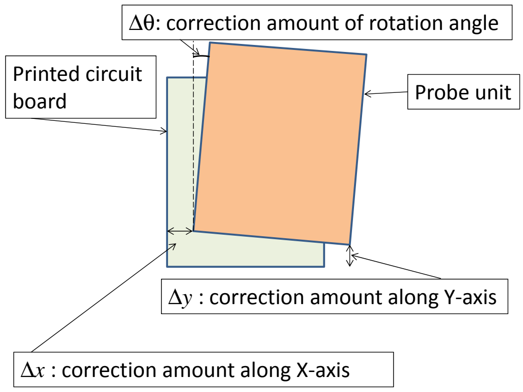

Figure 2: Probe jig position correction

The following is the steps of the existing position-correction method:

Step 1: An operator attaches a probe jig to an inspection unit and fixes the probe jig with a robot arm.

Step 2: The operator presses a probe jig onto a PCB sheet so that the pins mark the contact pads (as shown in Figure 1).

Step 3: The operator looks into the surface of the PCB sheet via a microscope and checks whether the gap of the pin marks and the pads are within acceptable distances. If the operator is satisfied with the current position of the probe jig, then stop. Otherwise, go to Step 4.

Step 4: The operator determines the correction amounts with respect to 3 parameters such as Δx (x-axis: vertical direction), Δy (y-axis: horizontal direction) and $\Delta\theta$ (rotational angle), and corrects the position of the probe jig, as shown in Figure 2. The operator attaches the PCB sheet to the machine. Return to Step 2.

2.3 Difficulty of correcting a probe jig position Because of production errors, there is no correction amount that makes all the pin marks completely fit in with the center of the corresponding contact pads. The real position of pins and contact pads are generally different from the ideal position determined by a blueprint which is caused by production errors. In addition, the amounts and directions of these gaps are not constant but dependent on pins and compact pads. If all the real positions of pins and contact pads would be completely consistent with those ideal positions determined by a blueprint, operators could choose only two pairs of pins and contact pads and calculate probe jig correction amounts, and then exactly correct the probe jig position. However, as already stated, the fact is that there are gaps between real positions and ideal positions, which makes the problem difficult to solve.

Once the position of a probe jig is fixed, the machine sequentially inspects a number of PCB sheets. It should be noted here that even if the position of a probe jig is proper for one PCB sheet, the position is not always proper for another PCB sheet because the amounts of production errors are dependent on PCB sheets. Assuming that a pin barely hits the borderline of the corresponding contact pad in one PCB sheet, then such a pin may not hit the corresponding contact pad in another PCB sheet because of production errors. Therefore, operators need to make all the pin marks approach the center of the corresponding contact pad as close as possible.

2.4 Necessity of a new probe-jig position correction method In the existing method, the correction amounts are determined by operators' own feelings, which means that the time necessary for completing the task is dependent on skill levels of operators. Due to the miniaturization of PCBs, it takes even skilled operators 4 hours averagely (one-and-a-half-day in the worst case) to complete the task because operators need to correct the position bit by bit. The correction amounts are extremely small, and they are tens of nanometers in vertical and horizontal directions and less than an angle of 3 degrees. Thus, the procedure of setting probe jigs has been considered as one of the most burdensome tasks for operators in the field of PCB electrical inspections.

In order to explain the fact that the existing correction method is time-consuming, let us give a very simple example of situations that the probe-jig-position correction is needed, as shown in Figure 3.

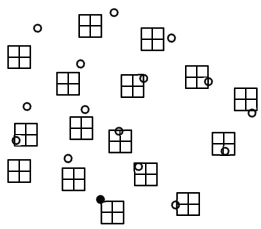

Figure 3: State before correction

Readers can try to estimate the correction vector $(\Delta x,\ \Delta y,\ \Delta\theta)$ so that all the pin marks coincide with their corresponding contact pads, but it is probably almost impossible for most readers to precisely estimate it by eye. In fact, one of the precise (near-optimal) correction vectors is $(\Delta x^*,\ \Delta y^*,\ \Delta\theta^*) = (7.14,\ -14.27,\ -11.52)$. Figure 4 shows the state after the probe-jig position is corrected based on $(\Delta x^*,\ \Delta y^*,\ \Delta\theta^*)$.

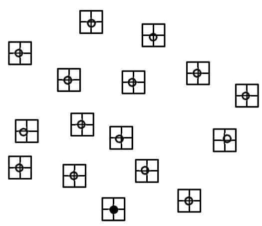

Figure 4: State after correction

Thus, since it is quite difficult even for skilled operators to precisely estimate the correction vector at one time, trial-and-error adjustments are needed. Therefore, the cycles of Steps 2-4 in the current algorithm described at the end of Section 2.2 are repeated again and again until the operator judges that the probe-jig position is sufficiently good, that is, until she/he is satisfied with the probe-jig position. It takes skilled operators about 4 hours to complete the task in the most difficult case. In this way, the probe-jig-position correction is a time-consuming task in the field of PCB inspections.

More generally, in order to enhance the efficiency of electrical PCB inspections, it is very important to reduce time for setting up probe jigs. Recently, Allahverdi and Soroush [2] have claimed the significance of reducing setup times/setup costs in PCB assembly. Bard et al. [3] presented studies on machine setup and component placement in PCB assembly. Trovinger and Bohn [6] developed PCB assembly tools for setup time reduction. These research papers discussed setup time/setup costs for PCB assembly, not for PCB inspections. As far as the authors know, there is no study on optimization techniques for reducing setup time for PCB inspections. In this sense, this paper tackles one of the most challenging problems in PCB manufacturing.

3 Nonlinear programming-based model and its natured-inspired solution algorithms In this section, we propose a new probe jig position correction algorithm in order to reduce setup time for PCB inspections. Firstly, we model the problem as a nonlinear programming problem. Next, we propose natured-inspired algorithms, which draw attentions to, in order to correct a probe jig position through interactive processes.

3.1 Problem formulation: correcting a probe jig position Firstly, we introduce the notations used throughout this paper.

N:	Number of pins attached on a probe jig
k:	Number of trials for correcting a probe jig position (Number of iterations)
S_k:	A set of pairs of pins and contact pads selected until kth trial
$\boldsymbol{p}_i^k = (p_{ix}^k, p_{iy}^k)$:	Coordinate of the mark of ith pin in the kth trial
$\boldsymbol{t}_i = (t_{ix}, t_{iy})$:	Coordinate of the contact pad to be contacted with ith pin in the kth trial
$\boldsymbol{r}^k = (r_x^k, r_y^k)$:	Coordinate of the rotational center of the robot holding a probe jig in the kth trial

In general, the coordinate origin and the coordinate of the rotational center are different. For convenience of computation, coordinate transformation is performed by subtracting the coordinate of the rotational center from the coordinates of the marks and the center of contact pads, which means that the coordinate of the rotational center is regarded as the coordinate origin. Let $\hat{\boldsymbol{p}}_i^k$ and $\hat{\boldsymbol{t}}_i^k$ $(i \in S_k)$ be the coordinates of the mark hit by the ith pin and the corresponding contact pad for the kth trial after the coordinate transformation, respectively. Then, we have

$$\hat{\boldsymbol{p}}_i^k = (p_{ix}^k - r_x^k, p_{iy}^k - r_y^k),\ \hat{\boldsymbol{t}}_i^k = (t_{ix} - r_x^k, t_{iy} - r_y^k),$$

where the vector with hat (ˆ) means the coordinate after coordinate transformation.

In each trial, probe jig position correction is initiated by rotation of PCB sheets at $\Delta\theta^k$ degrees, and then parallel translation of PCB sheets is performed in the direction of (Δx^k, Δy^k), as shown in Figure 5.

As a result of the combination of rotation and transformation, the coordinate of the ith pin on the probe jig, denoted by $\hat{\boldsymbol{p}}_i^{k+1} = (\hat{p}_{ix}^{k+1}, \hat{p}_{iy}^{k+1})$, is calculated as follows:

$$\begin{pmatrix} \hat{p}_{ix}^{k+1} \\ \hat{p}_{iy}^{k+1} \end{pmatrix} = \begin{pmatrix} \cos\Delta\theta^k & -\sin\Delta\theta^k \\ \sin\Delta\theta^k & \cos\Delta\theta^k \end{pmatrix} \begin{pmatrix} \hat{p}_{ix}^{k} \\ \hat{p}_{iy}^{k} \end{pmatrix} + \begin{pmatrix} \Delta x^k \\ \Delta y^k \end{pmatrix}$$

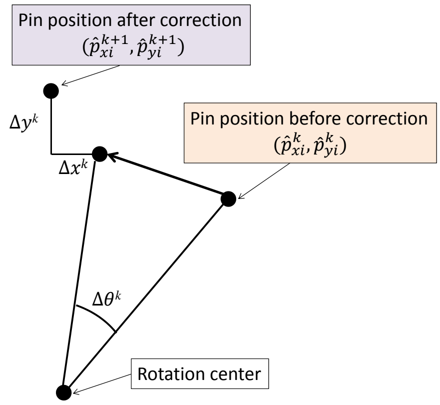

Figure 5: Position correction of a probe jig

As stated previously, in this research, we propose a computational method for minimizing the largest gap among all pairs of pins and the corresponding contact pads. In addition, needless to say, the smaller the total gap is, the better it is. With this observation in mind, in this research, we formulate the following nonlinear programming problem in order to obtain an (approximate) optimal correction amounts of a probe jig:

$$(1)\quad \begin{cases} \text{minimize} & \max_{i \in S_k} z_i^k + \rho \sum_{i \in S_k} z_i^k \\ \text{subject to} & z_i^k = \left|\left| \hat{\boldsymbol{t}}_i^k - \hat{\boldsymbol{p}}_i^{k+1} \right|\right|, \ i \in S_k \\ & (\Delta x^k,\ \Delta y^k,\ \Delta\theta^k) \in \mathbb{R}^3 \end{cases}$$

where

$$\begin{pmatrix} \hat{p}_{ix}^{k+1} \\ \hat{p}_{iy}^{k+1} \end{pmatrix} = \begin{pmatrix} \cos\Delta\theta^k & -\sin\Delta\theta^k \\ \sin\Delta\theta^k & \cos\Delta\theta^k \end{pmatrix} \begin{pmatrix} \hat{p}_{ix}^{k} \\ \hat{p}_{iy}^{k} \end{pmatrix} + \begin{pmatrix} \Delta x^k \\ \Delta y^k \end{pmatrix},$$

$$\hat{\boldsymbol{p}}_i^k = (p_{ix}^k - r_x^k, p_{iy}^k - r_y^k),\ \hat{\boldsymbol{t}}_i^k = (t_{ix} - r_x^k, t_{iy} - r_y^k),$$

ρ is a sufficiently small constant, and $||\cdot||$ denotes the Euclid norm. The main objective of problem (1) is to minimize the maximum gap, and the second objective is to minimize the total gap. Problem (1) is a non-convex nonlinear programming problem. In order to efficiently solve the problem, we apply some nature-inspired metaheuristic algorithms such as FA, BA, CA and FPA, which are known as the most state-of-the-art algorithms. It should be noted here that an exact optimal solution is not always obtained when metaheuristic algorithms is used to solve optimization problems.

3.2 Nature-inspired solution algorithms Nature-inspired optimization algorithms provide a systematic introduction to all major nature-inspired algorithms for optimization. Algorithm collective behavior of insects called natural inspiration, in which individual behavior is very simple, but when they work together, can be very complicated inspired behavioral characteristics. This paper introduces four kinds of nature-inspired optimization algorithm: Firefly algorithm (FA), cuckoo search (CS), bat algorithm (BA), flower pollination algorithm (FPA) to study the mechanism of composing four algorithms, the basic model and process implementation.

a. Firefly algorithm (FA) [7]

Firefly algorithm is affected by the nature of the information exchange fireflies which were inspired by fluorescence behavior of such groups evolved. The primary purpose for a firefly's flash is to act as a signal system to attract other fireflies. Fireflies with low attractiveness move to fireflies with high attractiveness. Attractiveness is proportional to their brightness, and the degree of attractiveness decrease as their mutual distance between two fireflies increases. If there are no fireflies brighter than a given firefly, it will move randomly. The brightness is associated with the objective function.

b. Cuckoo search (CS) [10]

Cuckoo search is an optimization algorithm, which is inspired by the obligate brood parasitism of some cuckoo species by laying their eggs in the nests of other host birds. Eggs are always stored in their nests of other birds. Other birds hatch their next generation in order to reduce the probability of discovery of some cuckoo birds. When the other birds find that their nest eggs have exotic, alien eggs will discard or abandon their nest. In optimization problems, many eggs in host nests are generated as individuals in population. A feasible solution corresponding to an egg in a host nest is randomly chosen, and the chosen solution (or egg) is replaced by a new solution generated by using Lévy flight based on Lévy distribution. Lévy distribution is a special case of the inverse-gamma distribution, and is one of stable distributions which have probability density functions that can be expressed analytically. The idea of Lévy flight is incorporated in the Flower Pollination Algorithm (FPA) which will be explained later.

c. Bat algorithm (BA) [8]

The Bat algorithm is inspired by the echolocation behavior of microbats, with varying pulse rates of emission and loudness. The algorithm search process will be modeled as individuals search for prey and bats' movement. First, the individual is mapped into a bat at some point in the search space. Each virtual bat flies randomly with a velocity at position (solution) with a varying frequency or wavelength and loudness. As it searches and finds its prey, it changes frequency, loudness and pulse emission rate. Search is intensified by a local random walk.

d. Flower pollination algorithm (FPA) [9]

Flower pollination algorithm is a new swarm intelligence optimization algorithm. It utilizes transition probability parameters to well balance the global search and local search algorithm, while the Lévy flight mechanism is used, which leads to a good global diversity optimization capabilities.

The pseudo code of the proposed Flower Pollination Algorithm (FPA) is as follows:

Flower Pollination Algorithm

1: Set an objective *min* or *max* $f(\boldsymbol{x})$, $\boldsymbol{x} = (x_1, x_2, ..., x_d)$;

2: Initialize a population of n flowers/pollen gametes with random solutions ;

3: Find the best solution g_* in the initial population;

4: Define a switch probability $p \in [0, 1]$;

5: **while** $(t < MaxGeneration)$

6:     **for** $i = 1 : n$ (all n flowers in the population)

```
        if rand < p
            Draw a (d-dimensional) step vector L which obeys a Lévy distribution;
            Global pollination via X_i^{t+1} = X_i^t + L(g_* − X_i^t);
        else
            Draw ε from a uniform distribution in [0, 1];
            Randomly choose j and k among all the solutions;
            Do local pollination via X_i^{t+1} = X_i^t + ε(X_j^t − X_k^t);
        end if
        Evaluate new solutions;
        If new solutions are better, update them in the population;
    end for
    Find the current best solution g_*;
end while
```

In the above algorithm, *rand* represents a rational number that is generated at random from the interval $[0, 1]$. Lévy distribution is one of fat-tailed distributions and play an important role in bringing the diversity of populations.

Now we are ready to propose a computational algorithm for correcting the position of a probe jig as follows:

An interactive algorithm for correcting a probe jig position

Step 1: Let $k := 0$ and $S_k := \emptyset$.

Step 2: An operator presses a probe jig onto a PCB sheet.

Step 3: The operator looks into the gaps between pins and the corresponding contact pads via a monitor. If the operator considers the amounts of the gaps acceptable, then stop. Otherwise, set $k := k + 1$ and go to Step 4.

Step 4: The operator chooses several pairs of pins and the corresponding contact pads. Let A_k be a set of the chosen pairs. Update $S_k := S_{k-1} \cup A_k$.

Step 5: Solve problem (1) using a certain nature-inspired metaheuristic algorithm, and obtain (approximate) optimal correction amounts. Correct the probe jig position in accordance with the obtained correction amounts. Return to Step 2.

4 Numerical experiments

4.1 Performance comparison of four nature-inspired algorithms In order to verify the effectiveness of the proposed method, we conduct several numerical experiments using a simple example of PCB sheets, in which a wiring pattern is formed as 100 (10 × 10) square-shaped contact pads. Each contact pad is a 10-μm square. The numerical experiments are conducted using Windows 7 Home premium as OS, Intel(R) Core(TM) i7-2620 CPU @2.70GHz as CPU, 6GB RAM and C++ (Microsoft Visual C++ 2010 Express) programming language. In (1), assume that the operator sets $\rho = 0.05$.

We employ 4 nature-inspired algorithms such as FA, CS, BA and FPA to compute the correction amounts of the probe jig. Table 1 shows the experimental results of the four algorithms. FPA is best among four algorithms, because both the maximum gap and the iteration number are smaller than any other methods.

Table 1: Example of numerical experiments

Method	Maximum Gap $(k = 2)(\mu m)$	Minimum Iterations	Average Iterations	Maximum Iterations	Average Time (*sec.*)
FA	30.5	5	7.3	9	0.09
CS	22.1	3	4.0	5	0.14
BA	21.8	3	4.0	5	0.16
FPA	15.6	1	2.0	3	0.12

4.2 Experimental results of FPA Since it has shown that FPA is best among four algorithms, we present one of experimental results of FPA in more details. According to Step 1 in the proposed algorithm, let $k := 0$ and $S_k(= S_0) = \emptyset$. An operator pressed a probe jig onto a PCB sheet in Step 2, and the operator observed that many pins did not hit the corresponding contact pads in Step 3. The maximum gap was 98.5 μm.

Since the operator was not satisfied with the current state, $k := k + 1$ $(= 1)$ was set, and the operator selected 4 pins and pads at the corners in order to correct a probe jig position. In Step 4, A_1 consisted of 4 pairs of the pins and pads at the corners, and $S_1 := S_0 \cup A_1 = A_1$. In Step 5, using the coordinates of the marks of these pins and those of the center of the pads, the calculated correction amounts using FPA were $\Delta x^1 = 12.3$ (μm), $\Delta y^1 = -15.9$ (μm) and $\Delta\theta^1 = 5.27$ $(degree)$. The maximum gap was 17.3 μm. Then, the operator returned to Step 2 and pressed the probe jig onto the PCB sheet again.

Although all the pins hit the corresponding contact pads, the operator was not satisfied with the current state because he considered the maximum gap was a little big. Then, set $k := k + 1$ $(k = 2)$. In Step 5 of the second trial $(k = 2)$, the proposed algorithm based on FPA computed the correction amounts of the probe jig, and they were $\Delta x^2 = -1.20$ (μm), $\Delta y^2 = 0.93$ (μm) and $\Delta\theta^2 = -0.35$ $(degree)$. The maximum gap was 15.6 μm. Since the operator was satisfied with the current probe-jig position, an interactive algorithm stopped.

Table 2 shows the experimental result, where the maximum gap $\Delta^k_{\max}$ for the kth trial becomes smaller and smaller as k is incremented.

5 Conclusion This paper has focused on an optimization problem in PCB electrical inspections. In the field of PCB inspections, the position-correction of a probe jig had been considered as a time-consuming hard task. In order to speed up the position-correction of a probe jig, we have modeled a real decision making situation as a nonlinear programming-based optimization problem and proposed solution methods based on nature-inspired metaheuristic algorithms. We have compared four nature-inspired metaheuristic algorithms such as firefly algorithm (FA), bat algorithm (BA), cuckoo search (CS) and flower pollination

Table 2: Experimental result of FPA-based method

	Δx^k (μm)	Δy^k (μm)	$\Delta\theta^k$ (degree)	$\Delta^k_{\max}$ (μm)
$k=0$				98.5
$k=1$	12.3	-15.9	5.27	17.3
$k=2$	-1.20	0.93	-0.35	15.6

algorithm (FPA), and concluded from the experimental results that FPA is best for obtaining the optimal position correction of a probe jig in PCB inspections. It is estimated that it takes less than 20 minutes to complete the task of correcting the probe-jig position if operators use the proposed method, while it took around 4 hours when the existing method was used. In the future, the FPA-based optimization method will be installed to inspection machines in the field.

References

[1] K. Altinkemer, B. Kazaz, M. Koksalan, H. Moskowitz, "Optimization of printed circuit board manufacturing: Integrated modeling and algorithms," *European Journal of Operational Research*, Vol. 124, No. 2, pp. 409–421, 2000.

[2] A. Allahverdi, H.M. Soroush, "The significance of reducing setup times/setup costs," *European Journal of Operational Research*, Vol. 187, No. 3, pp. 978–984, 2008.

[3] J.F. Bard, R.W. Clayton, T.A. Feo, "Machine setup and component placement in printed circuit board assembly," *International Journal of Flexible Manufacturing Systems*, Vol. 6, No. 1, pp. 5–31, 1994.

[4] Y. Crama, A.G. Oerlemans, F.C.R. Spieksma, *Production Planning in Automated Manufacturing*, Springer, Berlin Heidelberg, 2nd ed., 2011.

[5] Y. Crama, J. van de Klundert, F.C.R. Spieksma, "Production planning problems in printed circuit board assembly," *Discrete Applied Mathematics*, Vol. 123, No. 1–3, pp. 339–361, 2002.

[6] S.C. Trovinger and R.E. Bohn, "Setup time reduction for electronics assembly: combining simple (SMED) and IT-based methods," *Production and Operations Management*, Vol. 14, No. 2, pp. 205–217, 2005.

[7] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*, *Luniver Press*, 2008.

[8] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," *Nature inspired cooperative strategies for optimization (NICSO 2010)*, Springer, Berlin Heidelberg, pp. 65–74, 2010.

[9] X.-S. Yang, "Flower pollination algorithm for global optimization," *Proceedings of International Conference on Unconventional Computing and Natural Computation*, Springer, Berlin Heidelberg, pp. 240–249, 2012.

[10] X.-S. Yang, S. Deb, "Cuckoo search via Lévy flights," *Proceedings of IEEE World Congress on Nature & Biologically Inspired Computing (NaBIC 2009)*, pp. 210–214, 2009.

Communicated by *Hiroaki Ishii*

Hideki Katagiri,
Department of Industrial Engineering and Management ,
Faculty of Engineering, Kanagawa University
katagiri@kanagawa-u.ac.jp