

## A ROUTE OPTIMIZATION PROBLEM IN ELECTRICAL PCB INSPECTIONS WITH ALIGNMENT OPERATIONS

HIDEKI KATAGIRI, HONGWEI WU, YOSUKE KAKIUCHI, HIROSHI HAMORI  
AND KOSUKE KATO

Received December 8, 2015

Revised January 30, 2016

**ABSTRACT.** This paper considers a route optimization problem in advanced electrical PCB inspections. By considering the constraint that “camera-based alignment of position” needs to be conducted before electrical tests, the PCB inspection route optimization problem (PCBIRP) is modeled as a precedence-constrained traveling salesman problem (PCTSP), especially, as a pickup and delivery traveling salesman problem (PDTSP). Two of mixed 0-1 integer programming problem formulations are proposed. The computational times for the proposed formulations are compared by solving benchmark instances using some of well-known mathematical programming solvers.

**1 Introduction** Printed circuit boards (PCBs) have been used in almost all electric devices. There are many of previous studies on optimization techniques for PCB manufacturing processes such as assembly operations [1, 5, 11] and drilling processes [2]. On the other hand, optimization techniques for PCB inspections have not been sufficiently developed so far except for some studies on multi-chip module substrate testing [10, 14].

PCB inspections are quite important to enhance the reliability of manufactured PCBs. In addition, since the number of PCBs to be inspected has been recently increasing, the speedup of PCB inspections has become one of the most important issues in the field. In production processes of PCBs, defect generation may arise due to some trouble, which prevents PCBs from working properly. In electrical PCB inspections, all the PCBs arrayed in a plain are visited and tested by an inspection jig in some sequence or order.

Since the inspection time is dependent on the length of traveling (visiting) route of an inspection jig, it is worth finding the best inspection sequence or route in order to reduce the inspection time. On the other hand, the procedure of the camera-based “alignment” of position (hereafter we call it just an *alignment operation*) is additionally needed before electrical tests in recently-developed PCB inspection machines. However, a route optimization problem in such advanced inspections with alignment operations has not been discussed so far, and there has been no article to model the problem using mathematical programming.

This paper is organized as follows: Section 2 reviews an advanced electrical PCB inspection method involving “alignment” operations, and discusses the necessity of route optimization. In Section 3, we model the PCB inspection route optimization problem (PCBIRP) as a class of pickup and delivery traveling salesman problems (PDTSPs) [3, 15] and provide two of mixed 0-1 integer programming problem formulations. In Section 4, numerical experiments are conducted by solving benchmark instances based on real PCB wiring patterns, using some of well-known mathematical programming solvers. Finally, in Section 5, we summarize this paper and discuss future works.

---

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

*Key words and phrases.* Printed circuit board (PCB), inspection route optimization, pickup and delivery traveling salesman problems, mixed 0-1 integer programming problem, exact solutions.

**2 Preliminaries** In this section, we review an advanced electrical PCB inspection method and explain the reason why the *alignment operations* via a camera have been recently necessary in PCB inspections. In addition, we discuss the necessity of considering route optimization problems in electrical PCB inspections with alignment operations.

**2.1 Electrical test of PCB** In production process of PCBs, various wiring patterns are etched on PCBs. When some trouble happens in forming wiring patterns, PCBs may include some defects such as open (disconnection) defects and/or short-circuit defects. PCBs have bulged parts, called *contact pads*, as shown in Figure 1, which are used to electrically inspect PCBs. In some cases, the number of pins is more than 1,000. The diameter of contact pads is about  $100 \sim 300 \mu\text{m}$ .

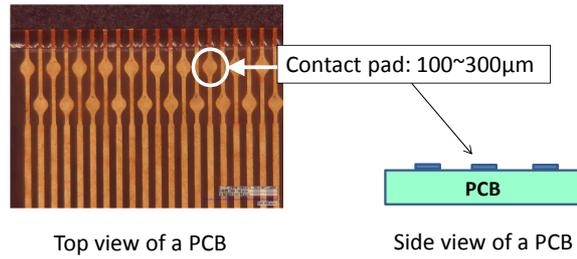


Figure 1: Contact pads in a PCB

In order to electrically test wiring patterns on PCBs, a test jig, called a *probe jig*, is used, as shown on the left-hand side of Figure 2. A probe jig contains many sharp, thin and conductive pins with a diameter of  $20 \sim 130 \mu\text{m}$ . The number of pins is usually equal to that of contact pads, and there is a one-to-one correspondence between contact pads and pins.

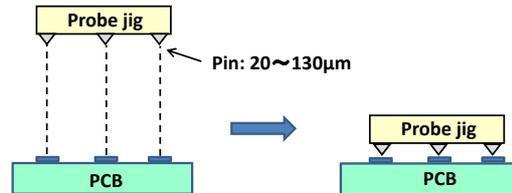


Figure 2: Electrical test via a probe jig

Electrical tests are conducted by pressing a probe jig onto a PCB and by feeding electric currents through pins into contact pads of PCB wirings, as shown on the right-hand side of Figure 2. Electric currents are fed into wiring through pins in order to check if each wiring has defects such as open or short circuits. Therefore, for valid tests, every pin of the probe jig needs to have a contact with the corresponding contact pad.

To exactly conduct the electrical continuity test, each PCB has the so-called “test position.” For valid tests, it is necessary to make the reference point of a probe jig exactly located at the test position of a PCB, as shown on the left-hand side of Figure 3.

To be more specific, when the reference point of a probe jig is precisely placed at the test position of a PCB, each pin of the probe jig has a contact with its corresponding contact

pad of the PCB, as shown on the right-hand side of Figure 3.

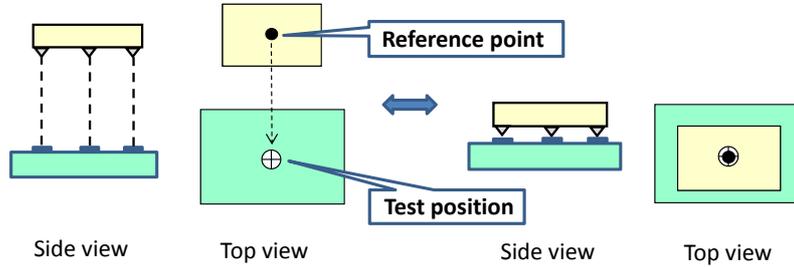


Figure 3: Proper inspection

On the other hand, as shown in Figure 4, when the reference point is not placed at the test position, inspection failures occur. In this case, because some of pins do not have contacts with their corresponding contact pads, electric currents are not fed into wiring lines through the pins, which causes the misjudgment of whether there is a defect or not.

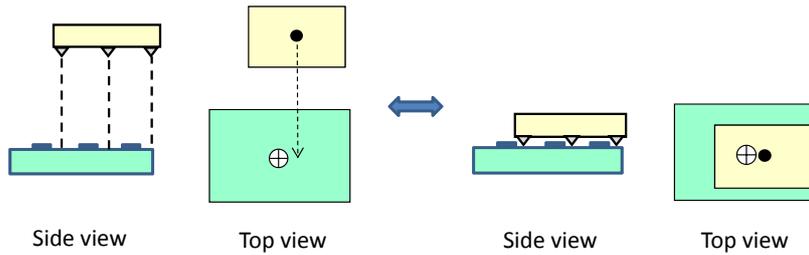


Figure 4: Failed inspection

**2.2 Electrical PCB inspection with alignment operations** Recently, inspection failures tend to occur more frequently than before, because the miniaturization of electronic devices makes it more difficult to exactly put the reference point of a probe jig onto the test position of a PCB. Since PCB sheets are very thin like pieces of paper, they easily undergo deflections. When there is a deflection in a PCB sheet, the position of the reference point of a probe jig is changed from the regular position, which leads to the situation that some pins of the probe jig do not have contacts with their corresponding contact pads.

In order to deal with such a change of the test position due to the deflection of PCB sheets, the “alignment” operation has been introduced in advanced PCB inspection systems. A camera is attached to the probe jig, and each PCB has one or two alignment marks, as shown in Figure 5 (in this case, the PCB has two alignment marks). We call a probe jig with a camera “a probe unit.” The camera is used to capture the images of alignment marks, which acquires the information on the exact coordinate of the test position.

Thus, the advanced PCB inspection method involving alignment operations consists of two steps, *Alignment operation* and *Electrical test*, as follows:

[Procedures of electrical PCB inspections with alignment operations]

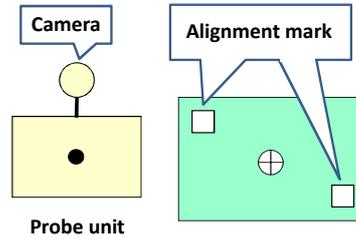


Figure 5: Alignment marks (captured by the camera of a probe unit)

**Step 1 (Alignment operation):**

Move the camera of the probe unit to the position of an alignment mark so that the camera can capture the image of the alignment mark (cf. the left-hand side of Figure 6).

**Step 2 (Electrical test):**

Move the reference point of the probe unit to the test position of the PCB so that the probe jig can properly press onto each wiring pattern of the PCB (cf. the right-hand side of Figure 6).

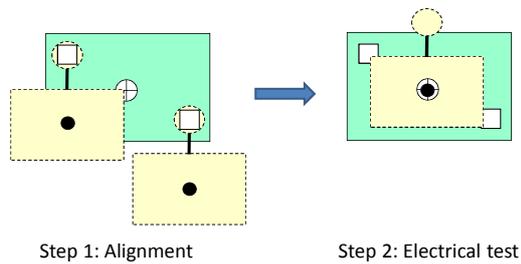


Figure 6: Electrical PCB inspection (alignment operation and electrical test)

**2.3 Route optimization** Route optimization in PCB inspections is considerably important. Even if the reduced inspection time is several percent, it can bring a great effect on cost reduction as well as on productive efficiency of PCBs. It is because there are a large number of PCB sheets to be inspected in the field, and the number of PCB sheets to be inspected by one machine per day is more than 1,000.

Each of PCB sheets consists of many PCBs with the same wiring patterns which are arrayed in a plane. Figure 7 is a simple example of a PCB sheet which consists of only 4 ( $2 \times 2$ ) PCBs. In general, the number of the same wiring pattern arrayed in one PCB sheet ranges from 4 to around 200.

Our interest is to find the shortest route for inspecting all the PCBs in the sheet. The route length is dependent on the visiting sequence (order) of the probe jig. Figure 8 shows a simple inspection sequence in which the test position for each PCB is visited immediately after the corresponding alignment marks are visited in order. It should be noted here that each test position does not need to be visited immediately after the corresponding alignment marks are visited.

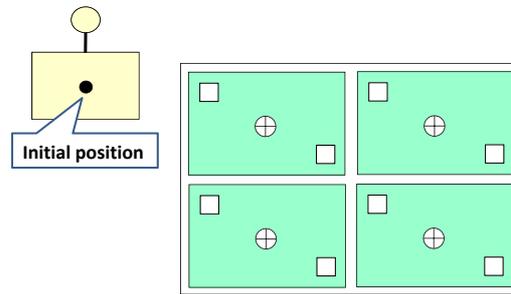


Figure 7: PCB sheet

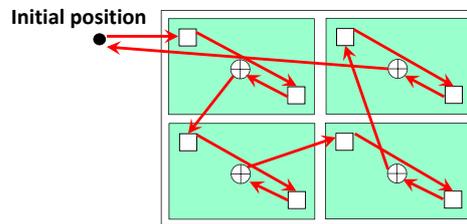


Figure 8: Simple visiting sequence (order): Example 1

Figure 9 shows another simple inspection sequence in which test positions are visited after all the alignment marks are visited. A probe unit firstly moves to the alignment mark located at the upper left, and then visits only alignment marks in order. After all the alignment marks are visited, the probe unit moves to the nearest test position from the lastly visited alignment mark, and visits only test positions in the inverse sequence (order) of alignment marks that were already visited.

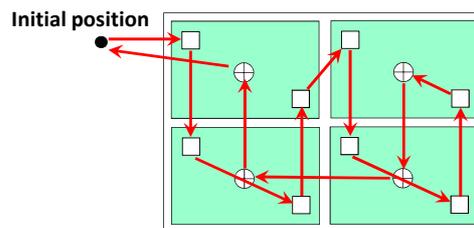


Figure 9: Simple visiting sequence (order): Example 2

Apparently, the inspection routes based on the sequence shown in Figures 8 and 9 are not optimal, and there may exist other shorter routes. This kind of optimization problem for obtaining the shortest route in electrical PCB inspections with alignment operations has not been discussed so far.

In the next section, we will address how to find an optimal inspection route in advanced electrical PCB inspections.

**3 Modelling and problem formulation** In this section, we show that the problem of finding an optimal inspection route, namely, an inspection route with the minimum length (an optimal route) can be modeled as a kind of PDTSPs, and that it is formulated as mixed 0-1 integer programming problems.

**3.1 Pickup and delivery traveling salesman problem (PDTSP)** A pickup and delivery traveling salesman problem (PDTSP) [3, 15] is a kind of TSPs in which all vertices are characterized as pickup and/or delivery vertices. PDTSPs can be roughly classified into three groups [3] such as 1) one-to-one, 2) many-to-many, 3) one-to-many-to-one. In one-to-one problems, each commodity has exactly one pickup vertex and one delivery vertex. In many-to-many PDTSPs, several origins (pickup vertices) and destinations (delivery vertices) are characterized for each commodity. In one-to-many-to-one PDTSPs, some commodities (e.g., food or drinks) are delivered from the depot to customers while other commodities (e.g., empty bottles) supplied by the customers are brought back to the depot.

As will be discussed in the next section, the PCB inspection route optimization problem (PCBIRP) can be modeled as a many-to-one or one-to-one problem in which each commodity has several (or one) pickup vertices (vertex) but only one delivery vertex.

**3.2 Modelling based on a PDTSP** To illuminate the readers' understanding of the ideas of this paper, we give graphical explanations with the example shown in Figure 7. In order for the camera of a probe unit to capture the images of alignment marks, the camera must be moved to alignment mark  $A$ , as shown in Figure 10. This operation is equivalent to moving the reference point of the probe unit (the center of the probe jig) to vertex  $A'$ . Thus, we consider the graph in which the alignment mark  $A$  is moved to  $A'$ .

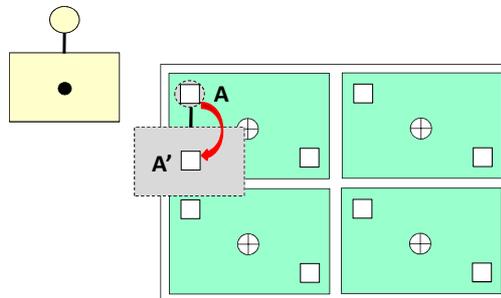


Figure 10: Image capturing of alignment mark  $A$

In a similar manner, we transfer all the positions of alignment marks, and obtain a new graph shown in Figure 11, which represents a set of vertices to be visited by the reference point of a probe unit.

As described before, there are precedence constraints between alignment marks and their corresponding test position for each PCB. In Figure 12, dotted lines represent precedence relations, which means that for each PCB, two alignment marks must be visited before the corresponding test position is visited.

In general, when the number of alignment marks is two, the PCBIRP can be modeled as a two-to-one PDTSP; two alignment marks are regarded as pickup vertices, and the corresponding one test position is regarded as a delivery vertex. It should be noted that each vertex is characterized as *either* of pickup and delivery vertices. This is different from

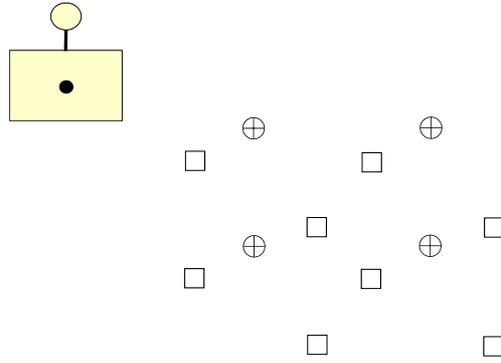


Figure 11: Vertices to be visited by the probe unit

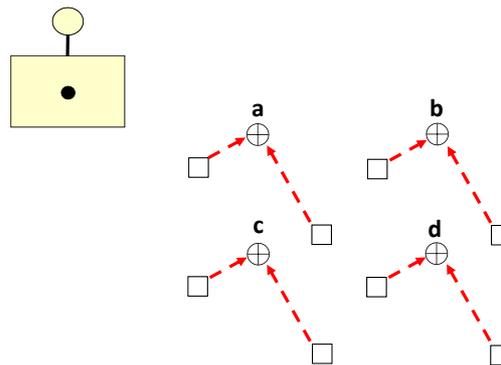


Figure 12: Precedence relationships between alignment marks and test positions

the original many-to-many PDTSPs; the original many-to-many PDTSPs allow each vertex to be *both* a pickup and a delivery vertex.

On the other hand, when the number of alignment marks is exactly one, the problem can be regarded as a one-to-one PDTSP, because there exists a one-to-one correspondence between each alignment mark and its corresponding test position.

The goal of the PCBIRP is to obtain an optimal (shortest) route. The length of the inspection route is dependent on the visiting sequence (order), and there are many feasible inspection routes. For example, Figure 13 shows the simple route constructed based on the visiting sequence (order) shown in Figure 8. In this figure, firstly, all the alignment marks are visited, and then all the test positions are visited. This is not an optimal (shortest) route.

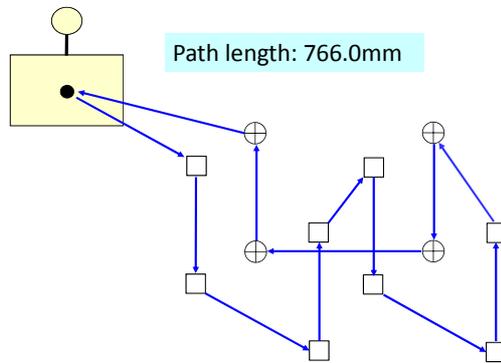


Figure 13: Simple inspection route

Figure 14 shows the optimal route, namely, the shortest route (cycle). In the next section, we shall give mathematical programming formulations in order to obtain an optimal route.

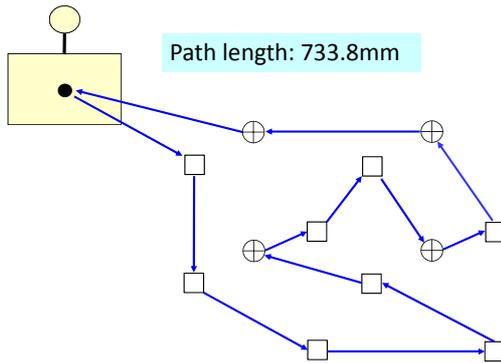


Figure 14: Optimal inspection route

**3.3 Integer programming-based problem formulation** Here, we formulate the PCBIRP as mixed 0-1 integer programming problems. In preparation for problem formulation, we

use the following mathematical notation:

- 0: Vertex corresponding to the initial point of a probe unit
- $B$ : Set of PCBs to be tested, defined by  $\{1, 2, \dots, l\}$
- $A_p$ : Set of vertices corresponding to alignment marks of  $p$ -th PCB ( $p \in B$ )
- $t_p$ : Vertex corresponding to the test position of  $p$ -th PCB ( $p \in B$ )
- $N$ : Set of all the vertices to be visited by a probe unit defined by  $N = \cup_{p=1}^l (A_p \cup \{t_p\})$
- $e_{ij}$ : Edge between vertices  $i$  and  $j$  ( $i, j \in N \cup \{0\}$ )
- $E$ : Set of all the edges  $e_{ij}$ ,  $\forall i, j \in N \cup \{0\}$
- $c_{ij}$ : Length of  $e_{ij}$  ( $e_{ij} \in E$ )

For notational convenience, the test position for the  $p$ -th PCB is represented as a singleton  $\{t_p\}$ , although there is only a single test position for each PCB. As for alignment marks, since there are one or two alignment marks, the number of elements in  $A_p$  is one or two. Figure 15 shows an example of 4 PCBs where there are 13 numbered vertices  $0 \cup N = \{0, 1, \dots, 12\}$  in which  $B = \{1, 2, 3, 4\}$  ( $l = 4$ ),  $A_1 = \{1, 2\}$ ,  $A_2 = \{3, 4\}$ ,  $A_3 = \{5, 6\}$ ,  $A_4 = \{7, 8\}$ ,  $t_1 = 9$ ,  $t_2 = 10$ ,  $t_3 = 11$ ,  $t_4 = 12$  and  $N = \{1, 2, \dots, 12\}$ .

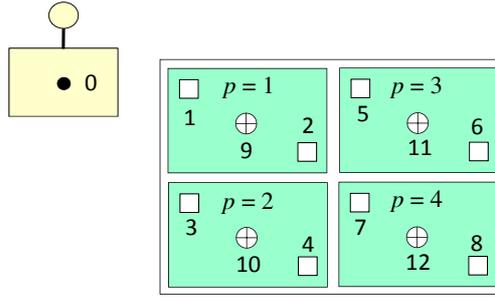


Figure 15: Number of vertices

In order to formulate PCBIRPs as mathematical programming problems, we introduce decision variables  $x_{ij}$ s as follows:

$$x_{ij} = \begin{cases} 1 & \text{if } j \text{ is visited immediately after } i \text{ is visited} \\ 0 & \text{otherwise.} \end{cases}$$

Decision variable  $x_{ij}$  is used to represent inspection routes, namely, to construct a route by connecting all edges with  $x_{ij} = 1$ .

In this paper, we consider the well-known ‘‘polynomial formulations’’ where the number of constraints and variables is a polynomial function of the number of vertices (cities). One of the most well-known polynomial formulations of TSPs is given by Miller-Tucker-Zemlin (MTZ) [13]. The PCBIRP cannot directly be modeled as the original MTZ formulation because the original MTZ formulation does not take into consideration the precedence relationship between vertices.

On the basis of MTZ formulation, we firstly provide the following new formulation, called PCBIRP-MTZ, in which the precedence constraints with respect to alignment marks and test positions are added to the original MTZ formulation:

**PCBIRP-MTZ:**

$$\begin{aligned}
(1) \quad & \text{minimize} && \sum_{i \in N \cup \{0\}} \sum_{\substack{j \in N \cup \{0\} \\ (j \neq i)}} c_{ij} x_{ij} \\
(2) \quad & \text{subject to} && \sum_{\substack{j \in N \cup \{0\} \\ (j \neq i)}} x_{ij} = 1, \quad \forall i \in N \cup \{0\} \\
(3) \quad & && \sum_{\substack{i \in N \cup \{0\} \\ (i \neq j)}} x_{ij} = 1, \quad \forall j \in N \cup \{0\} \\
(4) \quad & && u_j \geq u_i + 1 - (n-1)(1-x_{ij}), \quad \forall i, j \in N, \quad i \neq j \\
(5) \quad & && 1 \leq u_j \leq n-1, \quad \forall j \in N \\
(6) \quad & && u_i \leq u_j - 1, \quad \forall i \in A_p, \quad \forall j \in \{t_p\}, \quad \forall p \in B \\
(7) \quad & && x_{ij} \in \{0, 1\}, \quad \forall i, j \in N \cup \{0\}, \quad i \neq j,
\end{aligned}$$

where (1) represents the route length. Constraints (2) and (3) impose that the out-degree and in-degree of each vertex, respectively, is equal to one. Constraints (4) prevent subtours not containing node 0 and imply  $u_j \geq u_i + 1$  whenever  $x_{ij} = 1$ , where  $u_i, i \in N$  is an arbitrary real number representing the order of vertex  $i$  in the optimal tour. Together with (2) and (3), constraints (4) guarantee that subtours containing node 0 are not allowed. Constraints (6) guarantee that all the alignment marks  $i \in A_p$  of the  $p$ -th PCB are visited before the corresponding test position  $j \in \{t_p\}$  is visited.

We propose another formulation which is an extended version of PCBIRP-MTZ. Desrochers and Laporte [6] proposed a formulation in which the MTZ constraints were lifted into facets of the underlying TSP polytope. Along this line, we provide the following new formulation, called PCBIRP-DL, which is obtained by replacing constraints (4) and (5) in PCBIRP-MTZ with the lifted constraints (11)-(13):

**PCBIRP-DL:**

$$\begin{aligned}
(8) \quad & \text{minimize} && \sum_{i \in N \cup \{0\}} \sum_{\substack{j \in N \cup \{0\} \\ (j \neq i)}} c_{ij} x_{ij} \\
(9) \quad & \text{subject to} && \sum_{\substack{j \in N \cup \{0\} \\ (j \neq i)}} x_{ij} = 1, \quad \forall i \in N \cup \{0\} \\
(10) \quad & && \sum_{\substack{i \in N \cup \{0\} \\ (i \neq j)}} x_{ij} = 1, \quad \forall j \in N \cup \{0\} \\
(11) \quad & && u_j \geq u_i + 1 - (n-1)(1-x_{ij}) + (n-3)x_{ji}, \quad \forall i, j \in N, \quad i \neq j \\
(12) \quad & && 1 + (1-x_{0j}) + (n-3)x_{j0} \leq u_j, \quad \forall j \in N \\
(13) \quad & && u_j \leq (n-1) - (n-3)x_{0j} - (1-x_{j0}), \quad \forall j \in N, \\
(14) \quad & && u_i \leq u_j - 1, \quad \forall i \in A_p, \quad \forall j \in \{t_p\}, \quad \forall p \in B \\
(15) \quad & && x_{ij} \in \{0, 1\}, \quad \forall i, j \in N \cup \{0\}, \quad i \neq j,
\end{aligned}$$

where constraints (11) are obtained by lifting (4), and constraints (12) and (13) are obtained by lifting (5). By introducing lifted constraints (11)-(13), PCBIRP-DL is a tighter formulation than PCBIRP-MTZ, which means that the formulation of PCBIRP-DL is expected to solve PCBIRPs faster than the formulation of PCBIRP-MTZ.

**4 Numerical Experiments** In order to compare the performances of the proposed two formulations, we solved benchmark instances constructed based on real PCB wiring patterns. In the benchmark instances, the numbers of wiring patterns on a PCB sheet are between 1 and 21. For all instances, there are two alignment marks. Tables 1 and 2 show the experimental results. We use a personal computer with Intel Core i5 Processor 2.5 GHz , RAM:8GB, OS:Windows 7 (64bit), and the coding was done with Python 2.7 + PuLP 1.6.0.

To solve benchmark instances, we used three mathematical programming solvers, CPLEX [4], Gurobi [8] and SCIP [16], and compared their performances (computational times). Table 1 shows the computational times for the different-size instances when two proposed formulations are solved by using three well-known solvers, CPLEX 12.6.1.0, Gurobi 6.0 and SCIP 3.1.1.1. Each figure represents the computational time for obtaining an optimal solution. Bold figures express the best one among three solvers. We started to solve the minimum-size problem (the case of  $n = 1$ ), and increment the sizes of problems to be solved in such a manner as  $n = 1, n = 2, \dots$ . When the computational time for solving the problem exceeded one hour (3,600 seconds), we stopped solving larger-size problems.

Table 1: Computational time for solving different-size benchmark instances

$n$	PCBIRP-MTZ			PCBIRP-DL		
	CPLEX	Gurobi	SCIP	CPLEX	Gurobi	SCIP
1	0.01	0.02	<b>0.001</b>	<b>0.001</b>	<b>0.001</b>	<b>0.001</b>
2	0.01	0.08	0.05	<b>0.001</b>	<b>0.001</b>	<b>0.001</b>
3	0.06	0.09	0.64	<b>0.001</b>	<b>0.001</b>	0.02
4	0.31	0.70	1.73	0.19	<b>0.03</b>	0.09
5	0.67	0.64	9.50	<b>0.09</b>	0.11	0.79
6	3.81	6.46	52.59	0.14	<b>0.11</b>	1.11
7	8.50	26.60	189.62	0.17	<b>0.16</b>	3.12
8	14.90	15.21	45.20	0.25	<b>0.16</b>	3.05
9	158.11	677.69	3989.33	0.98	<b>0.78</b>	9.58
10	149.14	143.06		0.69	<b>0.50</b>	7.72
11	947.63	1328.30		5.43	<b>2.64</b>	14.43
12	724.00	1079.55		<b>5.51</b>	8.01	31.76
13	4030.57	53046.92		<b>30.05</b>	40.39	119.07
14				101.24	<b>65.57</b>	1014.06
15				107.30	<b>37.86</b>	441.24
16				156.87	<b>155.20</b>	2684.08
17				326.53	<b>174.30</b>	12498.04
18				247.73	<b>109.28</b>	
19				<b>1687.96</b>	4883.33	
20				<b>1467.39</b>		
21				<b>6017.80</b>		

Figure 16 shows the results of Table 1 graphically. The vertical axis of the plot is scaled logarithmically by taking logs to base 10, namely,  $\log_{10}(x)$ . When the formulation of PCBIRP-MTZ was used, CPLEX was the fastest among three solvers, and SCIP was the slowest one. On the other hand, when PCBIRP-DL was used, CPLEX and Gurobi were competitive. In both formulations, the computational time increases exponentially. Since the most typical number of PCBs in a sheet is from 9 to 16, the formulation of PCBIRP-DL

can be considered practical when CPLEX or Gurobi is used because it takes less than 3 minutes to solve the problem with 16 PCBs in a sheet.

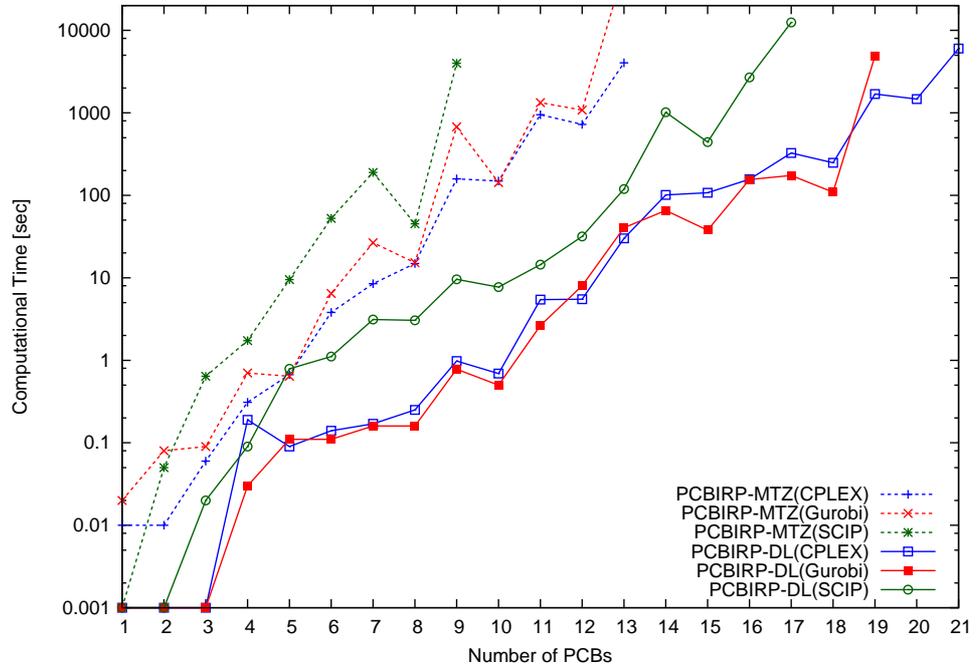


Figure 16: Computational time for solving different-size benchmark instances

Table 2 shows the route length of the optimal solution obtained using mathematical programming solvers. In this table, "Simple route" represents the simple PCB inspection route in which all the alignment marks are firstly visited and then all the test positions are visited in a simple order, like the route as shown in Figure 13. It is observed from Table 2 that the PCB inspection routes obtained by the proposed formulations are averagely around 50% shorter than simple routes that had been previously employed in the field of PCB inspections with alignment operations.

**5 Conclusion** In this paper, we have newly modeled a route optimization problem in advanced PCB electrical inspections, which had not been discussed so far. We have formulated the PCB inspection route optimization problem (PCBIRP) as a class of PDTSPs, and provided two types of mixed 0-1 integer programming problem formulations based on MTZ formulation and its extension. Some experiments have been conducted using benchmark instances based on real PCB wiring patterns. The proposed method can yield averagely 50% shorter inspection route than the previous method. Also, it has been shown that the procedure of "lifting" is promising for solving the PCBIRP.

As a future study, we will address other alternative polynomial formulations (formulations in which the number of constraints and variables is a polynomial function of the number of vertices) of PCBIRP using flow-based formulations [17]. In addition, it is interesting to consider branch-and-cut methods by extending the polytope of PDTSPs [7]. Since the number of PCBs is 100 to 200 in some cases, it is also important to consider some efficient heuristic algorithms. Hence, another future work is to consider novel heuristic

Table 2: The lengths of PCB inspection routes

$n$	Simple route	Optimal route	Improvement rate (%)
1	<b>647.1</b>	<b>647.1</b>	0.0
2	1062.2	<b>762.2</b>	28.2
3	1478.2	<b>895.3</b>	39.4
4	1894.8	<b>1057.4</b>	44.2
5	2311.6	<b>1198.8</b>	48.1
6	2640.0	<b>1380.4</b>	47.7
7	2855.4	<b>1459.4</b>	48.9
8	3072.2	<b>1531.2</b>	50.2
9	3290.9	<b>1693.7</b>	48.5
10	3512.9	<b>1808.0</b>	48.5
11	3878.2	<b>2050.0</b>	47.1
12	4277.6	<b>2114.4</b>	50.6
13	4681.7	<b>2225.7</b>	52.5
14	5089.2	<b>2322.7</b>	54.3
15	5498.9	<b>2383.6</b>	56.7
16	5850.7	<b>2570.7</b>	56.1
17	6074.9	<b>2644.2</b>	56.4
18	6302.1	<b>2715.9</b>	56.9
19	6533.2	<b>2883.2</b>	55.9
20	6769.6	<b>2992.8</b>	55.8
21	7148.0	<b>3241.9</b>	54.6

algorithms as some extensions of conventional efficient algorithms such as Lin-Kernighan method [12] and its variants [9]. These extensions will be discussed elsewhere in near future.

## 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] M. Ancău, "The optimization of printed circuit board manufacturing by improving the drilling process productivity," *Computers & Industrial Engineering*, Vol. 55, No. 2, pp. 279–294, 2008.
- [3] G. Berbeglia, J.F. Cordeau, I. Grinbkovskaia, G. Laporte, "Static pickup and delivery problems: a classification scheme and survey," *Top*, Vol. 15, No. 1, pp. 1–31, 2007.
- [4] IBM ILOG CPLEX Optimizer: <http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/>
- [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] M. Desrochers, G. Laporte, "Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints," *Operations Research Letters*, Vol. 10, No. 1, pp. 27–36, 1991.
- [7] I. Dumitrescu, S. Ropke, J.-F. Cordeau, G. Laporte, "The traveling salesman problem with pickup and delivery: polyhedral results and a branch-and-cut algorithm," *Mathematical Programming*, Vol. 121, No. 2, pp. 269–305, 2010.
- [8] Gurobi Optimizer: <http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/>
- [9] K. Helsgaun, "An effective implementation of the Lin-Kernighan traveling salesman heuristic," *European Journal of Operational Research*, Vol. 126, No. 1, pp. 106–130, 2000.
- [10] A.B. Kahng, G. Robins, E.A. Walkup, "Optimal algorithms for substrate testing in multi-chip modules," *International Journal of High Speed Electronics and Systems*, Vol. 6, No. 4, pp. 595–612, 1995.
- [11] R. Kumar, H. Li, "Integer programming approach to printed circuit board assembly time optimization," *IEEE Transactions on Components, Packaging and Manufacturing Technology, Part B: Advanced Packaging*, Vol. 18, No. 4, pp. 720–727, 1995.
- [12] S. Lin, B.W. Kernighan, "An effective heuristic algorithm for the traveling-salesman program," *Operations Research*, Vol. 21, No. 2, pp. 498–516, 1973.
- [13] C.E. Miller, A. W. Tucker, R.A. Zemlin, "Integer programming formulation of traveling salesman problems," *Journal of the ACM*, Vol. 7, No. 4, pp. 326–329, 1960.
- [14] K. Murakami, "Formulation and heuristic algorithms for multi-chip module substrate testing," *Computers and Electrical Engineering*, Vol. 39, No. 4, pp. 1049–1060, 2013.
- [15] S.N. Parragh, K.F. Doerner, R.F. Hartl, "A survey on pickup and delivery problems," *Journal für Betriebswirtschaft*, Vol. 58, No. 1, pp. 21–51, 2008.
- [16] SCIP (Solving Constraint Integer Programs): <http://scip.zib.de/>
- [17] H.D. Sherali, S.C. Sarin, P.-F. Tsai, "A class of lifted path and flow-based formulation for the asymmetric traveling salesman problem with and without precedence constraints," *Discrete Optimization*, Vol. 3, No. 1, pp. 20–32, 2006.

Communicated by *Hiroaki Ishii*