mathematics

MDPI

Article

Mixed-Integer Linear Programming Model and Heuristic for Short-Term Scheduling of Pressing Process in Multi-Layer Printed Circuit Board Manufacturing

Teeradech Laisupannawong [1], Boonyarit Intiyot [1,*] and Chawalit Jeenanunta [2]

[1] Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University, Bangkok 10330, Thailand; teeradech.lai@gmail.com
[2] School of Management Technology, Sirindhorn International Institute of Technology (SIIT), Thammasat University, Pathum Thani 12120, Thailand; chawalit@siit.tu.ac.th
* Correspondence: boonyarit.i@chula.ac.th

Abstract: The main stages of printed circuit board (PCB) manufacturing are the design, fabrication, assembly, and testing. This paper focuses on the scheduling of the pressing process, which is a part of the fabrication process of a multi-layer PCB and is a new application since it has never been investigated in the literature. A novel mixed-integer linear programming (MILP) formulation for short-term scheduling of the pressing process is presented. The objective function is to minimize the makespan of the overall process. Moreover, a three-phase-PCB-pressing heuristic (3P-PCB-PH) for short-term scheduling of the pressing process is also presented. To illustrate the proposed MILP model and 3P-PCB-PH, the test problems generated from the real data acquired from a PCB company are solved. The results show that the proposed MILP model can find an optimal schedule for all small- and medium-sized problems but can do so only for some large-sized problems using the CPLEX solver within a time limit of 2 h. However, the proposed 3P-PCB-PH could find an optimal schedule for all problems that the MILP could find using much less computational time. Furthermore, it can also quickly find a near-optimal schedule for other large-sized problems that the MILP could not solved optimally.

Keywords: pressing process; printed circuit board; scheduling; mixed-integer linear programming; heuristic

Citation: Laisupannawong, T.; Intiyot, B.; Jeenanunta, C. Mixed-Integer Linear Programming Model and Heuristic for Short-Term Scheduling of Pressing Process in Multi-Layer Printed Circuit Board Manufacturing. *Mathematics* **2021**, *9*, 653. https://doi.org/10.3390/math9060653

Academic Editor: Frank Werner

Received: 9 February 2021
Accepted: 16 March 2021
Published: 18 March 2021

1. Introduction

A printed circuit board (PCB) is a major component in most electronics, such as televisions, mobile phones, digital cameras, computers, and medical devices. The manufacturing of PCBs has become a competitive industry due to the increased demand for electronic products. The PCBs can be classified into three types, according to the number of their layers, as single-layer PCBs, double-layer PCBs, and multi-layer PCBs.

According to Khandpur [1], PCB manufacturing consists of the design, fabrication, assembly, and testing. The PCB design is the process of creating a circuit schematic by PCB designers. Then, PCB fabrication is the process of constructing the PCB (bare board) before placing electronic components in the PCB assembly. The fabrication of each type of PCB is different. In this paper, we consider only the fabrication of multi-layer PCBs. As stated in Reference [1], the main materials used in multi-layer PCB fabrication include the copper-clad laminate sheets and prepregs. The fabrication of multi-layer PCBs can be summarized in the following five steps:

1. The laminate sheets are cut to the required size in the cutting process.
2. The circuit pattern is created on the cut laminate in the etching process.
3. A number of etched laminates (or cores) are stacked together with a prepreg inserted between each pair of them. The stack (or panel) is pressed using heat and pressure in the pressing process.

4. Holes will be drilled in the pressed board in the drilling process and the circuit pattern will be made on the outer surfaces.
5. The remaining steps are the quality control and labeling processes.

Figure 1 shows a schematic representation of the steps in multi-layer PCB fabrication. A major cost-consuming process in multi-layer PCB fabrication is the cutting process. Most PCB companies aim to cut the laminates so that the waste areas from cutting the laminates are minimized. This process could be formulated as a two-dimensional cutting stock problem (2DCSP). The drilling process is another time-consuming process in the multi-layer PCB fabrication. Most PCB companies aim to find an optimal path for drilling the holes in the designed positions in the circuit pattern so that the travel time or distance of the drilling device is minimized, and so the overall processing time is reduced. A mathematical problem that relates to the drilling process is the hole drill routing optimization problem (HDROP).

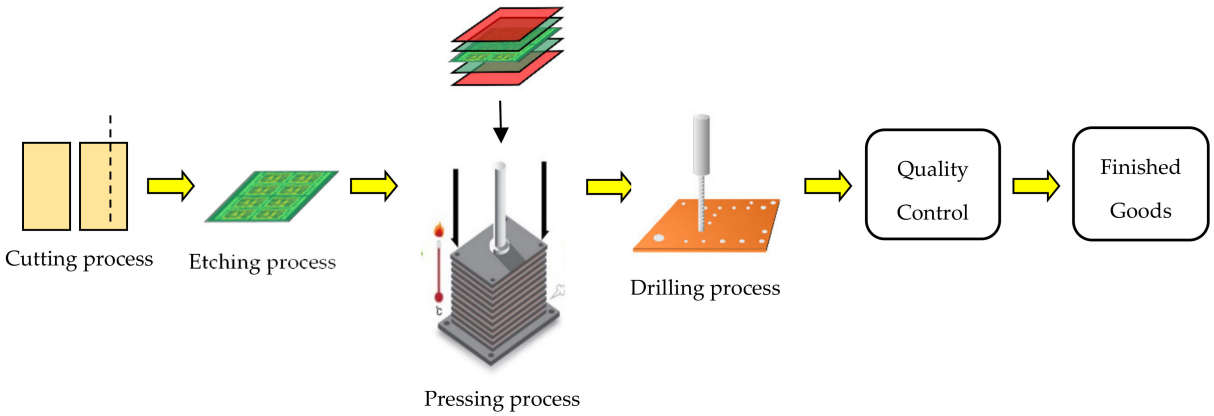

Figure 1. The steps of multi-layer printed circuit board (PCB) fabrication.

There have been many research studies reported on the 2DCSP and HDROP, where diverse techniques have been used to solve the 2DCSP, such as an integer linear programming model using a column generation technique [2], an exact arc-flow model [3], a branch-and-price algorithm [4], and heuristic algorithms based on column generation [5,6]. There are some reports on the cutting process that have used real data from PCB companies, such as in References [7,8]. As for the HDROP, numerous research studies have been developed to solve it, such as a particle swarm optimization (PSO) [9], an ant colony system [10], a cuckoo search algorithm [11], and a hybridized cuckoo search-genetic algorithm [12].

The PCB assembly is the process of placing electronic components, such as resistors, capacitors, and transistors, at the specified location on a bare board. In a PCB assembly line, there are many placement machines with different unit assembly times for the same component. A board is passed through all the machines to complete the component placement. Therefore, the components should be allocated to appropriate machines so that the assembly time is minimized. This leads to the problem of getting an optimal workload balance in the PCB assembly line [13–16]. The aim of this problem is to minimize the production cycle time of the assembly line for a given PCB type, which is the maximum time needed by one of the placement machines. Some techniques have been proposed to solve this problem, such as using a genetic algorithm [13] and a branch-and-bound-based optimization algorithm [14]. Some extended problems with additional constraints can be found in the literature, such as the use of feeder modules, precedence constraints between components, and feeder duplications [15], as well as an integrated workload balancing and single-machine optimization problem [16].

The testing is the process after assembling all components to the board. Environmental stress-screening chambers are commonly used to test PCBs to identify early fallouts before they are used in the field. The chamber can process multiple PCBs simultaneously, i.e., the PCBs are processed in batches. Therefore, the process of PCB testing can be considered

as the batch-processing machine scheduling (BPMS) problem and has been addressed extensively in the literature. For example, a simulated annealing approach was proposed to minimize the makespan of a single BPMS problem [17]. A PSO algorithm was presented to minimize the makespan when scheduling non-identical parallel batch-processing machines [18]. A simulation-based intelligence optimization method was developed to minimize the makespan of a flow-shop scheduling problem with multiple heterogeneous batch-processing machines [19]. In addition, a PSO algorithm was presented to minimize the total weighted tardiness of non-identical parallel BPMS problems [20].

This paper focuses on the pressing process, which is also another time- and cost-consuming process in multi-layer PCB manufacturing. The pressing process, a stage in multi-layer PCB fabrication, consists of many phases that require a lot of materials and expensive machines. A good schedule is needed to reduce the production time and to increase the machine utilization, which requires effective assignment and scheduling. After extensively reviewing the literature on the scheduling problems that relate to PCB manufacturing, we have not found any studies linked to the scheduling of the pressing process. A similar mathematical problem in the literature is the flexible job shop scheduling problem (FJSP) [21–24]. The pressing process scheduling and FJSP have similar backgrounds, which are assignment and sequencing. In the FJSP, there are an operation-to-machine assignment and sequencing operation in each machine, but the pressing process scheduling has more than one stage of the assignment. In practice, most PCB companies manually schedule the pressing process, which may not yield the best resource utilization. Therefore, this paper aims to provide a mathematical model for scheduling the pressing process that maximizes the resource utilization. Furthermore, due to the complexity of the pressing process, an effective heuristic algorithm for solving this problem is also presented.

Novelties of the Paper

This paper investigates the pressing process scheduling, which is an application in real-world PCB industries, and, to the best of our knowledge, has never been investigated in the literature. Some PCB companies usually schedule the pressing process by dividing the planning horizon into fixed time intervals. Then, each time interval is assigned either to be in a cycle of a machine or to be vacant. However, this may not be the best way of scheduling the pressing process since in reality, the starting time and completion time of a cycle do not need to follow the fixed time intervals. It is more flexible if the starting and completion times of the cycles are considered as continuous variables. The contributions of this paper can be summarized as follows:

1. This paper proposes a novel mixed integer linear programming (MILP) model for the pressing process scheduling that can find an optimal schedule to meet the objective of maximizing the resource utilization, while the times are continuous values.
2. This paper presents a three-phase-PCB-pressing heuristic algorithm (3P-PCB-PH) for solving the pressing process scheduling, based on the proposed MILP, which can find a near-optimal solution within a reasonable computational time and is practical for real-life applications.

The remainder of this paper is organized as follows. In Section 2, the problem description of the pressing process scheduling is introduced. The proposed MILP model is presented in Section 3, while the 3P-PCB-PH algorithm is presented in Section 4. Numerical examples are shown in Section 5. The discussions and the conclusions are drawn in Sections 6 and 7, respectively.

2. Problem Description

This section explains the pressing process in multi-layer PCB manufacturing. The aim of the pressing process is to press the panel that consists of copper foils, prepregs, and core(s), and is shown schematically in Figure 2.

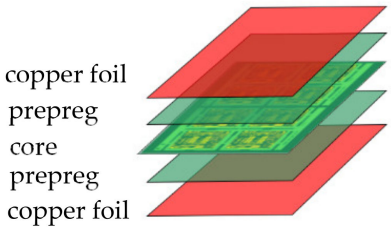

Figure 2. An example of a panel.

The overall processes of one cycle of a press machine are shown schematically in Figure 3. A single cycle of a press machine takes 360 min, which includes the following three phases:

1. Lay-up process phase: The panels are arranged on a selected stainless-steel template (SST), where the number of panels on the SST depends on the size of the SST, the gap between each panel on the SST, and the pattern layout of arrangement. The final arrangement of panels on a SST is called a book. Then, each book is loaded into slots (openings) of a press machine. The number of loaded books is equal to the number of openings of the press machine. This phase takes 120 min.
2. Pressing process phase: The press machine that is already loaded with books is sent into an oven, where the books are heated and pressed. After 120 min, the press machine is removed from the oven.
3. Cool-down process phase: The pressed books in the press machine are cooled down for 120 min. Finally, the books will be removed from the press machine to complete one cycle of the press machine.

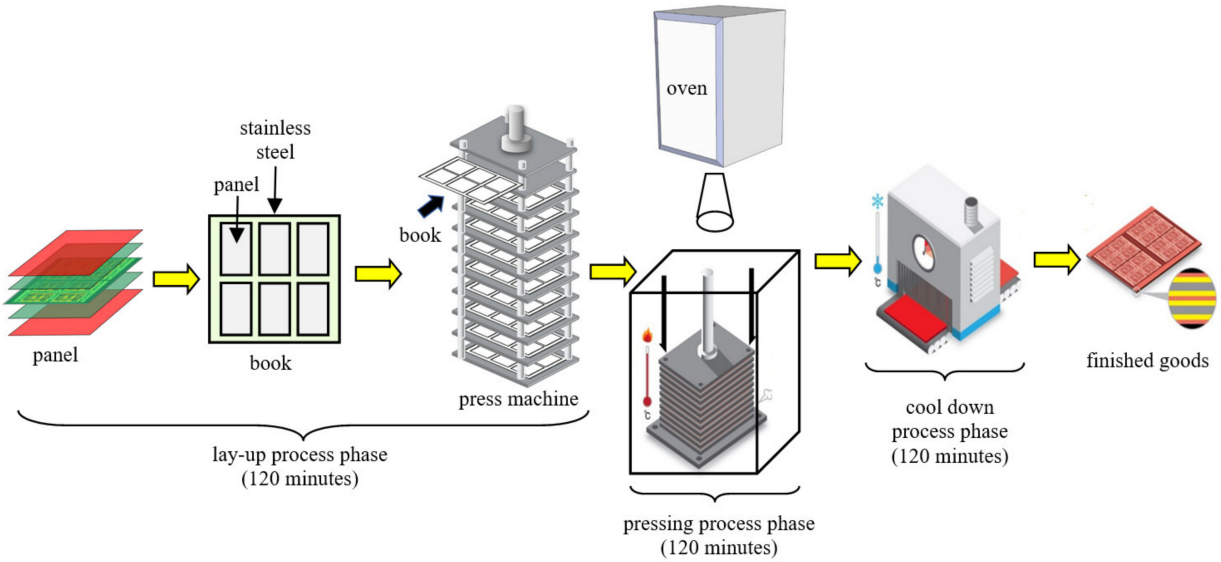

Figure 3. Schematic diagram showing one cycle of a press machine.

Note that, after a press machine has finished one cycle, it is immediately available for a new cycle. Similarly, an oven is immediately available for another press machine after finishing the pressing process phase. Moreover, the following assumptions are made:

- The three phases of a press machine cycle must be performed continuously (no idle time between phases).
- The number of press machines and ovens are known, and the number of ovens is less than the number of press machines. This is because the cost of an oven is very high, and hence the company usually has a small number of ovens.
- Each press machine has the same number of openings.

- There are many types of panels to press and the demand of each type of panel is given. The type of panel depends on the customer's design.
- All panels can be finished within the given due date and resources, i.e., the demands of panels, which are inputs from the customer, yield a feasible schedule.
- The maximum number of available cycles of each press machine to be operated within the due date is the same and this value is given. In practice, the production planning department can estimate this value from the order of the customers and the available resources.
- There are many sizes of SSTs, and each size is unlimitedly available.
- A layout is a pattern of arrangement of panels on a SST. In this study, there are eight layouts, as shown in Figure 4. For example, Figure 4a illustrates the layout with two horizontal sections and the panels are arranged vertically in each section.
- The inner gap is the minimal gap among two panels in a book and the outer gap is the minimal gap between each panel and the borders of the SST. The inner gap (g) and outer gap (G) of an arrangement of panels on a SST depend on the type of panel and these values are known.

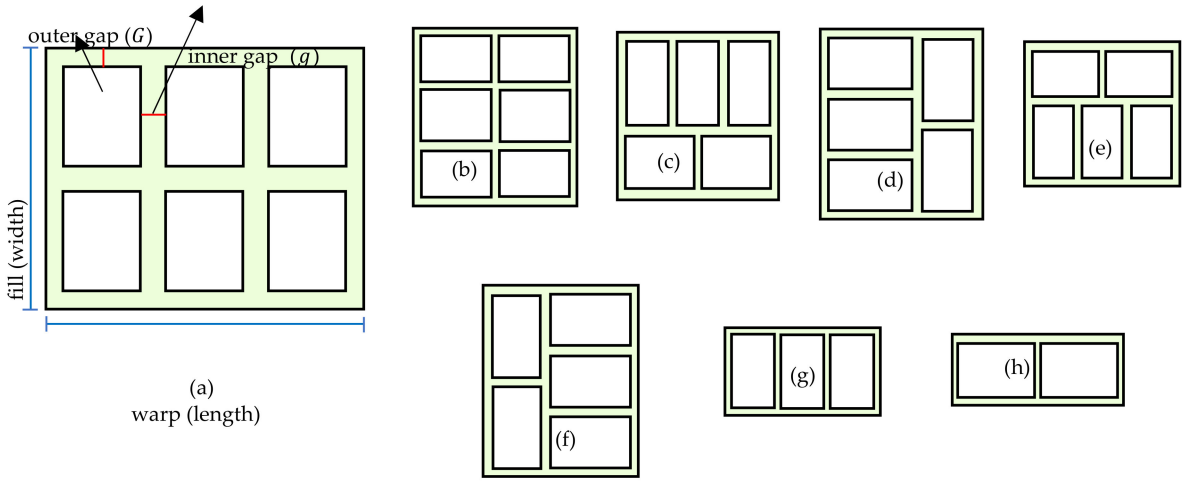

Figure 4. Illustration of the eight layouts (**a**–**h**) of the panel arrangement.

Note that Figure 4a–h are meant to show only the direction of the panel arrangement on a SST, and the number of panels in a book is not limited to those shown in the illustration. In fact, the actual number of panels on a SST using a given layout depends on the size of panel, the size of the SST, and the gaps. Normally, each PCB company may have its own formula for computing the number of panels on a SST with a layout.

The four principal constraints for the pressing process are as follows:

1. Only one type of panel can be arranged and pressed in a cycle of a press machine.
2. The books that are inserted in the same press machine must have the same layout and the same SST size.
3. Each oven can be used by only one press machine at a time to operate the pressing process phase.
4. The number of finished goods of each type of panel must be greater than or equal to the demand.

Constraints 1 and 2 are required so that the pressure from the press machine will be equally distributed to each panel. The objective of the process is to maximize utilization of all press machines and ovens.

3. Proposed Mathematical Model

This section presents a MILP model for scheduling the pressing process as described in Section 2. The indices, sets, parameters, and variables used in the proposed model are defined below.

Indices:

i The index of panel types.
k The index of SST sizes.
l The index of layouts.
p The index of press machines.
o The index of ovens.
t The index of cycles of a press machine.

Parameters:

I The number of types of panels.
K The number of all SST sizes.
L The number of layouts.
P The number of press machines.
O The number of ovens.
T The maximum number of available cycles of each press machine.
m The number of openings of each press machine.
n The processing time of each phase in the pressing process, i.e., the lay-up, pressing, and cool-down process phases. In our case, $n = 120$ min.
a_{ikl} The number of panels of type i per opening using stainless size k and layout l.
d_i Total demand of panel type i.
M A big positive number.

Sets:

$\hat{I}$ The set of all types of panels, $\hat{I} = \{1, 2, \ldots, I\}$.
$\hat{K}$ The set of all SST sizes, $\hat{K} = \{1, 2, \ldots, K\}$.
$\hat{L}$ The set of all layouts, $\hat{L} = \{1, 2, \ldots, L\}$.
$\hat{P}$ The set of all press machines, $\hat{P} = \{1, 2, \ldots, P\}$.
$\hat{O}$ The set of all ovens, $\hat{O} = \{1, 2, \ldots, O\}$.
$\hat{T}$ The set of all numbers of available cycles of each press machine, $\hat{T} = \{1, 2, \ldots, T\}$.

Decision variables:

x_{iklpt} 1, if panel type i is assigned with SST size k and layout l to press machine p at cycle t.
X_{pto} 1, if press machine p is put in oven o at cycle t.
$Y_{ptp't'o}$ 1, if cycle t of press machine p precedes cycle $t\prime$ of press machine $p\prime$ in oven o.
A_{pt} The starting time of the lay-up process phase in cycle t of press machine p.
B_{pto} The starting time of the pressing process phase in cycle t of press machine p in oven o.
C_{pt} The completion time of cycle t of press machine p.
D_{pto} The completion time of the pressing process phase in cycle t of press machine p in oven o.
C'_{pt} The auxiliary variable, which is equal to C_{pt} if there are a panel, a SST, and a layout assigned in press machine p at cycle t. Otherwise, it is equal to 0.
C_{max} The maximum completion time of the last cycle of all press machines which operate the pressing process, i.e., the makespan of the overall process.

In this model, the variable $Y_{ptp't'o}$ is a precedence binary variable that is only defined when $p \neq p\prime$. It is used to avoid the case where an oven operates the pressing process phase for more than one press machine at the same time. This variable is adapted from the precedence binary variable $Y_{iji'j'k}$ that is used to handle the sequencing operations on a machine in the mathematical model of the flexible job shop scheduling problem in Reference [21]. The proposed MILP model can be stated as follows:

$$Min \quad C_{max} \tag{1}$$

Subject to:

$$\sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{l=1}^{L} x_{iklpt} \le 1,\ \forall p \in \hat{P},\ \forall t \in \hat{T}, \tag{2}$$

$$x_{iklpt} \le a_{ikl},\ \forall i \in \hat{I},\ \forall k \in \hat{K},\ \forall l \in \hat{L},\ \forall p \in \hat{P},\ \forall t \in \hat{T}, \tag{3}$$

$$\sum_{k=1}^{K}\sum_{l=1}^{L}\sum_{p=1}^{P}\sum_{t=1}^{T} x_{iklpt}(ma_{ikl}) \ge d_i,\ \forall i \in \hat{I}, \tag{4}$$

$$\sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{l=1}^{L} x_{iklp(t-1)} \ge \sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{l=1}^{L} x_{iklpt},\ \forall p \in \hat{P},\ \forall t \in \hat{T} - \{1\}, \tag{5}$$

$$\sum_{o=1}^{O} X_{pto} = 1,\ \forall p \in \hat{P},\ \forall t \in \hat{T}, \tag{6}$$

$$B_{pto} + D_{pto} \le \left(X_{pto}\right)M,\ \forall p \in \hat{P},\ \forall t \in \hat{T},\ \forall o \in \hat{O}, \tag{7}$$

$$A_{p,t} \ge C_{p,t-1},\ \forall p \in \hat{P},\ \forall t \in \hat{T} - \{1\}, \tag{8}$$

$$\sum_{o=1}^{O} B_{pto} = A_{pt} + n,\ \forall p \in \hat{P},\ \forall t \in \hat{T}, \tag{9}$$

$$C_{pt} = A_{pt} + 3n,\ \forall p \in \hat{P},\ \forall t \in \hat{T}, \tag{10}$$

$$\left(B_{pto} + n\right) - \left(1 - X_{pto}\right)M \le D_{pto},\ \forall p \in \hat{P},\ \forall t \in \hat{T},\ \forall o \in \hat{O}, \tag{11}$$

$$D_{pto} \le \left(B_{pto} + n\right) + \left(1 - X_{pto}\right)M,\ \forall p \in \hat{P},\ \forall t \in \hat{T},\ \forall o \in \hat{O}, \tag{12}$$

$$B_{pto} \ge D_{p't'o} - \left(Y_{ptp't'o}\right)M,\ \forall p, p' \in \hat{P},\ p \ne p',\ \forall t, t' \in \hat{T},\ \forall o \in \hat{O}, \tag{13}$$

$$B_{p't'o} \ge D_{pto} - \left(1 - Y_{ptp't'o}\right)M,\ \forall p, p' \in \hat{P},\ p \ne p',\ \forall t, t' \in \hat{T},\ \forall o \in \hat{O}, \tag{14}$$

$$C_{pt} - M\left[1 - \sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{l=1}^{L} x_{iklpt}\right] \le C'_{pt},\ \forall p \in \hat{P},\ \forall t \in \hat{T}, \tag{15}$$

$$C'_{pt} \le C_{pt} + M\left[1 - \sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{l=1}^{L} x_{iklpt}\right],\ \forall p \in \hat{P},\ \forall t \in \hat{T}, \tag{16}$$

$$C'_{pt} \le M\left(\sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{l=1}^{L} x_{iklpt}\right),\ \forall p \in \hat{P},\ \forall t \in \hat{T}, \tag{17}$$

$$C_{max} \ge C'_{pt},\ \forall p \in \hat{P},\ \forall t \in \hat{T}, \tag{18}$$

and,

$x_{iklpt} \in \{0,1\}$ $\quad \forall i \in \hat{I},\ \forall k \in \hat{K},\ \forall l \in \hat{L},\ \forall p \in \hat{P},\ \forall t \in \hat{T},$
$X_{pto} \in \{0,1\}$ $\quad \forall p \in \hat{P},\ \forall t \in \hat{T}, \forall o \in \hat{O},$
$Y_{ptp't'o} \in \{0,1\}$ $\quad \forall p, p' \in \hat{P},\ p \ne p',\ \forall t, t' \in \hat{T},\ \forall o \in \hat{O},$
$A_{pt}, C_{pt} \ge 0$ $\quad \forall p \in \hat{P},\ \forall t \in \hat{T},$
$B_{pto},\ D_{pto} \ge 0$ $\quad \forall p \in \hat{P},\ \forall t \in \hat{T}, \forall o \in \hat{O},$
$C'_{pt} \ge 0$ $\quad \forall p \in \hat{P},\ \forall t \in \hat{T},$
$C_{max} \ge 0$

The objective function (1) is to minimize the makespan of the overall process. This can imply maximizing the utilization of all resources.

Constraint (2) is the panel-SST-layout assignment constraint. It is used to ensure that at most one panel type, one SST size, and one layout can be assigned in each cycle of each press machine. If there is an assignment of a panel type, a SST size, and a layout in a cycle of a press machine, it is assumed that these must be the same in all openings.

Constraint (3) is the panel-SST-size-layout compatibility constraint. If panel type i cannot use SST size k with layout l ($a_{ikl} = 0$), then constraint (3) ensures that this pattern cannot be assigned to any press machine p and any cycle t.

Constraint (4) is the demand constraint. It requires that the total outputs of each type of panel from all openings, all cycles, and all press machines must satisfy the demand.

Constraint (5) enforces that a panel type, a SST size, and a layout must be assigned in a press machine at cycle $t-1$ before cycle t. This helps push empty cycles (the cycles of a press machine with no panel assignment) to be after the cycles with a panel assignment.

Constraint (6) is the press machine assignment constraint. It is used to ensure that each cycle of each press machine must be assigned to one oven only.

Constraint (7) enforces that if cycle t of press machine p is assigned to oven o, then the starting time and completion time of the pressing process phase in cycle t of press machine p in oven o can be any non-negative value. Otherwise, these are set to be 0.

Constraint (8) makes sure that any cycle of a press machine can be started after the previous cycle has been finished.

Constraint (9) sets the starting time of the pressing process phase in cycle t of press machine p in its assigned oven to be equal to the starting time of this cycle of press machine p plus the processing time n that it takes in the lay-up process phase.

Constraint (10) sets the completion time of cycle t of press machine p to be equal to its starting time plus the processing time $3n$ (the processing time of one cycle).

Constraints (11) and (12) ensure that if $X_{pto} = 1$, the completion time of the pressing process phase in cycle t of press machine p in its assigned oven will be equal to its starting time plus the processing time n that it takes in the oven.

Constraints (13) and (14) take care of that the pressing process phase in cycle t of press machine p and the pressing process phase in cycle $t\prime$ of press machine $p\prime$, which are assigned in the same oven, cannot be done at the same time.

Constraints (15)–(17) require that if there is assignment of a panel, a SST size, and a layout in the press machine p at cycle t, then the variable C'_{pt} is equal to C_{pt}. Otherwise, it is equal to 0.

Constraint (18) determines the maximum completion time of the last cycle of press machines that has a panel assignment (non-empty cycles), which is the makespan of the overall process.

Note that for the cycle of the press machine that has no assignment of a panel, the proposed MILP model will still return its starting time (A_{pt}) and completion time (C_{pt}), which can be considered as it does not do any work (empty cycle). Also, note that the objective function (1) is to minimize the makespan of all the cycles of all the press machines that actually do the work (non-empty cycles). It means that the objective tries to minimize the makespan of all the cycles of all the press machines that are needed for the respective outputs to satisfy the demands.

The solution to the proposed MILP model provides information about the panel type, SST size, and layout that should be assigned in each cycle of a press machine. In addition, it also tells that each cycle of a press machine should be put into which oven, as well as its starting time and completion time. Hence, the proposed model can be an option to provide an optimized schedule in the pressing process of any PCB manufacturing industry.

4. Proposed 3P-PCB-PH Algorithm

Due to the complexity of the pressing process, using a mathematical programming model may not be suitable for solving a large-sized problem. This section presents a heuristic algorithm for scheduling the pressing process. The idea of this algorithm is to solve the proposed MILP model in three phases. Phase 1 consists of matching each panel type with a SST size and a layout and determining the number of cycles that is needed for the demands to be satisfied. Next, all cycles that are needed to be used are scheduled in Phase 2, which yields the number of non-empty cycles of each press machine and their starting and completion times. In Phase 3, each panel type with its selected SST

size and layout from Phase 1 is assigned to a non-empty cycle of a press machine. The parameters that are used in the proposed 3P-PCB-PH algorithm are the same as described Section 3. The details of the designed 3P-PCB-PH algorithm include Steps 1–5, which are expressed below.

Step 1: Take the inputs I, K, L, P, O, T, m, n, a_{ikl}, $\forall i \in \hat{I}$, $k \in \hat{K}$, $l \in \hat{L}$, and d_i, $\forall i \in \hat{I}$.

Step 2 (Phase 1): Selecting the SST size and layout.

In this phase, an appropriate SST size and a layout are chosen for each panel type. The inputs of Phase 1 include I, K, L, m, d_i, and a_{ikl}, $\forall i \in \hat{I}$, $k \in \hat{K}$, $l \in \hat{L}$. For each panel type i, we select a SST size $\bar{k}_i$ and a layout $\bar{l}_i$ that give the maximum number of panels of type i, say $a_{i\bar{k}_i\bar{l}_i}$. Hence, the number of produced panels of this type per cycle of a press machine is $ma_{i\bar{k}_i\bar{l}_i}$. Next, the minimum number of cycles needed for pressing each panel of type $i \in \hat{I}$ can be computed from $d_{c_i} = \left\lceil \frac{d_i}{ma_{i\bar{k}_i\bar{l}_i}} \right\rceil$ (note that the notation $\lceil x \rceil$ is the smallest integer that is greater than or equal to x). Let d_c be the sum of these values of all panel types, which is the minimum number of total cycles that are needed to be used for pressing, so that the demands of all panel types are satisfied. Note that the value d_c does not exceed the number of all available cycles $P \times T$, since we have the assumption that the demands of panels (which are inputs from the customer) yield a feasible schedule. The flowchart of the algorithm for Phase 1 is shown in Figure 5.

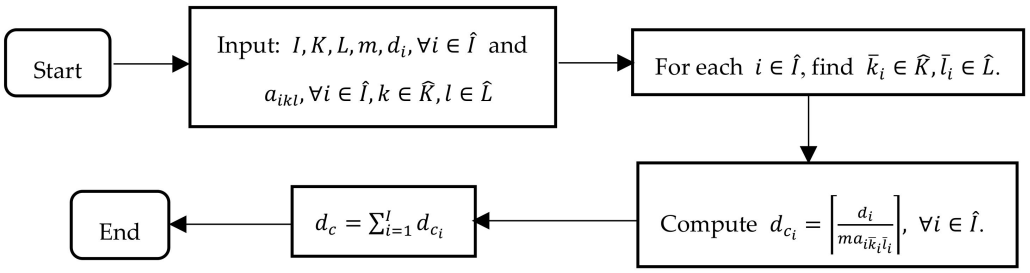

Figure 5. Flowchart for the 3P-PCB-PH algorithm for Phase 1.

Step 3 (Phase 2): Scheduling the press machines and ovens.

In this phase, all d_c cycles are distributed to all press machines to generate a schedule such that the makespan is minimized. The algorithm for Phase 2 is comprised of the following components.

1. $A = [A_{pt}]_{P \times T}$: the matrix that collects the starting time of cycle t of press machine p (the starting time of the lay-up process phase). Initially, A is set to be $[0]_{P \times T}$.
2. $C = [C_{pt}]_{P \times T}$: the matrix that collects the completion time of cycle t of press machine p. Initially, C is set to be $[0]_{P \times T}$.
3. *Can*: the candidate list represents the next earliest available cycle number to use each press machine. Initially, *Can* is set to be $[1]_{1 \times P}$, i.e., for each press machine, the cycle that is ready to start is cycle 1.
4. (*start_time*, *end_time*, *press_machine*, *cycle*): A scheduled pressing job which collects the starting and end times of the pressing process phase of a press machine at a cycle, where the *start_time*, *end_time*, *press_machine*, and *cycle* are the starting time, end time, press machine number, and cycle number, respectively. For example, if we have a scheduled pressing job (240, 360, 1, 1), it means that the pressing job occurs from time 240 to 360 min and is the task of press machine 1 at cycle 1.
5. *Oven_Schedule_List*: the list of scheduled pressing jobs to use in each oven in a sequential order. Each element in the *Oven_Schedule_List* is also a list, which collects the scheduled pressing job tuples that are assigned in the corresponding oven. Figure 6 shows an example of an *Oven_Schedule_List* when the number of ovens (O) is three and the processing time of the pressing process phase (n) is 120 min. The first list in *Oven_Schedule_List* contains the scheduled pressing jobs that are already assigned to

oven 1. There are two pressing jobs in the first list. The first is (120, 240, 1, 1), which means oven 1 has to press from 120 to 240 min and is the task of press machine 1 at cycle 1, while the second is (480, 600, 1, 2) which means oven 1 has to press from 480 to 600 min and is the task of press machine 1 at cycle 2. Similarly, the list for oven 2 has only one job that is already assigned, and there is no job that is currently assigned to oven 3 since the third list is empty. Note that, initially, the list *Oven_Schedule_List* is set to be the list of O empty lists $[\,[\,]\,]_{1\times O}$. The algorithm for Phase 2 will later populate this list with suitable jobs.

6. *Oven_Idle_time_List*: the list of idle time intervals of each oven in a sequential order. Each element in the *Oven_Idle_time_List* is also a list which collects all the idle time intervals in the corresponding oven. Initially, each oven has only one idle time interval $[0, \infty)$, indicating that no task had been assigned to it yet.

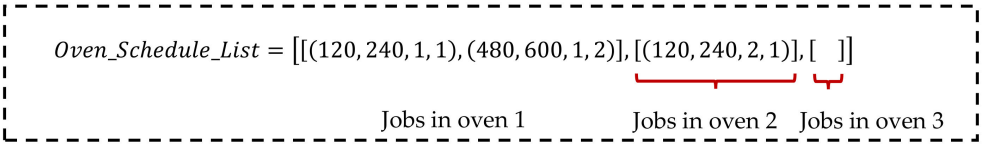

Figure 6. An example of an *Oven_Schedule_List*.

After introducing all the components, we proposed the algorithm for Phase 2 as follows. The inputs for the algorithm are $P, O,\ n$, and d_c, where d_c is used as the total number of iterations. For each iteration, a press machine with the minimum workload is selected, say $p\prime$. Next, we check whether $Can[p\prime]$, the next earliest available cycle of press machine $p\prime$, is the first cycle. If yes, the starting time of press machine $p\prime$ at cycle $Can[p\prime]$ is set to be 0. Otherwise, it is set to be the end time of the previous cycle. Let this starting time be *start_time_press_machine*. Note that this starting time is not yet a final starting time of the press machine since we need to check the feasibility with the assigned oven first. Then, the press machine $p\prime$ at cycle $Can[p\prime]$ will be assigned to the oven with the minimum workload, say $o\prime$, to operate the pressing process phase. Next, we check whether the oven $o\prime$ has been used yet. If not (i.e., the $Oven\_Idle\_time\_List[o\prime]$ has only one idle time interval $[0, \infty)$), the cycle $Can[p\prime]$ of press machine $p\prime$ can be started at *start_time_press_machine*, and sequentially, $p\prime$ is sent into the oven $o\prime$ at the time *start_time_press_machine* $+ n$. Otherwise, we consider all idle time intervals in the $Oven\_Idle\_time\_List[o']$. These intervals are examined from left to right to find the earliest time that the press machine $p\prime$ at cycle $Can[p\prime]$ can start the pressing process phase in the oven $o\prime$. An example is illustrated in Figure 7. Suppose that $o\prime$ is oven 1 that already has a task of cycle 1 from press machine 1 assigned before, and the processing time of the pressing process phase (n) is 120 min. Suppose $p\prime$ is press machine 2 and $Can[2]$ is cycle 1. Since this is the first cycle, the value *start_time_press_machine* is 0. However, since oven 1 has been used, we will examine the idle time intervals from left to right. From Figure 7, $Oven\_Idle\_time\_List[1] = [[0, 120],\ [240, \infty)]$. It is clear that the first interval [0, 120] is not feasible since the lay-up process phase has not been done. So, press machine 2 at cycle 1 can start the pressing process phase as early as possible in oven 1 at time 240 min in the second idle time interval [240, ∞). Let this time be *start_time_oven*. We can then find the time that the press machine $p\prime$ is removed from the oven $o\prime$ (*end_time_oven* $=$ *start_time_oven* $+ 120$) as well as the actual starting time (*start_time_press_machine*) and completion time of press machine $p\prime$ at cycle $Can[p\prime]$, which are the *start_time_oven* $- 120$ and *end_time_oven* $+ 120$, respectively. We update these values in matrices A and C as well as update the list $Oven\_Schedule\_List[o\prime]$ and $Oven\_Idle\_time\_List[o\prime]$. Then, $Can[p\prime]$ is incremented by 1 so that the next cycle of the press machine $p\prime$ is a new candidate. The algorithm is repeated until all d_c cycles are scheduled. The flowchart of the algorithm for Phase 2 is shown in Figure 8.

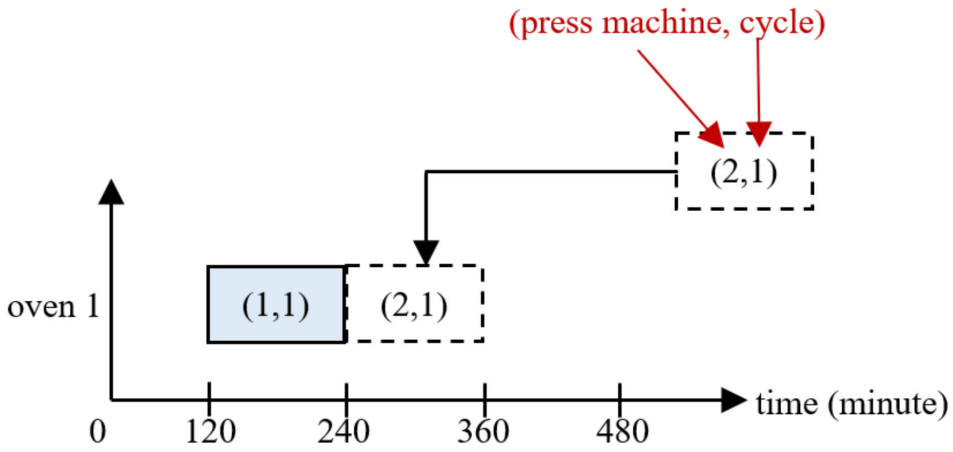

Figure 7. An example of finding *start_time_oven*.

Step 4 (Phase 3): Assigning the panel-SST-size-layout combinations to cycles of the press machines.

From Phase 2, the number of working cycles for each press machine is known. In Phase 3, each panel type with its selected SST size and layout will be assigned to a cycle of a press machine as follows. Recall that d_{c_i} is the minimum number of cycles needed to be used for pressing each panel of type $i \in \hat{I}$. The d_{c_1} cycles for the first panel type are chosen from the first cycles of all press machines such that the work is distributed among the press machines equally. The d_{c_2} cycles for the second panel type are then chosen from the next available cycles of all the press machines so that the work is distributed equally, and so on. As a result of this panel-cycle assignment, the panels of the same type are finished in a group, which is preferable in real-world situations. Figure 11 depicts an example of this assignment.

Step 5: Output the number of finished goods of each panel type $i \in \hat{I}$; A_{pt}, C_{pt}, $\forall p \in \hat{P}$, $t \in \hat{T}$, the schedule of press machines and ovens, x_{iklpt}, $\forall i \in \hat{I}$, $k \in \hat{K}$, $l \in \hat{L}$, $p \in \hat{P}$, $t \in \hat{T}$, and the makespan.

The total number of finished goods of each panel type i can be computed from $ma_{i\bar{k}_i\bar{l}_i}d_{c_i}, \forall i \in \hat{I}$. The value of A_{pt}, C_{pt}, $\forall p \in \hat{P}$, $t \in \hat{T}$ can be obtained from matrices A and C in Phase 2, and these values can then be used for creating the schedule of press machines. The schedule of ovens can be interpreted from the list *Oven_Schedule_List* in Phase 2. The makespan of the overall processes is the maximum element in C. The output $x_{iklpt}, \forall i \in \hat{I}$, $k \in \hat{K}$, $l \in \hat{L}$, $p \in \hat{P}$, $t \in \hat{T}$, which is equal to 1, can be obtained from Phase 3. From all three phases, the computational complexity of the proposed 3P-PCB-PH algorithm is $O\left(P^2T^2 + IKL\right)$.

Note that PCB manufacturing companies prefer to finish each PCB type in a group, since it is easier to prepare material and sequence the next work. The proposed MILP in the previous section can find an optimal schedule for a pressing process with the minimum makespan, but cycles of the same panel type may not be scheduled consecutively. This is a limitation of the proposed MILP model, whereas the proposed 3P-PCB-PH algorithm can handle this preference. Therefore, the proposed 3P-PCB-PH algorithm is more practical for real PCB manufacturing industries.

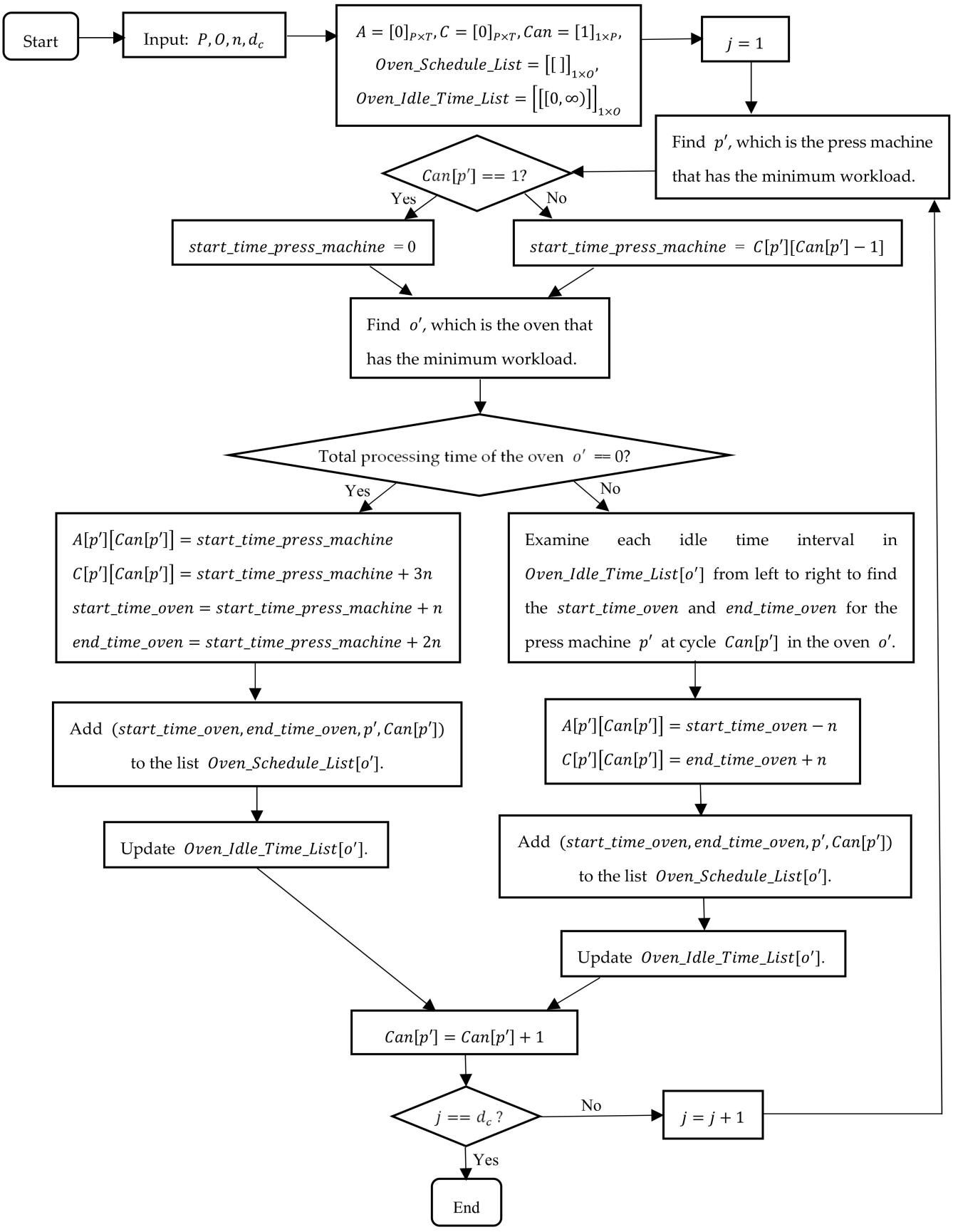

Figure 8. Flowchart of the 3P-PCB-PH algorithm for Phase 2.

5. Numerical Experiments

To demonstrate the proposed MILP model and 3P-PCB-PH algorithm, we used real-world data from a PCB company. The data and test problems are shown in Sections 5.1 and 5.2, respectively. The computational results from the proposed MILP model and heuristic algorithm are shown in Sections 5.3 and 5.4, respectively.

5.1. Data

The data acquired from an actual PCB company included seven panel types, six SST sizes, eight layouts, six press machines, each of which had 10 openings, and three ovens. We assumed that the processing time of each phase in the pressing process (the lay-up, pressing, and cool-down process phase) was 120 min, with a planning horizon of 3 days and a maximum number of available cycles of each press machine to be 12. This is because one cycle of a press machine takes 360 min (6 h). If a press machine works continuously, it can carry out up to 12 cycles of the pressing process in 3 days. We also considered a planning horizon of 2 and 1.5 days for the small problem, where the maximum number of available cycles of each press machine was eight and six cycles, respectively. The information of each type of panel, which consisted of warp (or length), fill (or width), inner gap, and outer gap, is shown in Table 1. The size of each SST is shown in Table 2.

Table 1. Sizes, inner gap, and outer gap of each panel type.

Panel	Warp (a)	Fill (b)	Inner Gap (g)	Outer Gap (G)
1	20.5	24	0.5	0.25
2	25.65	22.25	1	0.5
3	26	24	0.5	0.25
4	26.5	22.5	1	0.5
5	19	22.25	0.5	0.25
6	15	23.8	0.5	0.25
7	27.75	20.5	0.5	0.25

Table 2. Sizes of each SST.

Stainless-Steel	Warp (X)	Fill (Y)
1	50	44
2	50	53
3	50	56
4	50	58
5	43	25.5
6	43	27

The number of layouts was eight, as described in Figure 4 (in Section 2). The formulas for computing the number of panels (per book) based on the size of the SST and the layout are shown in Table 3. In the formulas, the values a, b, g, and G are the warp, fill, inner gap, and outer gap of panel type i, respectively. The values X and Y are the warp and fill of the SST size k, respectively. Note that the notation x is the greatest integer that is less than or equal to x.

5.2. Test Problems

According to Pan [25], the speed that mixed-integer linear programming problems can be solved at depends upon the number of binary variables, constraints, and continuous variables, where the most deciding factor is the number of binary variables. Therefore, the generated test problems are categorized to be 3 groups, i.e., small-, medium-, and large-sized test problems, depending on the number of binary variables.

Table 3. Formulas for computing the number of panels of type i per opening using SST size k and layout (a_{ikl}).

Layout (l)	a_{ikl}
1	$\left\lfloor \frac{X-2\left(G-\frac{g}{2}\right)}{a+g} \right\rfloor \times \left\lfloor \frac{Y-2\left(G-\frac{g}{2}\right)}{b+g} \right\rfloor$
2	$\left\lfloor \frac{X-2\left(G-\frac{g}{2}\right)}{b+g} \right\rfloor \times \left\lfloor \frac{Y-2\left(G-\frac{g}{2}\right)}{a+g} \right\rfloor$
3	$\left\lfloor \frac{X-2\left(G-\frac{g}{2}\right)}{a+g} \right\rfloor + \left(\left\lfloor \frac{X-2\left(G-\frac{g}{2}\right)}{b+g} \right\rfloor \times \left\lfloor \frac{Y-b-G-2\left(G-\frac{g}{2}\right)}{a+g} \right\rfloor \right)$
4	$\left\lfloor \frac{Y-2\left(G-\frac{g}{2}\right)}{a+g} \right\rfloor + \left(\left\lfloor \frac{Y-2\left(G-\frac{g}{2}\right)}{b+g} \right\rfloor \times \left\lfloor \frac{X-b-G-2\left(G-\frac{g}{2}\right)}{a+g} \right\rfloor \right)$
5	$\left\lfloor \frac{X-2\left(G-\frac{g}{2}\right)}{b+g} \right\rfloor + \left(\left\lfloor \frac{X-2\left(G-\frac{g}{2}\right)}{a+g} \right\rfloor \times \left\lfloor \frac{Y-a-G-2\left(G-\frac{g}{2}\right)}{b+g} \right\rfloor \right)$
6	$\left\lfloor \frac{Y-2\left(G-\frac{g}{2}\right)}{b+g} \right\rfloor + \left(\left\lfloor \frac{Y-2\left(G-\frac{g}{2}\right)}{a+g} \right\rfloor \times \left\lfloor \frac{X-a-G-2\left(G-\frac{g}{2}\right)}{b+g} \right\rfloor \right)$
7	$\left\lfloor \frac{X-2\left(G-\frac{g}{2}\right)}{a+g} \right\rfloor$
8	$\left\lfloor \frac{X-2\left(G-\frac{g}{2}\right)}{b+g} \right\rfloor$

5.2.1. Small-Sized Test Problems

The small-sized test problems were generated where the number of binary variables in each problem is less than 8500. The parameters in the small-sized test problems are as follows. The number of SST sizes (K) and the number of layouts (L) were six and eight respectively, as described in the previous subsection. The number of panel types (I) was three, which are the panel types 1–3 in Table 1. The number of press machines (P) and the number of ovens (O) were varied at three to four and two to three, respectively. The maximum number of available cycles of each press machine (T) was varied as six, eight, and 12 cycles, and the demand of each type of panel (d_i) was randomly generated. The details of the small-sized test problems are summarized in Table 4.

Table 4. The small-sized test problems for the proposed MILP model and 3P-PCB-PH algorithm.

No.	I	K	L	P	O	T	d_i, $i \in \{1,2,\ldots,I\}$
1	3	6	8	3	2	6	110, 150, 125
2	3	6	8	3	2	8	200, 220, 230
3	3	6	8	3	2	12	270, 250, 210
4	3	6	8	4	2	6	110, 150, 125
5	3	6	8	4	3	6	110, 150, 125

5.2.2. Medium-Sized Test Problems

The medium-sized test problems were generated where the number of binary variables in each problem is between 8500 to 30,000. The parameters in the medium-sized test problems are as follows. The number of panel types (I) was three to five, while the number of SST sizes (K), the number of layouts (L), the number of press machines (P), and the number of ovens (O) were six, eight, six, and three respectively, which are the real data from the previous subsection. The maximum number of available cycles (T) was varied at six, eight, or 12. The demands of each test problem were randomly generated. The details of the medium-sized test problems are shown in Table 5. In Problems 1–3, the number of types of panels was three, which included panel types 1–3 in Table 1. Problems 4–6 had panel types 1–4, and the other problems had panel types 1–5, as described in Table 1.

Table 5. The medium-sized test problems for the proposed MILP model and 3P-PCB-PH algorithm.

No.	*I*	*K*	*L*	*P*	*O*	*T*	d_i, $i \in \{1,2,\ldots,I\}$
1	3	6	8	6	3	6	300, 300, 300
2	3	6	8	6	3	8	450, 480, 500
3	3	6	8	6	3	12	720, 900, 600
4	4	6	8	6	3	6	200, 300, 400, 100
5	4	6	8	6	3	8	300, 400, 200, 500
6	4	6	8	6	3	12	500, 700, 700, 500
7	5	6	8	6	3	6	200, 250, 200, 250, 200
8	5	6	8	6	3	8	400, 300, 200, 250, 300

5.2.3. Large-Sized Test Problems

The large-sized test problems were generated where the number of binary variables in each problem is greater than 30,000. The parameters in the large-sized test problems are as follows. The number of panel types (I) was varied at five to seven. The number of SST sizes (K), the number of layouts (L), the number of press machines (P), and the number of ovens (O) were six, eight, six, and three respectively, which are the real data from the previous subsection. Furthermore, we also evaluated slightly larger-sized problems by increasing the number of press machines and ovens by one. The maximum number of available cycles (T) was 12 and the demand of each type of panel (d_i) was randomly generated. The details of the large-sized test problems are shown in Table 6. In Problems 1–3, the number of types of panels was five, which included panel types 1–5 in Table 1. Problems 4–6 had panel types 1–6, and the other problems had all seven panel types, as described in Table 1.

Table 6. The large-sized test problems for the proposed MILP model and 3P-PCB-PH algorithm.

No.	*I*	*K*	*L*	*P*	*O*	*T*	d_i, $i \in \{1,2,\ldots,I\}$
1	5	6	8	6	3	12	500, 500, 500, 500, 500
2	5	6	8	7	3	12	500, 500, 500, 500, 500
3	5	6	8	6	4	12	500, 500, 500, 500, 500
4	6	6	8	6	3	12	500, 360, 220, 180, 380, 720
5	6	6	8	7	3	12	500, 360, 220, 180, 380, 720
6	6	6	8	6	4	12	500, 360, 220, 180, 380, 720
7	7	6	8	6	3	12	300, 325, 290, 425, 450, 475, 200
8	7	6	8	7	3	12	300, 325, 290, 425, 450, 475, 200
9	7	6	8	6	4	12	300, 325, 290, 425, 450, 475, 200

5.3. Result of the Test Problems Using the Proposed MILP Model

In this section, all the test problems were solved using the proposed MILP model and the ILOG OPL CPLEX 12.6 software running on a personal computer with a core i7 2.20 GHz CPU and 8 GB RAM. The maximum running time was limited to 2 h.

5.3.1. Results of the Small-Sized Test Problems Using the Proposed MILP Model

The model size and computational results of each small-sized test problem using the proposed model are shown in Table 7. The model size consisted of the number of binary variables, continuous variables, and constraints. The results included the number of finished goods of each type of panel (outputs), CPU time, and the optimal makespan (C_{max}) of the overall process.

Table 7. Computational results of the small-sized test problems using the proposed MILP model.

No.	I	K	L	P	O	T	d_i	Model Size				Results	
								Binary	Continuous	Constraint	Outputs	CPU Time	C_{max}(min)
1	3	6	8	3	2	6	110, 150, 125	3060	128	3741	120, 160, 160	2.31 s	1440 [a]
2	3	6	8	3	2	8	200, 220, 230	4272	170	5373	200, 240, 240	2.48 s	2160 [a]
3	3	6	8	3	2	12	270, 250, 210	6984	254	9213	280, 280, 240	3.21 s	2520 [a]
4	3	6	8	4	2	6	110, 150, 125	4368	170	5563	120, 160, 160	8.15 s	1200 [a]
5	3	6	8	4	3	6	110, 150, 125	4824	218	6499	120, 160, 160	3.18 s	1080 [a]

[a] Optimal solution.

As shown in Table 7, all the small-sized test problems could be solved to an optimal solution within the 2 h time limit. The computational time of each problem is small. Note that Problems 1, 4, and 5 have the same demands. The results of Problem 4 indicate that if the number of press machines was increased by one from Problem 1, the pressing process of Problem 1 could be finished ahead of time for 240 min (i.e., the makespan was reduced from 1440 to 1200 min). However, the results of Problem 5 indicate that the pressing process of Problem 1 could be finished ahead of time for 360 min (i.e., the makespan was reduced from 1440 to 1080 min) if the number of press machines and ovens were increased by one from Problem 1. These show that the proposed MILP model can help in deciding which resources should be increased to reduce the production time.

5.3.2. Results of the Medium-Sized Test Problems Using the Proposed MILP Model

Table 8 shows the size and computational results of each medium-sized test problem using the proposed MILP model. The number of binary variables of each problem is between 8500 to 30,000. The results showed that all the medium-sized test problems could be solved to an optimal solution within the 2 h time limit. Note that the maximum computational time for solving the medium-sized test problems (9 min and 31 s in Problem 7 of the medium-sized test problems) increased significantly compared with the maximum computational time for solving the small-sized test problems, which is only around 8 s (in Problem 4 of the small-sized test problems).

5.3.3. Results of the Large-Sized Test Problems Using the Proposed MILP Model

The model size and computational results of each large-sized test problem using the proposed model are shown in Table 9. The results show that only Problems 1, 2, and 4 of the large-sized test problems could be solved to an optimal solution within the 2 h time limit, while the other problems could not, but we report the best feasible solution that could be found within the time limit. Note that the maximum computational time for solving the large-sized test problems to get an optimal solution (48 min and 14 s in Problem 4 of the large-sized test problems) increased significantly compared with the maximum computational time for solving the medium-sized test problems (9 min and 31 s in Problem 7 of the medium-sized test problems). In addition, an optimal solution could not be found for most large-sized test problems within the time limit of 2 h. This is common when solving large-sized mixed-integer linear programming problems. Since some practitioners can accept a promise solution within reasonable time instead of an optimal solution, this paper also presents a heuristic algorithm for solving the pressing process scheduling that could find a good solution within reasonable time, and the results of the proposed heuristic algorithm are presented in the next subsection.

Table 8. Computational results of the medium-sized test problems using the proposed MILP model.

No.	I	K	L	P	O	T	d_i	Model Size				Results	
								Binary	Continuous	Constraint	Outputs	CPU Time	C_{max}(min)
1	3	6	8	6	3	6	300, 300, 300	8532	326	12,339	320, 320, 320	32.81 s	1560 [a]
2	3	6	8	6	3	8	450, 480, 500	12,816	434	19,335	480, 480, 520	16.79 s	2520 [a]
3	3	6	8	6	3	12	720, 900, 600	23,544	650	37,647	720, 920, 600	1 min 28 s	3600 [a]
4	4	6	8	6	3	6	200, 300, 400, 100	10,260	326	14,068	200, 320, 400, 120	20.65 s	1800 [a]
5	4	6	8	6	3	8	300, 400, 200, 500	15,120	434	21,640	320, 400, 200, 520	3 min 59 s	2280 [a]
6	4	6	8	6	3	12	500, 700, 700, 500	27,000	650	41,104	520, 720, 720, 520	4 min 2 s	3960 [a]
7	5	6	8	6	3	6	200, 250, 200, 250, 200	11,988	326	15,797	200, 280, 200, 280, 200	9 min 31 s	1920 [a]
8	5	6	8	6	3	8	400, 300, 200, 250, 300	17,424	434	23,945	400, 320, 200, 280, 320	49.31 s	2520 [a]

[a] Optimal solution.

Table 9. Computational results of the large-sized test problems using the proposed MILP model.

No.	I	K	L	P	O	T	d_i	Model size				Results	
								Binary	Continuous	Constraint	Outputs	CPU Time	C_{max}(min)
1	5	6	8	6	3	12	500, 500, 500, 500, 500	30,456	650	44,561	520, 520, 520, 520, 520	44 min 38 s	4080 [a]
2	5	6	8	7	3	12	500, 500, 500, 500, 500	38,556	758	58,035	520, 520, 520, 520, 520	23 min 42 s	3600 [a]
3	5	6	8	6	4	12	500, 500, 500, 500, 500	34,848	794	53,417	520, 520, 520, 520, 520	2 h	4080 [b]
4	6	6	8	6	3	12	500, 360, 220, 180, 380, 720	33,912	650	48,018	520, 360, 240, 200, 400, 770	48 min 14 s	3360 [a]
5	6	6	8	7	3	12	500, 360, 220, 180, 380, 720	42,588	758	62,068	520, 360, 240, 200, 400, 770	2 h	3000 [b]
6	6	6	8	6	4	12	500, 360, 220, 180, 380, 720	38,304	794	56,874	520, 360, 240, 200, 400, 770	2 h	3360 [b]
7	7	6	8	6	3	12	300, 325, 290, 425, 450, 475, 200	37,368	650	51,475	320, 360, 320, 440, 480, 490, 200	2 h	3720 [b]
8	7	6	8	7	3	12	300, 325, 290, 425, 450, 475, 200	46,620	758	66,101	320, 360, 320, 440, 480, 490, 200	2 h	3360 [b]
9	7	6	8	6	4	12	300, 325, 290, 425, 450, 475, 200	41,760	794	60,331	320, 360, 320, 440, 480, 490, 200	2 h	3720 [b]

[a] Optimal solution. [b] The best-known solution from the proposed MILP model.

An example of an optimal solution from the proposed MILP model is described below. For the results of Problem 1 in Table 9, the number of outputs of panel types 1–5 that were obtained after the finishing pressing process was 520 each, which satisfied the demands. The variables x_{iklpt} and X_{pto}, which were equal to 1 in the optimal solution of Problem 1, are shown in Tables 10 and 11, respectively. The corresponding Gantt charts of the press machines and ovens are presented in Figures 9 and 10 respectively, where the same color represents the same panel type.

Table 10. List of non-zero x_{iklpt} values in the solution of Problem 1 of the large-sized problems using the proposed MILP model.

Press Machine	Non-Zero x_{iklpt}
1	$x_{13111}, x_{51212}, x_{24213}, x_{51214}, x_{33215}, x_{51216}, x_{51217}, x_{24218}, x_{43219}, x_{3221,10}, x_{4321,11}$
2	$x_{14621}, x_{53622}, x_{53223}, x_{54524}, x_{32225}, x_{12626}, x_{43227}, x_{44228}, x_{32229}, x_{1222,10}, x_{3222,11}$
3	$x_{23431}, x_{51532}, x_{44233}, x_{43234}, x_{43235}, x_{43236}, x_{12437}, x_{34238}, x_{12239}, x_{2343,10}, x_{4423,11}$
4	$x_{14141}, x_{43242}, x_{32243}, x_{24244}, x_{13445}, x_{32246}, x_{32247}, x_{32248}, x_{24649}, x_{5414,10}, x_{5134,11}$
5	$x_{12151}, x_{14352}, x_{54553}, x_{23454}, x_{32255}, x_{23656}, x_{42257}, x_{32258}, x_{43259}, x_{2365,10}$
6	$x_{51361}, x_{13362}, x_{24663}, x_{42264}, x_{24665}, x_{23466}, x_{24467}, x_{54468}, x_{11269}, x_{1126,10}, x_{3426,11}$

Table 11. List of non-zero X_{pto} values in the solution of Problem 1 of the large-sized problems using the proposed MILP model.

Press Machine	Non-Zero X_{pto}
1	$X_{111}, X_{122}, X_{131}, X_{142}, X_{151}, X_{161}, X_{171}, X_{181}, X_{191}, X_{1,10,3}, X_{1,11,3}$
2	$X_{212}, X_{223}, X_{233}, X_{241}, X_{253}, X_{263}, X_{272}, X_{282}, X_{292}, X_{2,10,1}, X_{2,11,1}$
3	$X_{311}, X_{322}, X_{331}, X_{341}, X_{353}, X_{363}, X_{373}, X_{383}, X_{393}, X_{3,10,2}, X_{3,11,2}$
4	$X_{412}, X_{421}, X_{433}, X_{442}, X_{451}, X_{461}, X_{472}, X_{482}, X_{491}, X_{4,10,1}, X_{4,11,3}$
5	$X_{513}, X_{522}, X_{531}, X_{543}, X_{553}, X_{563}, X_{573}, X_{581}, X_{593}, X_{5,10,2}$
6	$X_{613}, X_{623}, X_{632}, X_{643}, X_{652}, X_{662}, X_{673}, X_{681}, X_{692}, X_{6,10,3}, X_{6,11,1}$

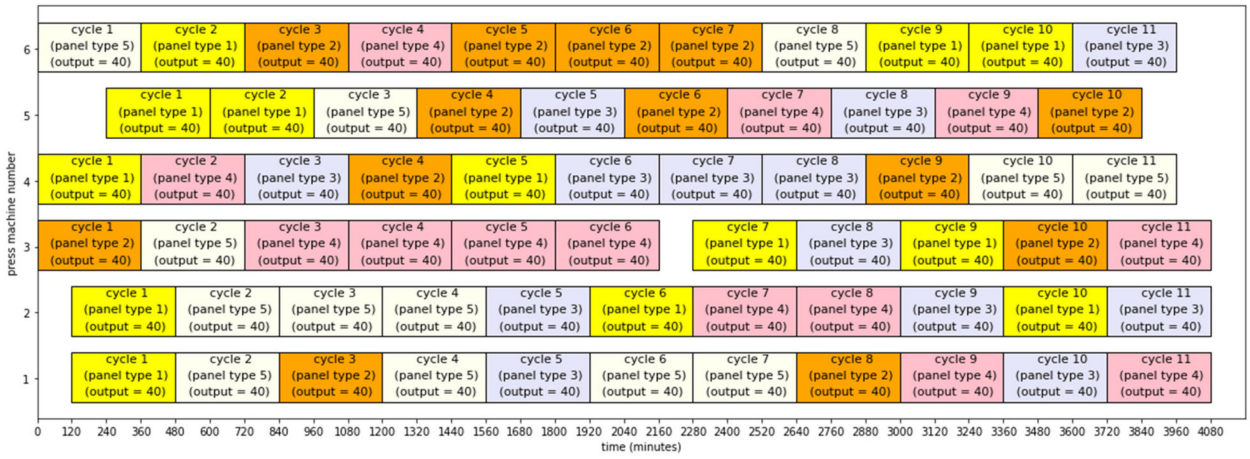

Figure 9. Gantt chart of the press machines for Problem 1 of the large-sized problems using the proposed MILP model.

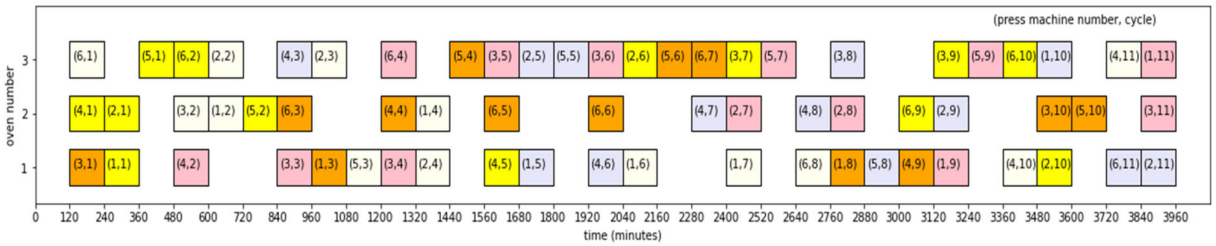

Figure 10. Gantt chart of the ovens for Problem 1 of the large-sized problems using the proposed MILP model.

From Table 10, the list of non-zero x_{iklpt} were sorted by cycle numbers (index t) in ascending order, while the list of non-zero X_{pto} values (Table 11) were also sorted in a similar manner.

Figure 9 shows the starting time and completion time of each cycle of each press machine. One cycle of the press machine takes 360 min, i.e., 120 min for each lay-up, pressing, and cool-down process phase. The time for the pressing process phase for each cycle of each press machine is depicted in Figure 10. For example, press machine 3 at cycle 1 had to lay up at 0–120 min (Figure 9), move into the oven 1 at 120–240 min (Figure 10), and cool-down at 240–360 min (Figure 9). The minimum makespan of the overall process was 4080 min (Figure 9).

5.4. Result of the Test Problems Using the Proposed 3P-PCB-PH Algorithm

In this section, all the test problems were solved using the proposed 3P-PCB-PH algorithm implemented in Python version 3.7.3 running under the same hardware environment as in the previous subsection. Each problem was run 10 times to capture the variation in the computational time. The results of each test problem when using the proposed heuristic algorithm were compared with the results from the proposed MILP model.

5.4.1. Results of the Small-Sized Test Problems Using the Proposed 3P-PCB-PH Algorithm

The results of the small-sized test problems from the heuristic algorithm and the proposed MILP model are compared in Table 12. The results included the number of finished goods of each type of panel (outputs), the average CPU time over 10 runs (Avg CPU time), and the makespan (C_{max}) of the overall process. The last column of Table 12 reports the percentage gap ($\%gap$) between the makespan from the proposed heuristic algorithm and the optimal makespan or best-known makespan from the proposed MILP model.

Table 12. Computational results of the small-sized test problems using the proposed 3P-PCB-PH algorithm.

No.	I	K	L	P	O	T	d_i	Results Using Proposed MILP Model			Results Using Proposed 3P-PCB-PH Algorithm			$\%gap$
								Outputs	CPU Time	C_{max} (min)	Outputs	Avg CPU Time (SD)	C_{max} (min)	
1	3	6	8	3	2	6	110, 150, 125	120, 160, 160	2.31 s	1440 [a]	120, 160, 160	0.00349 s (0.00085 s)	1440 [a]	0%
2	3	6	8	3	2	8	200, 220, 230	200, 240, 240	2.48 s	2160 [a]	200, 240, 240	0.00488 s (0.00246 s)	2160 [a]	0%
3	3	6	8	3	2	12	270, 250, 210	280, 280, 240	3.21 s	2520 [a]	280, 280, 240	0.00658 s (0.00346 s)	2520 [a]	0%
4	3	6	8	4	2	6	110, 150, 125	120, 160, 160	8.15 s	1200 [a]	120, 160, 160	0.00598 s (0.00342 s)	1200 [a]	0%
5	3	6	8	4	3	6	110, 150, 125	120, 160, 160	3.18 s	1080 [a]	120, 160, 160	0.00509 s (0.00371 s)	1080 [a]	0%

[a] Optimal solution.

As shown in Table 12, the proposed 3P-PCB-PH algorithm could solve all the small-sized test problems with an average and standard deviation (SD) computational time of less than 1 s respectively, for solving each problem. The makespans from the proposed heuristic algorithm were the same as the optimal makespans from the proposed MILP model ($\%gap = 0\%$), but the proposed heuristic algorithm used less computational times

than the proposed MILP model. This shows that the proposed 3P-PCB-PH algorithm is very efficient and effective.

5.4.2. Results of the Medium-Sized Test Problems Using the Proposed 3P-PCB-PH Algorithm

Table 13 shows the computational results of each test problem when using the proposed heuristic algorithm compared with the results from the proposed MILP model. All the medium-sized test problems could still be solved to an optimal solution ($\%gap = 0\%$) by the proposed heuristic algorithm using only a very small average and SD computational time of less than 1 s each. This shows the efficiency and effectiveness of the proposed 3P-PCB-PH algorithm.

Table 13. Computational results of the medium-sized test problems using the proposed 3P-PCB-PH algorithm.

No.	I	K	L	P	O	T	d_i	Results Using Proposed MILP Model			Results Using Proposed 3P-PCB-PH Algorithm			%*gap*
								Outputs	CPU Time	C_{max} (min)	Outputs	Avg CPU Time (SD)	C_{max} (min)	
1	3	6	8	6	3	6	300, 300, 300	320, 320, 320	32.81 s	1560 [a]	320, 320, 320	0.00469 s (0.00141 s)	1560 [a]	0%
2	3	6	8	6	3	8	450, 480, 500	480, 480, 520	16.79 s	2520 [a]	480, 480, 520	0.00519 s (0.00248 s)	2520 [a]	0%
3	3	6	8	6	3	12	720, 900, 600	720, 920, 600	1 min 28 s	3600 [a]	720, 920, 600	0.00658 s (0.00245 s)	3600 [a]	0%
4	4	6	8	6	3	6	200, 300, 400, 100	200, 320, 400, 120	20.65 s	1800 [a]	200, 320, 400, 120	0.00599 s (0.00266 s)	1800 [a]	0%
5	4	6	8	6	3	8	300, 400, 200, 500	320, 400, 200, 520	3 min 59 s	2280 [a]	320, 400, 200, 520	0.00768 s (0.00342 s)	2280 [a]	0%
6	4	6	8	6	3	12	500, 700, 700, 500	520, 720, 720, 520	4 min 2 s	3960 [a]	520, 720, 720, 520	0.00927 s (0.00509 s)	3960 [a]	0%
7	5	6	8	6	3	6	200, 250, 200, 250, 200	200, 280, 200, 280, 200	9 min 31 s	1920 [a]	200, 280, 200, 280, 200	0.00768 s (0.00282 s)	1920 [a]	0%
8	5	6	8	6	3	8	400, 300, 200, 250, 300	400, 320, 200, 280, 320	49.31 s	2520 [a]	400, 320, 200, 280, 320	0.00909 s (0.00331 s)	2520 [a]	0%

[a] Optimal solution.

5.4.3. Results of the Large-Sized Test Problems Using the Proposed 3P-PCB-PH Algorithm

The results of the large-sized test problems from the proposed heuristic algorithm and the proposed MILP model are compared in Table 14. Each problem was solved by the proposed heuristic algorithm using an average and SD computational time of less than 1 s each. For Problems 1, 2, and 4, the makespans from the proposed heuristic algorithm are the same as the optimal makespans from the proposed MILP model ($\%gap = 0\%$), but the

proposed heuristic algorithm used much less computational time than the proposed MILP model. Furthermore, the proposed heuristic algorithm could find a near-optimal schedule with the same makespan as the best-known solution from the proposed MILP model for the other large-sized test problems using very small computational times. Note that the computational time of the proposed heuristic algorithm slightly increases when the size of problem is increased from small size to large size, which is different from the computational time of the proposed MILP model. These results show that the proposed 3P-PCB-PH algorithm is very efficient and effective for solving the pressing process scheduling.

Table 14. Computational results of the large-sized test problems using the proposed 3P-PCB-PH algorithm.

No.	I	K	L	P	O	T	d_i	Results Using Proposed MILP Model			Results Using Proposed 3P-PCB-PH Algorithm			%*gap*
								Outputs	CPU Time	C_{max} (min)	Outputs	Avg CPU Time (SD)	C_{max} (min)	
1	5	6	8	6	3	12	500, 500, 500, 500, 500	520, 520, 520, 520, 520	44 min 38 s	4080 [a]	520, 520, 520, 520, 520	0.00928 s (0.00346 s)	4080 [a]	0%
2	5	6	8	7	3	12	500, 500, 500, 500, 500	520, 520, 520, 520, 520	23 min 42 s	3600 [a]	520, 520, 520, 520, 520	0.00918 s (0.00297 s)	3600 [a]	0%
3	5	6	8	6	4	12	500, 500, 500, 500, 500	520, 520, 520, 520, 520	2 h	4080 [b]	520, 520, 520, 520, 520	0.00987 s (0.00447 s)	4080	0% [c]
4	6	6	8	6	3	12	500, 360, 220, 180, 380, 720	520, 360, 240, 200, 400, 770	48 min 14 s	3360 [a]	520, 360, 240, 200, 400, 770	0.00997 s (0.00326 s)	3360 [a]	0%
5	6	6	8	7	3	12	500, 360, 220, 180, 380, 720	520, 360, 240, 200, 400, 770	2 h	3000 [b]	520, 360, 240, 200, 400, 770	0.00908 s (0.00291 s)	3000	0% [c]
6	6	6	8	6	4	12	500, 360, 220, 180, 380, 720	520, 360, 240, 200, 400, 770	2 h	3360 [b]	520, 360, 240, 200, 400, 770	0.00993 s (0.00575 s)	3360	0% [c]
7	7	6	8	6	3	12	300, 325, 290, 425, 450, 475, 200	320, 360, 320, 440, 480, 490, 200	2 h	3720 [b]	320, 360, 320, 440, 480, 490, 200	0.01015 s (0.00319 s)	3720	0% [c]
8	7	6	8	7	3	12	300, 325, 290, 425, 450, 475, 200	320, 360, 320, 440, 480, 490, 200	2 h	3360 [b]	320, 360, 320, 440, 480, 490, 200	0.01250 s (0.00504 s)	3360	0% [c]
9	7	6	8	6	4	12	300, 325, 290, 425, 450, 475, 200	320, 360, 320, 440, 480, 490, 200	2 h	3720 [b]	320, 360, 320, 440, 480, 490, 200	0.01057 s (0.00566 s)	3720	0% [c]

[a] Optimal solution. [b] The best-known solution from the proposed MILP model. [c] The %*gap* between the solution from the heuristic algorithm and the best-known solution from the MILP model.

In addition, the results from the proposed heuristic algorithm can give valuable information. For example, from Problems 1–3, all parameters in the problems are the same except for the number of press machines and ovens. The results of Problem 2 indicate that if the number of press machines was increased by one from Problem 1, the pressing process of Problem 1 could be finished ahead of time for 360 min (i.e., the makespan was reduced from 4080 to 3600 min). However, the results of Problem 3 indicate that increasing the number of ovens by one from Problem 1 cannot reduce the makespan. The manager of the company should increase the number of press machines rather than the number of ovens if he/she wants to reduce the makespan of the pressing process of Problem 1. This is

the same in Problems 4–6, and Problems 7–9. Note that if the number of press machines is increased, the number of cycles that is needed for the demands to be satisfied can be distributed to more press machines and, as a consequence, all demands can be finished faster. These show that the proposed 3P-PCB-PH algorithm can also help in deciding which resources should be increased to reduce the production time.

An example of a solution from the proposed heuristic algorithm is described below, where the results of Problem 1 are shown in Tables 15 and 16 for the variables x_{iklpt} and X_{pto}, which are equal to 1, and in Figures 11 and 12 for the Gantt charts of the press machines and ovens. These Gantt charts were different from the Gantt charts from the MILP model (Figures 9 and 10), and this shows that Problem 1 of the large-sized problems has an alternative optimal schedule. Note that, in Figure 11, each type of panel is finished as a group, which is preferable in the real manufacturing industry. The makespan of the overall process was 4080 min and the number of outputs of each panel type was 520, which satisfied the demand.

Table 15. List of non-zero x_{iklpt} values in the solution of Problem 1 of the large-sized problems using the proposed 3P-PCB-PH algorithm.

Press Machine	Non-Zero x_{iklpt}
1	$x_{11211}, x_{11212}, x_{11213}, x_{23214}, x_{23215}, x_{32216}, x_{32217}, x_{43218}, x_{43219}, x_{5121,10}, x_{5121,11}$
2	$x_{11221}, x_{11222}, x_{23223}, x_{23224}, x_{23225}, x_{32226}, x_{32227}, x_{43228}, x_{43229}, x_{5122,10}, x_{5122,11}$
3	$x_{11231}, x_{11232}, x_{23233}, x_{23234}, x_{32235}, x_{32236}, x_{32237}, x_{43238}, x_{43239}, x_{5123,10}, x_{5123,11}$
4	$x_{11241}, x_{11242}, x_{23243}, x_{23244}, x_{32245}, x_{32246}, x_{43247}, x_{43248}, x_{43249}, x_{5124,10}, x_{5124,11}$
5	$x_{11251}, x_{11252}, x_{23253}, x_{23254}, x_{32255}, x_{32256}, x_{43257}, x_{43258}, x_{51259}, x_{5125,10}, x_{5125,11}$
6	$x_{11261}, x_{11262}, x_{23263}, x_{23264}, x_{32265}, x_{32266}, x_{43267}, x_{43268}, x_{51269}, x_{5126,10}$

Table 16. List of non-zero X_{pto} values in the solution of Problem 1 of the large-sized problems using the proposed 3P-PCB-PH algorithm.

Press Machine	Non-Zero X_{pto}
1	$X_{111}, X_{121}, X_{131}, X_{141}, X_{151}, X_{161}, X_{171}, X_{181}, X_{191}, X_{1,10,1}, X_{1,11,1}$
2	$X_{212}, X_{222}, X_{232}, X_{242}, X_{252}, X_{262}, X_{272}, X_{282}, X_{292}, X_{2,10,2}, X_{2,11,2}$
3	$X_{313}, X_{323}, X_{333}, X_{343}, X_{353}, X_{363}, X_{373}, X_{383}, X_{393}, X_{3,10,3}, X_{3,11,3}$
4	$X_{411}, X_{421}, X_{431}, X_{441}, X_{451}, X_{461}, X_{471}, X_{481}, X_{491}, X_{4,10,1}, X_{4,11,1}$
5	$X_{512}, X_{522}, X_{532}, X_{542}, X_{552}, X_{562}, X_{572}, X_{582}, X_{592}, X_{5,10,2}, X_{5,11,2}$
6	$X_{613}, X_{623}, X_{633}, X_{643}, X_{653}, X_{663}, X_{673}, X_{683}, X_{693}, X_{6,10,3}$

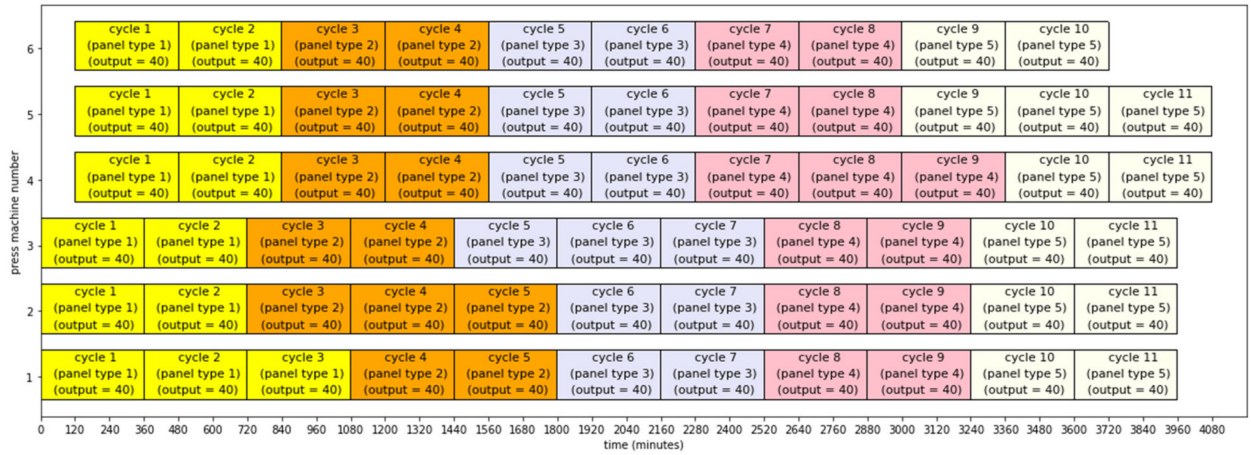

Figure 11. Gantt chart of the press machines for Problem 1 of the large-sized problems using the proposed 3P-PCB-PH algorithm.

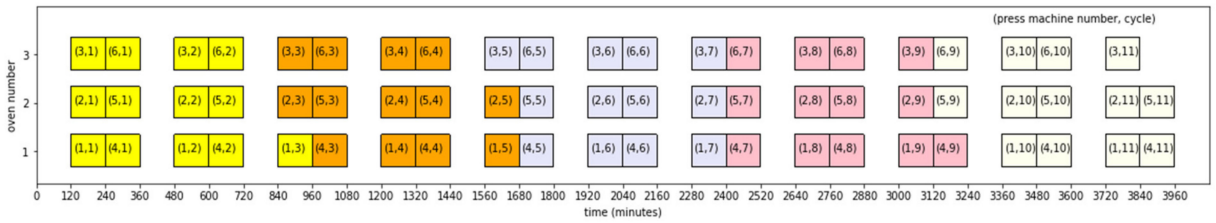

Figure 12. Gantt chart of the ovens for Problem 1 of the large-sized problems using the proposed 3P-PCB-PH algorithm.

6. Discussion

This paper presents a MILP model and a 3P-PCB-PH algorithm for solving the pressing process scheduling. From the numerical experiments, the proposed MILP model is suitable for the small-sized and medium-sized problems, where the number of binary variables is less than 30,000. The proposed MILP model tends to cause long computational times for solving the large-sized problems, where the number of binary variables is greater than 30,000. Furthermore, the running time was significantly increased as the size of the problem grows because there are a lot of feasible solutions to be verified for optimality due to many decision variables. However, the proposed MILP model has the benefit that it gives an optimal solution if one exists. On the other hand, the proposed 3P-PCB-PH algorithm is suitable for all sizes of problems. It could find an optimal solution for all problems that the proposed MILP model could find. It also can find the same best makespans as the proposed MILP model for all problems that the proposed MILP model could not find an optimal solution. The computational times of the proposed heuristic algorithm seem to be very fast and are not hugely increased when the size of the problem is increased from small size to large size. A benefit of the proposed heuristic algorithm is the saving in time to find a good solution since it used smaller computational times compared with the computational times of the proposed MILP model.

The proposed MILP model can also be easily extended to be more practical in the real-life application. For example, in the proposed model, the objective is to minimize the makespan of the overall process of the pressing process scheduling, where the demands must be satisfied. However, the surplus output of each panel type may be too large. If we also want to enforce that the surplus output of each panel type should not be too excessive with the main objective makespan, we can add the term $\varepsilon \sum_{i=1}^{I} \left[\sum_{k=1}^{K} \sum_{l=1}^{L} \sum_{p=1}^{P} \sum_{t=1}^{T} x_{iklpt}(ma_{ikl}) - d_i \right]$ to the objective function. The constant ε should be very small so that it has no effect on minimizing the main objective makespan.

7. Conclusions

This paper presented a new application of a mixed-integer linear programming to the scheduling of the pressing process in multi-layer PCB manufacturing. In the process, the panels are inserted into a press machine and then sent into an oven so that the panels are pressed and heated in the oven. The objective of the scheduling problem was to minimize the makespan of the overall pressing process. This objective can often imply increasing the utilization of available resources.

The goal of this study was to present two methods for solving the pressing process scheduling, i.e., a MILP model which is an exact method and a 3P-PCB-PH algorithm which is an approximation method. The first method illustrates a possible application of the integer linear programming that can handle a complicated problem from the real-world industry. The real data from a PCB company was used to generate the test problems. The computational results indicated that the proposed MILP model was suitable for small- and medium-size problems. The proposed MILP model could find an optimal solution for some large-sized problems and a good feasible solution for the other large-sized problems within the time limit. The MILP model has an advantage that it can guarantee to find an optimal

solution if the problem can be solved optimally within the time limit. On the other hand, the proposed 3P-PCB-PH algorithm could find the optimal solutions and near optimal solutions within very small computational time. It is more suitable than the proposed MILP model when the size of the problem is large. Furthermore, the schedule from the proposed heuristic is preferable in real manufacturing than the schedule from the proposed MILP model since each type of panel is finished in a single group. Both the proposed MILP model and 3P-PCB-PH algorithm could be options to provide an optimal schedule for the pressing process in any PCB industries or could be adapted to other industrial applications with similar aspects of scheduling.

Some additional constraints can be introduced into the pressing process for further development. For example, the cycle time depends on each type of panel, one cycle of a press machine can press more than one type of panel, and some types of panels have a higher priority or different due date. Adding these factors to the problem would also be a very challenging task for the future research, but also increase the complexity of the problem.

The limitations of this paper are that the problem is assumed to have the same size of press machines and the same size of ovens. In reality, however, a PCB company may have several sizes of press machines or ovens.

Author Contributions: Conceptualization, T.L., B.I., and C.J.; methodology, T.L., B.I., and C.J.; software, T.L. and C.J.; validation, T.L., B.I., and C.J.; formal analysis, T.L., B.I., and C.J.; investigation, T.L., B.I., and C.J.; resources, C.J.; data curation, C.J.; writing—original draft preparation, T.L.; writing—review and editing, B.I. and C.J.; visualization, T.L.; supervision, B.I. and C.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This work was supported in part by the Development and Promotion of Science and Technology Talented Project (DPST), Thailand, in part by Center of Excellence in Logistics and Supply Chain Systems Engineering and Technology (LogEn Tech), Sirindhorn International Institute of Technology, Thammasat University, Thailand, and in part by Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University, Thailand.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Khandpur, R.S. *Printed Circuit Boards: Design, Fabrication, Assembly and Testing*; McGraw-Hill: New York, NY, USA, 2006.
2. Gilmore, P.C.; Gomory, R.E. Multistage cutting stock problems of two and more dimensions. *Oper. Res.* **1965**, *13*, 94–120. [CrossRef]
3. Macedo, R.; Alves, C.; de Carvalho, J.M.V. Arc-flow model for the two-dimensional guillotine cutting stock problem. *Comput. Oper. Res.* **2010**, *37*, 991–1001. [CrossRef]
4. Mrad, M.; Meftahi, I.; Haouari, M. A branch-and-price algorithm for the two-stage guillotine cutting stock problem. *J. Oper. Res. Soc.* **2013**, *64*, 629–637. [CrossRef]
5. Alvarez-Valdes, R.; Parajon, A.; Tamarit, J.M. A computational study of LP-based heuristic algorithms for two-dimensional guillotine cutting stock problems. *OR Spectr.* **2002**, *24*, 179–192. [CrossRef]
6. Furini, F.; Malaguti, E.; Duran, R.M.; Persiani, A.; Toth, P. A column generation heuristic for the two-dimensional two-staged guillotine cutting stock problem with multiple stock size. *Eur. J. Oper. Res.* **2012**, *218*, 251–260. [CrossRef]
7. Tieng, K.; Sumetthapiwat, S.; Dumrongsiri, A.; Jeenanunta, C. Heuristics for two-dimensional rectangular guillotine cutting stock. *Thail. Stat.* **2016**, *14*, 147–164.
8. Sumetthapiwat, S.; Intiyot, B.; Jeenanunta, C. A column generation on two-dimensional cutting stock problem with fixed-size usable leftover and multiple stock sizes. *Int. J. Logist. Manag.* **2020**, *35*, 273–288. [CrossRef]
9. Onwubolu, G.C.; Clerc, M. Optimal path for automated drilling operations by a new heuristic approach using particle swarm optimization. *Int. J. Prod. Res.* **2004**, *42*, 473–491. [CrossRef]
10. Saealal, M.S.; Abidin, A.F.; Adam, A.; Mukred, J.; Khalil, K.; Yusof, Z.M.; Ibrahim, Z.; Nordin, N. An ant colony system for routing in PCB holes drilling process. *IJIMIP* **2013**, *4*, 50–56.
11. Lim, W.C.E.; Kanagaraj, G.; Ponnambalam, S.G. PCB drill path optimization by combinatorial cuckoo search algorithm. *Sci. World J.* **2014**, 264518. [CrossRef]

12. Kanagaraj, G.; Ponnambalam, S.G.; Lim, W.C.E. Application of a hybridized cuckoo search-genetic algorithm to path optimization for PCB holes drilling process. In Proceedings of the IEEE International Conference on Automation Science and Engineering, Taipei, Taiwan, 18–22 August 2014; pp. 18–22.
13. Ji, P.; Sze, M.T.; Lee, W.B. A genetic algorithm of determining cycle time for printed circuit board assembly lines. *Eur. J. Oper. Res.* **2001**, *128*, 175–184. [CrossRef]
14. Kodek, D.M.; Krisper, M. Optimal algorithm for minimizing production cycle time of a printed circuit board assembly line. *Int. J. Prod. Res.* **2004**, *42*, 5031–5048. [CrossRef]
15. Emet, S.; Knuutila, T.; Alhoniemi, E.; Maier, M.; Johnsson, M.; Nevalainen, O.S. Workload balancing in printed circuit board assembly. *Int. J. Adv. Manuf. Technol.* **2010**, *50*, 1175–1182. [CrossRef]
16. He, T.; Li, D.; Yoon, S.W. A heuristic algorithm to balance workloads of high-speed SMT machines in a PCB assembly line. *Procedia Manuf.* **2017**, *11*, 1790–1797. [CrossRef]
17. Damodaran, P.; Srihari, K.; Lam, S.S. Scheduling a capacitated batch-processing machine to minimize makespan. *Robot. Comput. Integr. Manuf.* **2007**, *23*, 208–216. [CrossRef]
18. Damodaran, P.; Diyadawagamage, D.A.; Ghrayeb, O.; Velez-Gallego, M.C. A particle swarm optimization algorithm for minimizing makespan of nonidentical parallel batch processing machines. *Int. J. Adv. Manuf. Technol.* **2012**, *58*, 1131–1140. [CrossRef]
19. Noroozi, A.; Mokhtari, H. Scheduling of printed circuit board (PCB) assembly systems with heterogeneous processors using simulation-based intelligent optimization methods. *Neural. Comput. Appl.* **2015**, *26*, 857–873. [CrossRef]
20. Hulett, M.; Damodaran, P.; Amouie, M. Scheduling non-identical parallel batch processing machines to minimize total weighted tardiness using particle swarm optimization. *Comput. Ind. Eng.* **2017**, *113*, 425–436. [CrossRef]
21. Ozguven, C.; Ozbakir, L.; Yavuz, Y. Mathematical models for job-shop scheduling problems with routing and process plan flexibility. *Appl. Math. Model.* **2010**, *34*, 1539–1548. [CrossRef]
22. Zhang, G.; Gao, L.; Shi, Y. An effective genetic algorithm for the flexible job-shop scheduling problem. *Expert Syst. Appl.* **2011**, *38*, 3563–3573. [CrossRef]
23. Li, X.; Gao, L. An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem. *Int. J. Prod. Econ.* **2016**, *174*, 93–110. [CrossRef]
24. Luan, F.; Cai, Z.; Wu, S.; Liu, S.Q.; He, Y. Optimizing the low-carbon flexible job shop scheduling problem with discrete whale optimization algorithm. *Mathematics* **2019**, *7*, 688. [CrossRef]
25. Pan, C.H. A study of integer programming formulations for scheduling problems. *Int. J. Syst. Sci.* **1997**, *28*, 33–41. [CrossRef]