



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

Teeradech Laisupannawong¹, Boonyarit Intiyot^{1,*} and Chawalit Jeenanunta²

- 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 pressing process is a part of the fabrication process of multi-layer printed circuit board (PCB) manufacturing. This paper presents the application of a new mixed-integer linear programming model to the short-term scheduling of the pressing process. The objective was to minimize the makespan. The proposed model is an improvement from our previous model in the literature. The size complexity of the proposed model is better than that of the previous model, whereby the number of variables, constraints, and the dimensionality of variables in the previous model are reduced. To compare their performance, problems from literature and additional generated test problems were solved. The proposed model was shown to outperform the previous model in terms of computational complexity. It can verify a new optimal solution for some problems. For the problems that could not be solved optimally, the proposed model could find the incumbent solution using much less computational time than the previous model, and the makespan of the incumbent solution from the proposed model was better than or equal to that of the previous model. The proposed model was better than or equal to that of the previous model. The proposed model was better than or equal to that of the previous model.

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

1. Introduction

Printed circuit boards (PCBs) are used as a major component in most electronic devices [1]. Nowadays, most PCB manufacturers are facing the challenge of improving production efficiency to cope with the strong competition [2]. In every aspect of production, the productivity and cost are crucial matters. Reducing the production cost whilst maximizing the efficiency of the production machines are the prime objectives of a manufacturing enterprise.

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. This paper only considers multi-layer PCBs. Multi-layer PCB manufacturing consists of the design, fabrication, assembly, and testing [3]. The design step is creating a circuit schematic by PCB designers. The fabrication step is creating the PCB or bare board. There are many processes in multi-layer PCB fabrication, such as cutting, pressing, and drilling processes. The assembly step is the placing of electronic components at the predefined positions on the bare board. Lastly, the testing step identifies early fallouts before the PCBs are sent for use in the field.

This paper focuses on the pressing process in multi-layer PCB fabrication, which involves various PCB components and costly machines. Scheduling an efficient pressing process requires backgrounds in assignment and sequencing to reduce the production time and to increase machine utilization. The mathematical problems in the literature that have



Citation: Laisupannawong, T.; Intiyot, B.; Jeenanunta, C. Improved Mixed-Integer Linear Programming Model for Short-Term Scheduling of the Pressing Process in Multi-Layer Printed Circuit Board Manufacturing. *Mathematics* **2021**, *9*, 2653. https:// doi.org/10.3390/math9212653

Academic Editors: Jorge Orestes Cerdeira and Cristina Requejo

Received: 17 September 2021 Accepted: 18 October 2021 Published: 20 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). similar backgrounds to the pressing process (assignment and sequencing) are the flexible job shop scheduling problem (FJSP) [4–7] and the hybrid flow shop scheduling problem (HFSP) [8,9]. However, literature about pressing process scheduling is quite limited. To the best of the authors' knowledge, there is no prior work that directly investigated pressing process scheduling, except the authors' prior publication [10], where a mixed integer linear programming (MILP) model and a heuristic algorithm were used for solving the pressing process scheduling. The objective of pressing process scheduling is to minimize makespan. The test problems generated from the data of an actual PCB company were solved using both methods to show the performance.

Although approximate methods, such as a heuristic algorithm, can solve an optimization problem within a relatively short time, it can be difficult to guarantee the quality of the solution to the problem. The first key step to study an optimization problem is to mathematically formulate it. A mathematical programming model is the first natural way to approach a scheduling problem [8,11]. It can explicitly describe the objective, constraints, and all the other characteristics of the scheduling problem. Moreover, it can serve as the key to develop a heuristic algorithm for solving the problem [12]. The MILP is a type of mathematical programming model that allows an exact method to solve a scheduling problem and gives an optimal solution if one exists. Therefore, it is important to improve the MILP model to be more effective at solving scheduling problems. The efficiency of a MILP depends on many factors, such as the number of binary variables (NBV), the number of continuous variables (NCV), the number of constraints (NC), the dimensionality of decision variables, and the constraints' tightness [9]. The most deciding factor that affects the performance of a MILP is the NBV, and the next influential factor is the NC [11].

To compare the performance between any two MILP models, there are two widely used performance measures of size and computational complexities [8,9,13]. The size complexity counts the NBV, NCV, and NC generated by the MILP model. If every size complexity factor of a MILP model is smaller than another model, it certainly is more superior. The computational complexity is determined by the computational time required to solve the problem. A MILP model outperforms another model in terms of computational complexity if it can solve the problem using less computational time. Table 1 shows the literature that aims to improve and compare the performance of MILP models of various types of scheduling using these performance measures.

Ref.	Problem	Objective	Highlight
[14]	Open shop, job shop, flow shop, and permutation flow shop scheduling problems with limited waiting time constraints (LWT-OSP, LWT-JSP, LWT-FSP, and LWT-PFSP)	Minimize makespan	For each problem (LWT-OSP, LWT-JSP, LWT-FSP, and LWT-PFSP), the authors proposed two MILP models and compared the size complexity between them to find the best model.
[15]	Open shop scheduling problem (OSP)	Minimize tardiness	The authors proposed four MILP models for the OSP and compared their performance in terms of both size and computational complexities to find the best model for the OSP.
[16]	FJSP with sequence dependent setup time (SDST-FJSP)	Minimize tardiness	The author proposed a MILP model for the SDST-FJSP and compared the performance of the proposed model with the existing MILP model [17] in terms of both size and computational complexities. The results showed that the proposed model outperformed the existing model [17] in terms of both complexities.
[7]	FJSP	Minimize makespan	The authors proposed two MILP models for the FJSP and compared the performance of these two proposed models with the existing models [5,18,19] in terms of both size and computational complexities. The results showed that the two proposed models outperformed the existing models [5,18,19] in terms of both complexities.

Table 1. Articles related to the comparison and improvement of MILP models of scheduling problems.

Table 1. Cont.

Ref.	Problem	Objective	Highlight
[8]	HFSP	Minimize makespan	The authors proposed four MILP models for the HFSP and compared the performance of them in terms of both size and computational complexities to find the best model for the HFSP.
[13]	Distributed job shop scheduling problem (DJSP)	Minimize makespan	The authors proposed a MILP model for the DJSP and compared the performance of the proposed model with the existing model [20] in terms of both size and computational complexities. The results showed that the proposed model outperformed the existing model [20] in terms of both complexities.
[9]	HFSP with unrelated parallel machines (UPM-HFSP)	Minimize makespan	The authors proposed eight MILP models for the UPM-HFSP and compared the performance of them with the existing model [21] in terms of both size and computational complexities to find the best model for the UPM-HFSP.
[22]	Distributed FJSP (DFJSP)	Minimize makespan	The authors proposed four MILP models for the DFJSP and compared the performance of these four models in terms of both size and computational complexities to find the best model for the DFJSP.

The aim of this paper is to propose an improvement of the MILP model for pressing process scheduling in the literature, where the proposed MILP model outperformed the previous MILP model in terms of both size and computational complexities. The motivation of this innovation is that the previous MILP model for pressing process scheduling suffered from its large size complexity. Specifically, it used a large number of binary variables (NBV) and constraints (NC). A lot of binary variables and constraints were used in the previous model to handle the non-overlapping condition in an oven. The previous model also used too many continuous variables to define the starting time of the pressing process phase of a cycle of a press machine. Furthermore, it used a set of variables to define the completion times, which can be essentially eliminated from the model. These lead to a poor model performance as it then requires a large computational time for solving the model (poor computational complexity). This paper aims to enhance these shortcomings. The proposed MILP model is better than the previous MILP model in terms of all three factors of size complexity, where it has the NBV, NCV, and NC less than the previous MILP model has. The proposed MILP model is also better than the previous MILP model in terms of computational complexity because it can find an optimal solution or incumbent solution for the problems in the literature and the newly generated test problems using less computational time than the previous MILP model. The contributions of this paper can be summarized as follows:

- 1. This paper shows an application of MILP for solving pressing process scheduling, which is a problem from real-world industry.
- 2. This paper proposes an improved MILP model (hereafter Model 2) for solving the pressing process scheduling that outperforms the previous model in the literature [10] (hereafter Model 1) in terms of both size and computational complexities. A comparison of the size complexity of both models is summarized in Table 2 in Section 3. For the computational complexity, Model 2 can reduce the computational time from the previous model with an average relative percentage improvement (RPI) of 34.71%.

Туре	Model 1	Model 2
Binary variable	$IKLPT + PTO + P(P-1)T^2O$	$IKLPT + PTO + \frac{P(P-1)}{2}T^2$
Continuous variable	3PT + 2PTO + 1	3PT + 1 2
Constraint	8PT + 2P(T - 1) +2P(P - 1)T2O + 3PTO +IKLPT + I	$\begin{array}{c} 7PT+2P(T-1)+\\ P(P-1)T^2O+IKLPT+I \end{array}$

Table 2. The size complexity of each model.

The remainder of this paper is organized as follows. The problem description of the pressing process is described in Section 2. The proposed Model 2 is presented in Section 3. Numerical results are shown in Section 4. Finally, the discussions and the conclusions are drawn in Section 5 and Section 6, respectively.

2. Problem Description

This section describes the pressing process in multi-layer PCB manufacturing. The purpose of the pressing process is to press the panel, which is the stack of copper foil, prepreg, and core(s). The details of the problem are as follows:

- There is a fixed number of panel types (*I*), and the demand of each panel type (*d*_{*i*}; *i* ∈ {1,2,...,*I*}) is given. The panel types rely on the customers' design.
- The company has a certain number of sizes (*K*) of stainless-steel templates (SST) and a certain number of layouts (*L*). A layout is a pattern of arrangement of panels on a SST. The arrangement of panels on a SST using a layout is called a book.
- Inner and outer gaps are needed when the panels are placed on a SST. The inner gap (g) is the minimal gap among two panels in a book, and the outer gap (G) is the minimal gap between each panel and the borders of the SST. The number of panels on a book depends on the panel size, the SST size, the layout, and the gaps. In general, a PCB company may have its own formula for computing the number of panels on a SST using a layout.
- The overall process of one cycle of a press machine in the pressing process takes 360 min. Figure 1 shows all the processes of a cycle of a press machine, which includes the following three phases:
 - 1. Lay-up process phase: The books are created and inserted into all openings (slots) of the press machine. The number of books that are inserted is equal to the number of openings of the press machine. This phase takes 120 min.
 - 2. Pressing process phase: The press machine, which is already inserted with books, is moved into an oven, and the books are pressed and heated in the oven. After 120 min, the press machine is removed from the oven.
 - 3. Cool-down process phase: The pressed books are cooled down for 120 min. Finally, the books are separated from the press machine to complete one press machine cycle.
- The three phases of the press machine cycle must be worked continually without idle time.
- A press machine can immediately start a new cycle after it has finished the previous cycle. Likewise, an oven is ready for the pressing process phase of another press machine after finishing the current pressing process phase.
- The company has a fixed number of press machines (*P*), each of which has *m* openings, and a fixed number of ovens (*O*). Normally, a PCB company has a small number of ovens because the cost of an oven is very expensive.
- The company considers a planning horizon of 3 d. The maximum number of available cycles (*T*) of each press machine to be operated in the pressing process is given. In reality, this value can be estimated from the order of the customers and the available resources by the production planning department.



Figure 1. The overall processes of one cycle of a press machine.

In addition, this paper has the following two assumptions:

- The demands of panels from the customer can be satisfied within the planning horizon and available resources, i.e., the demands or inputs from the customer have a feasible schedule.
- The number of SSTs is unlimitedly available.

The three main constraints for the pressing process are as follows:

- 1. The books that are pressed in all openings of a cycle of the press machine must have the same specifications, i.e., all books must have the same panel type, the same SST size, and the same layout. This constraint is needed to ensure that the pressure from the press machine will be equally distributed to each panel.
- 2. An oven can perform the pressing process phase of only one press machine at a time.
- 3. The number of outputs (finished goods) of each panel type must be greater than or equal to the demand.

The objective of the pressing process is to minimize the makespan. The minimum makespan implies a good utilization of available resources [23].

3. Improved MILP Model (Model 2)

This section presents Model 2 for pressing process scheduling. The parameters, sets, indices, and variables used in Model 2 are defined below.

Parameters:

- *I* The number of panel types.
- *K* The number of 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 book (or per opening) using SST size *k* and layout *l*.

 d_i The demand of panel type *i*.

M A large positive number.

Sets:

 \hat{I} The set of all panel types, $i \in \hat{I} = \{1, 2, \dots, I\}$.

 \hat{K} The set of all SST sizes, $k \in \hat{K} = \{1, 2, \dots, K\}$.

 \hat{L} The set of all layouts, $l \in \hat{L} = \{1, 2, \dots, L\}$.

 \hat{P} The set of all press machines, $p \in \hat{P} = \{1, 2, \dots, P\}$.

 \hat{O} The set of all ovens, $o \in \hat{O} = \{1, 2, \dots, O\}$.

 \hat{T} The set of all numbers of available cycles of each press machine, $t \in \hat{T} = \{1, 2, ..., T\}$.

Variables:

 x_{iklpt} 1, if panel type *i* is assigned with SST size *k* and layout *l* to press machine *p* at cycle *t*, and 0 otherwise.

 X_{pto} 1, if press machine *p* is sent into oven *o* at cycle *t*, and 0 otherwise.

 $Y_{ptp't'}$ 1, if cycle *t* of press machine *p* is processed before cycle *t'* of press machine *p'* in the same oven, and 0 otherwise.

 A_{pt} The starting time of cycle *t* of press machine *p* (the starting time of the lay-up process phase).

 B_{pt} The starting time of the pressing process phase in cycle *t* of press machine *p*.

 A'_{pt} The auxiliary variable which is equal to A_{pt} if there are a panel type, a SST size, 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 that perform the pressing process (the makespan of the overall process).

Model 2 for scheduling the pressing process is as follows.

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

subject to:

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

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

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

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

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

$$A_{p,t} \ge A_{p,t-1} + 3n, \ \forall p \in \hat{P}, \ \forall t \in \hat{T} - \{1\},$$
(7)

$$B_{pt} = A_{pt} + n, \,\forall p \in \hat{P}, \,\forall t \in \hat{T},$$
(8)

$$B_{p't'} \ge B_{pt} + n - \left(3 - Y_{ptp't'} - X_{pto} - X_{p't'o}\right)M, \forall p, p' \in \hat{P}, \ p < p', \ \forall t, t' \in \hat{T}, \ \forall o \in \hat{O},$$
(9)

$$B_{pt} \ge B_{p't'} + n - \left(2 + Y_{ptp't'} - X_{pto} - X_{p't'o}\right)M,$$

$$\forall p, p' \in \hat{P}, \ p < p', \ \forall t, t' \in \hat{T}, \ \forall o \in \hat{O},$$

$$(10)$$

$$A_{pt} - M\left[1 - \sum_{i=1}^{I} \sum_{k=1}^{K} \sum_{l=1}^{L} x_{iklpt}\right] \le A'_{pt}, \ \forall p \in \hat{P}, \ \forall t \in \hat{T},$$
(11)

$$A'_{pt} \le A_{pt} + M \left[1 - \sum_{i=1}^{L} \sum_{k=1}^{K} \sum_{l=1}^{L} x_{iklpt} \right], \ \forall p \in \hat{P}, \ \forall t \in \hat{T},$$
(12)

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

$$(13)$$

 $C_{max} \ge A'_{pt} + 3n, \, \forall p \in \hat{P}, \, \forall t \in \hat{T},$ (14)

and,

$$\begin{aligned} x_{iklpt} \in \{0,1\} & \forall i \in \hat{I}, \, \forall k \in \hat{K}, \, \forall l \in \hat{L}, \, \forall p \in \hat{P}, \, \forall t \in \hat{I} \\ X_{pto} \in \{0,1\} & \forall p \in \hat{P}, \, \forall t \in \hat{T}, \, \forall o \in \hat{O}, \\ Y_{ptp't'} \in \{0,1\} & \forall p, p' \in \hat{P}, \, p < p', \, \forall t, t' \in \hat{T}, \\ A_{pt} \ge 0 & \forall p \in \hat{P}, \, \forall t \in \hat{T}, \\ B_{pt} \ge 0 & \forall p \in \hat{P}, \, \forall t \in \hat{T}, \\ A'_{pt} \ge 0 & \forall p \in \hat{P}, \, \forall t \in \hat{T}, \end{aligned}$$

 $C_{max} \geq 0$

The objective function (1) is to minimize the makespan of the overall process.

Constraint (2) ensures that at most one type of panel, one size of SST, 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 press machine cycle, it means that all books that are inserted in all openings of this press machine in this cycle are created using this pattern.

Constraint (3) makes sure that if panel type *i* is not compatible with SST size *k* and layout l ($a_{ikl} = 0$), then this pattern cannot be assigned to any press machine cycle.

Constraint (4) enforces that the overall outputs of each panel type from all cycles of all press machines must satisfy the demand.

Constraint (5) ensures that a panel type, a SST size, and a layout must be assigned to consecutive cycles of a press machine. This helps push the empty cycles (the cycles of a press machine with no panel assignment) to be placed after the non-empty cycles (the cycles that really operate the pressing process).

Constraint (6) enforces that any press machine cycle must only be assigned to one oven. Constraint (7) ensures that the starting time of a cycle of a press machine is not less than the completion time of the previous cycle. The completion time of the previous cycle is equal to its starting time plus the processing time 3*n* (the processing time of one cycle). In other words, this constraint enforces that any cycle of a press machine can be started after the previous cycle has been completed.

Constraint (8) sets the starting time of the pressing process phase in cycle t of press machine p to be equal to the starting time of this press machine cycle plus the processing time n that it takes in the lay-up process phase.

Constraints (9) and (10) make sure that the pressing process phase in cycle *t* of press machine *p* and the pressing process phase in cycle *t'* of press machine *p'*, which are assigned in the same oven $o\left(X_{pto} = X_{p't'o} = 1\right)$, cannot be performed simultaneously. It depends on the binary variable $Y_{ptp't'}$. If $Y_{ptp't'} = 1$, the pressing process phase in cycle *t* of press machine *p* precedes the pressing process phase in cycle *t'* of press machine *p'* in oven *o*. Otherwise, the pressing process phase in cycle *t* of press machine *p* is performed after the pressing process phase in cycle *t'* of press machine *p*.

Constraints (11)–(13) enforce that if there is an assignment of a panel type, a SST size, and a layout in press machine p at cycle t, then the auxiliary variable A'_{pt} is equal to A_{pt} . Otherwise, it is equal to 0. The auxiliary variable A'_{pt} will be used in Constraint (14).

8 of 21

Lastly, Constraint (14) determines the maximum completion time of the last cycle of the press machines that really operate the pressing process or the makespan of the overall process.

Compared with Model 1 for scheduling the pressing process [10], Model 2, developed in this paper, has many improved parts as follows.

- The binary variable $Y_{ptp't'o}$ ($Y_{ptp't'o} = 1$ if press machine p at cycle t precedes press machine p' at cycle t' in oven o) in Model 1 is replaced by the new binary variable $Y_{ptp't'}$ in Model 2, where both variables have the same role to avoid an overlapping of any two pressing process phases that are processed in the same oven o. However, the definition of $Y_{ptp't'}$ is without reference to the oven. This can significantly reduce the NBV in the model. Model 2 can still retrieve the oven information from the variables X_{pto} and $X_{p't'o}$. For example, assume that the variables $X_{111} = X_{221} = 1$. This means that press machine 1 at cycle 1 and press machine 2 at cycle 2 are assigned to do the pressing process phase in oven 1. Suppose that $Y_{11221} = 1$ in Model 1 while $Y_{1122} = 1$ in Model 2. The variable $Y_{11221} = 1$ in Model 1 means that press machine 1 at cycle 1 is processed before press machine 2 at cycle 2 in oven 1, while the variable $Y_{1122} = 1$ in Model 2 means that press machine 1 at cycle 1 is processed before press machine 1 at cycle 1 is processed before press machine 2 at cycle 2 in oven 1, while the variable $Y_{1122} = 1$ in Model 2 means that press machine 1 at cycle 1 is processed before press machine 1 at cycle 1 is processed before press machine 2 at cycle 2 in oven 1, while the variable $Y_{1122} = 1$ in Model 2 means that press machine 1 at cycle 1 is processed before press machine 2 at cycle 2 in oven 1, while the variable $Y_{1122} = 1$ in Model 2 does not give the oven information, it can be known from the variables $X_{111} = X_{221} = 1$, i.e., both tasks are processed in the same oven 1.
- The continuous variable B_{ptot} the starting time of the pressing process phase in cycle t of press machine p in oven o, in Model 1 is replaced in Model 2 by the variable B_{pt} , which is without reference to the oven. This replacement can significantly reduce the NCV and eliminate some constraints in Model 1 that were related to the variable B_{pto} . For example, suppose that $X_{pt\tilde{o}} = 1$, which means that press machine *p* at cycle *t* is assigned to oven \tilde{o} . In Model 1, the constraint that forces the variable $B_{pt\bar{o}}$ to be 0 is needed when press machine p at cycle t must not be assigned to oven $\overline{o} \in \hat{O} - {\widetilde{o}}$ so that the model has only the non-zero value of $B_{vt\tilde{o}}$. In Model 2, however, this constraint is not needed because the model defines the variable B_{pt} , which has no oven index. If needed, Model 2 can retrieve the oven information from the variable X_{pto} to find which oven is assigned to press machine *p* at cycle *t* to do the pressing process phase. For example, assume that the variable $X_{221} = 1$, which means that press machine 2 at cycle 2 is assigned to do the pressing process phase in oven 1. Suppose that the variable $B_{221} = 480$ min in Model 1 while the variable $B_{22} = 480$ min in Model 2. The variable $B_{221} = 480$ in Model 1 means that press machine 2 at cycle 2 starts the pressing process phase in oven 1 at time 480 while the variable $B_{22} = 480$ in Model 2 means that press machine 2 at cycle 2 starts the pressing process phase at time 480. Although the variable $B_{22} = 480$ in Model 2 does not give the oven information, it can be known from the variables $X_{221} = 1$, i.e., the press machine 2 at cycle 2 starts the pressing process phase at time 480 in oven 1.
- Model 2 does not use the variables about the completion times. In Model 1, the variables C_{pt} and D_{pto} , the completion time of cycle t of press machine p and the completion time of the pressing process phase in cycle t of press machine p in oven o, respectively, were used. Model 2 can retrieve these values after solving the model where the former is equal to $A_{pt} + 3n$ and the latter is equal to $B_{pt} + n$ (the completion time is equal to the starting time plus the processing time). This can significantly reduce not only the NCV but also the NC in the model because the constraints that are related to the completion time variables are no longer needed.
- Model 1 used too many constraints to handle the non-overlapping conditions in the oven. It formulated the constraint in cases *p* ≠ *p*' to avoid an overlapping of any two pressing process phases in the same oven. However, Model 2 formulates the constraint only in cases *p* < *p*', which can completely define the non-overlapping conditions in the oven. This is because either Constraint (9) or Constraint (10) is active for any two pressing process phases from cycle *t* of press machine *p* and cycle *t*' of press

machine p', which are processed in the same oven $o\left(X_{pto} = X_{p't'o} = 1\right)$. Constraint (9) is active if the task from cycle t of press machine p is processed before the task from cycle t' of press machine p'. Otherwise, Constraint (10) is active if the task from cycle t' of press machine p' is processed before the task from cycle t of press machine p' is processed before the task from cycle t of press machine p. From the cases p < p', it can guarantee that any two pressing process phases from any cycle of two press machines p and p', which are processed in the same oven, cannot overlap. It is clear that the non-overlapping conditions in the oven are completely defined, and the constraint in cases p > p' need not be considered.

Table 2 shows the NBV, NCV, and NC that were used to formulate Model 1 and Model 2. An evaluation of Model 2 in comparison to Model 1 is presented in Section 4.

4. Numerical Results

To evaluate the proposed MILP model, the previous data and test problems [10] were used. Furthermore, additional larger-sized (than in [10]) test problems were also generated in this paper. All problems were solved using both the previous Model 1 [10] and the proposed Model 2, and the results from both models were compared. The data and test problems are shown in Section 4.1, and the results from both models are shown in Section 4.2.

4.1. Data and Test Problems

The same data previously used for Model 1 [10], which were acquired from an actual PCB company, were used in this study and consisted of the following. The number of panel types (I), the number of SST sizes (K), the number of layouts (L), the number of press machines (P), and the number of ovens (O) were seven, six, eight, six, and three, respectively. The number of openings of each press machine (m) was 10. The sizes of each panel type, which included the warp (a) and fill (b) as well as the inner gap (g) and outer gap (*G*), are shown in Table 3. The sizes of each SST, which included the warp (X)and fill (Y), are shown in Table 4. Figure 2 shows the illustration of eight layouts [10]. Note that Figure 2 illustrates only examples that show the direction of the arrangement of panels on a SST. The number of panels in the book is not restricted to those shown in the illustration. For example, Figure 2a illustrates the first layout with two horizontal sections, and the panels are arranged vertically in each section. The company had the formulas for computing the number of panels of type $i \in I$ per book based on SST size $k \in K$ and layout $l \in \hat{L}(a_{ikl})$, which are shown in Table 5. It should be noted that the notation x is the largest integer that is less than or equal to x. The processing time of each phase in the pressing process (n) was assumed to be 120 min. In addition, the maximum number of available cycles (T) of each press machine was 12. This is because the processing time of one cycle of a press machine is 360 min (6 h). If a press machine operates continually, it can conduct up to 12 cycles of the pressing process in 3 d.

Table 3. 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 4. Sizes of each SST.



Figure 2. Illustration of the eight layouts (a–h) of the panel arrangement [10].

Table 5. Formulas for computing a_{ikl} , which is the number of panels of type $i \in \{1, 2, ..., I\}$ per book using SST size $k \in \{1, 2, ..., K\}$, and layout $l \in \{1, 2, ..., L\}$.

Layout (<i>l</i>)	a _{ikl}
1	$\left \frac{X-2(G-\frac{\delta}{2})}{a+s} \right \times \left \frac{Y-2(G-\frac{\delta}{2})}{b+s} \right $
2	$\mid \frac{X-2(\overset{\circ}{G}-\overset{\circ}{2})}{b+g} \mid \times \mid \frac{Y-2(\overset{\circ}{G}-\overset{\circ}{2})}{a+g} \mid$
3	$\mid \frac{X-2(G-\frac{g}{2})}{a+g} \mid + \left(\mid \frac{X-2(G-\frac{g}{2})}{b+g} \mid \times \mid \frac{Y-b-G-2(G-\frac{g}{2})}{a+g} \mid \right)$
4	$\left \frac{Y-2\left(G-\frac{g}{2}\right)}{a+g} \right + \left(\left \frac{Y-2\left(G-\frac{g}{2}\right)}{b+g} \right \times \left \frac{X-b-G-2\left(G-\frac{g}{2}\right)}{a+g} \right \right)$
5	$\mid \frac{X-2(G-\frac{g}{2})}{b+g} \mid + \left(\mid \frac{X-2(G-\frac{g}{2})}{a+g} \mid \times \mid \frac{Y-a-G-2(G-\frac{g}{2})}{b+g} \mid \right)$
6	$\left \frac{Y-2(G-\frac{g}{2})}{b+g} \right + \left(\left \frac{Y-2(G-\frac{g}{2})}{a+g} \right \times \left \frac{X-a-G-2(G-\frac{g}{2})}{b+g} \right \right)$
7	$\left \frac{X-2(G-\frac{g}{2})}{a+g} \right $
8	$\mid \frac{X-2(\widetilde{G-\frac{g}{2}})}{b+g} \mid$

Moreover, the planning horizons of 2 and 1.5 d were also considered in [10] for different problem sizes, where the maximum number of available cycles of each press machine was eight and six, respectively. The previous test problems [10] were generated using the acquired data, and the parameters P and O in some test problems were also slightly different from the real data for a variety of the problem sizes. The demand of each panel type i (d_i) in each problem was randomly generated. The test problems were categorized into the three types of small, medium, and large problems, according to the NBV, and were totaled five, eight, and nine problems, respectively, as shown in Tables 6–8. A problem that has three panel types means that there are panel types 1–3 (in Table 3). Similarly, a problem that has four, five, six, or seven panel types means that there are panel types 1–4, 1–5, 1–6, or 1–7 (in Table 3), respectively.

No.	Ι	K	L	Р	0	Т	d_i , $i \in I$
S1	3	6	8	3	2	6	110, 150, 125
S2	3	6	8	3	2	8	200, 220, 230
S3	3	6	8	3	2	12	270, 250, 210
S4	3	6	8	4	2	6	110, 150, 125
S5	3	6	8	4	3	6	110, 150, 125

Table 6. The small problems.

Table 7. The medium problems.

No.	Ι	K	L	Р	0	Т	$d_i, i \in \hat{I}$
M1	3	6	8	6	3	6	300, 300, 300
M2	3	6	8	6	3	8	450, 480, 500
M3	3	6	8	6	3	12	720, 900, 600
M4	4	6	8	6	3	6	200, 300, 400, 100
M5	4	6	8	6	3	8	300, 400, 200, 500
M6	4	6	8	6	3	12	500, 700, 700, 500
M7	5	6	8	6	3	6	200, 250, 200, 250, 200
M8	5	6	8	6	3	8	400, 300, 200, 250, 300

Table 8. The large problems.

No.	Ι	K	L	Р	0	Т	$d_i, i \in I$
L1	5	6	8	6	3	12	500, 500, 500, 500, 500
L2	5	6	8	7	3	12	500, 500, 500, 500, 500
L3	5	6	8	6	4	12	500, 500, 500, 500, 500
L4	6	6	8	6	3	12	500, 360, 220, 180, 380, 720
L5	6	6	8	7	3	12	500, 360, 220, 180, 380, 720
L6	6	6	8	6	4	12	500, 360, 220, 180, 380, 720
L7	7	6	8	6	3	12	300, 325, 290, 425, 450, 475, 200
L8	7	6	8	7	3	12	300, 325, 290, 425, 450, 475, 200
L9	7	6	8	6	4	12	300, 325, 290, 425, 450, 475, 200

Furthermore, in this paper, additional test problems were generated using the data in [10]. The planning horizon of these problems was 4 d, where the maximum number of available cycles (T) of each press machine was 16. The demand of each panel type in each problem was randomly generated. The additional test problems are shown in Table 9. Similarly, a problem that has five, six, or seven panel types means that there are panel types 1–5, 1–6, or 1–7 (in Table 3), respectively.

Table 9. The additional test problems.

No.	Ι	K	L	Р	0	Т	$d_i, i \in I$
A1	5	6	8	6	3	16	660, 525, 740, 850, 480
A2	5	6	8	7	3	16	660, 525, 740, 850, 480
A3	5	6	8	6	4	16	660, 525, 740, 850, 480
A4	6	6	8	6	3	16	400, 495, 800, 630, 700, 800
A5	6	6	8	7	3	16	400, 495, 800, 630, 700, 800
A6	6	6	8	6	4	16	400, 495, 800, 630, 700, 800
A7	7	6	8	6	3	16	500, 420, 595, 375, 330, 680, 580
A8	7	6	8	7	3	16	500, 420, 595, 375, 330, 680, 580
A9	7	6	8	6	4	16	500, 420, 595, 375, 330, 680, 580

4.2. Computational Results

To compare the performance between Model 1 and Model 2, the two widely used performance measures of size and computational complexities were used. All problems were solved using ILOG OPL CPLEX 12.6 software running on a personal computer with a core i7 2.20 GHz CPU and 8 GB RAM, within a maximum time limit of 2 h.

4.2.1. Computational Results of the Small Problems

The model size, which included the NBV, the NCV, and the NC as well as the computational results of each small problem using Model 1 and Model 2, are shown in Table 10. The results included the number of outputs (finished goods) of each panel type, CPU time, and the makespan (C_{max}) of the overall process. The last column in Table 10 reports the RPI_{time} or the relative percentage improvement of the CPU times of Model 2 over Model 1, which is defined as $\frac{CPUtime_{Model1} - CPUtime_{Model1}}{CPUtime_{Model1}} \times 100\%$.

Table 10. Computational results of the small problems using Model 1 and Model 2.

			Resu	ts Using Model	1		Results Using Model 2							
No.	Model Size			Results]	Model Si	ze	Re	Results			
	NBV	NCV	NC	Outputs	CPU Time	C _{max} (Min)	NBV	NCV	NC	Outputs	CPU Time	C _{max} (Min)	KP1 _{time}	
S1	3060	127	3741	120, 160, 160	2.31 s	1440 ^a	2736	55	3183	120, 160, 160	1.51 s	1440 ^a	34.63%	
S2	4272	169	5373	200, 240, 240	2.48 s	2160 ^a	3696	73	4437	200, 240, 240	1.76 s	2160 ^a	29.03%	
S3	6984	253	9213	280, 280, 240	3.21 s	2520 ^a	5688	109	7233	280, 280, 240	1.82 s	2520 ^a	43.30%	
S4	4368	169	5563	120, 160, 160	8.15 s	1200 ^a	3720	73	4531	120, 160, 160	5.29 s	1200 a	35.09%	
S5	4824	217	6499	120, 160, 160	3.18 s	1080 ^a	3744	73	4963	120, 160, 160	1.84 s	1080 ^a	42.14%	
												Average	36.84%	

^a Optimal solution. The better values are displayed in boldface.

As shown in Table 10, the NBV, NCV, and NC in Model 2 from each small problem were lower than those in Model 1. Figure 3a–c shows the NBV, NCV, and NC, respectively, for each model generated in each small problem. This reveals that Model 2 outperformed Model 1 in terms of the size complexity for the small problems. Both Model 1 and Model 2 could solve all the small problems to an optimal solution. The optimal makespan could be found, and the demand of each panel type was satisfied. For example, the optimal makespan of Problem S1 was 1,440 min, and the outputs of panel types 1–3 were 120, 160, and 160, respectively, which is greater than or equal to the demands of Problem S1 shown in Table 6. However, Model 2 could find an optimal solution for all small problems at a smaller CPU time. Figure 3d demonstrates the CPU time for solving each small problem using Model 1 and Model 2, where Model 2 also outperformed Model 1 in terms of the computational complexity for the small problems. From the last row of Table 10, Model 2 provided a better CPU time with an average RPI_{time} of 36.84% for the small problems.

4.2.2. Computational Results of the Medium Problems

Table 11 shows the NBV, NCV, NC, and the results of each medium problem using Model 1 and Model 2. The number of NBV, NCV, and NC that Model 2 generated in each medium problem was less than those that Model 1 generated. Figure 4a–c shows the NBV, NCV, and NC, respectively, that each model generated for each medium problem, where Model 2 outperformed Model 1 in terms of the size complexity for the medium problems. Both models could find an optimal solution for all medium problems, but Model 2 could find the optimal solution using a smaller CPU time in all problems with an average RPI_{time} of 69.98%. The CPU times used for solving each medium problem using Model 1 and Model 2 are shown in Figure 4d, where Model 2 also outperformed Model 1 in terms of the computational complexity for the medium problems.



Figure 3. Comparison of model sizes, as the (**a**) NBV (number of binary variables), (**b**) NCV (number of continuous variables), and (**c**) NC (number of constraints) and (**d**) the CPU times for solving the small problems between Model 1 and Model 2.

 Table 11. Computational results of the medium problems using Model 1 and Model 2.

			Re	esults Using Model 1				Results Using Model 2							
NT	M	odel Siz	ze	Re	esults			Model Si	ze	I	Results		- 		
140.	NBV	NCV	NC	Outputs	CPU C _m . Time (Mi		NBV	NCV	NC	Outputs	CPU Time	C _{max} (Min)	- KP1 _{time}		
M1	8532	325	12,339	320, 320, 320	32.81 s	1560 a	5832	109	8739	320, 320, 320	13.93 s	1560 a	57.54%		
M2	12,816	433	19,335	480, 480, 520	16.79 s	2520 ^a	8016	145	13,095	480, 480, 520	6.29 s	2520 ^a	62.54%		
M3	23,544	649	37,647	720, 920, 600	1 m 28 s	3600 ^a	12,744	217	23,967	720, 920, 600	25.46 s	3600 ^a	71.07%		
M4	10,260	325	14,068	200, 320, 400, 120	20.65 s	1800 ^a	7560	109	10,468	200, 320, 400, 120	2.40 s	1800 ^a	88.38%		
M5	15,120	433	21,640	320, 400, 200, 520	3 m 59 s	2280 a	10,320	145	15,400	320, 400, 200, 520	55.21 s	2280 a	76.90%		
M6	27,000	649	41,104	520, 720, 720, 520	4 m 2 s	3960 ^a	16,200	217	27,424	520, 720, 720, 520	20.48 s	3960 ^a	91.54%		
M7	11,988	325	15,797	200, 280, 200, 280, 200	9 m 31 s	1920 ^a	9288	109	12,197	200, 280, 200, 280, 200	6 m 59 s	1920 ^a	26.62%		
M8	17,424	433	23,945	400, 320, 200, 280, 320	49.31 s	2520 ^a	12,624	145	17,705	400, 320, 200, 280, 320	7.26 s	2520 ^a	85.28%		
												Average	69.98%		

^a Optimal solution. The better values are displayed in boldface.



Figure 4. Comparison of the model size, as the (**a**) NBV (number of binary variables), (**b**) NCV (number of continuous variables), and (**c**) NC (number of constraints) and (**d**) the CPU times for solving the medium problems between Model 1 and Model 2.

4.2.3. Computational Results of the Large Problems

Table 12 shows the NBV, NCV, NC, and the results for each large problem using Model 1 and Model 2. The results included the outputs, CPU time, %gap, and makespan (C_{max}) . In Table 12, the column "CPU time" denotes the computational time taken by CPLEX to solve the MILP model to reach optimality, or 2 h if CPLEX cannot solve the MILP model optimally within the time limit. The %gap is the relative gap tolerance of the objective value for the solution from CPLEX, which is defined as $\frac{|BestBound-BestInteger|}{RestInteger}$ × BestInteger 100%. The BestInteger is the current best feasible solution (or incumbent solution) that could be found by the model within the time limit, and the *BestBound* is the current lower bound that the model could obtain within the time limit. The smaller the %gap value is, the better performance the model will be. If the problem could be solved to reach optimality, then the %gap is equal to 0%. Note that, for the solution that we get within the time limit and $\text{\%gap} \neq 0$, it needs to use more computational time than 2 h for solving the problem to reach optimality. The column " C_{max} " reports the optimal makespan or the incumbent makespan that could be found within the time limit. Table 12 also reports time_{inc}, which is the computational time to reach the incumbent solution. For the problem that could be solved optimally, the time_{inc} is the time to reach the incumbent solution, which is an optimal solution, but the status of the problem at that time is not at optimality because the %gap is not equal to 0 at that time. The last column in Table 12 reports RPI_{time} and RPI^{inc}, where the RPI^{inc} is the relative percentage improvement of time_{inc} of Model 2 over Model 1, and is defined as $\frac{(\text{time}_{\text{inc}})_{\text{Model1}} - (\text{time}_{\text{inc}})_{\text{Model2}}}{(\text{time}_{\text{inc}})_{\text{Model1}}} \times 100\%.$

As shown in Table 12, the NBV, NCV, and NC in Model 2 for each large problem were lower than those in Model 1. Figure 5a–c shows the NBV, NCV, and NC, respectively, that each model generated in each large problem. Model 2 clearly outperformed Model 1 in terms of the size complexity for the large problems. Compared to Figures 3a–c and 4a–c, the differences in the NBV and NC between the two models become more significant when the problem size increased from small to large.



Figure 5. Comparison of the model size, as the (**a**) NBV (number of binary variables), (**b**) NCV (number of continuous variables), and (**c**) NC (number of constraints) and the (**d**) CPU time and (**e**) time_{inc} (computational time to reach the incumbent solution) for solving the large problems between Model 1 and Model 2.

				Results	Using Model	1						Results Usi	ng Model 2				
No	Ν	Iodel Siz	ze]	Results			Model Size					Results			BPL. /BPIinc
100.	NBV	NCV	NC	Outputs	CPU Time	%gap	C _{max} (Min)	Time _{inc}	NBV	NCV	NC	Outputs	CPU Time	%gap	C _{max} (Min)	Time _{inc}	time
L1	30,456	649	44,561	520, 520, 520, 520, 520	44 m 38 s	0%	4080 ^a	4 m 44 s	19,656	217	30,881	520, 520, 520, 520, 520	23 m 34 s	0%	4080 ^a	33.68 s	47.20%/88.14%
L2	38,556	757	58,035	520, 520, 520, 520, 520	23 m 42 s	0%	3600 ^a	6 m 42 s	23,436	253	39,051	520, 520, 520, 520, 520	4 m 43 s	0%	3600 ^a	2 m 50 s	80.10%/57.71%
L3	34,848	793	53,417	520, 520, 520, 520, 520	2 h	2.94%	4080 ^b	5 m 13 s	19,728	217	35,201	520, 520, 520, 520, 520	2 h	2.94%	4080 ^b	1 m 5 s	0%/79.23%
L4	33,912	649	48,018	520, 360, 240, 200, 400, 770	48 m 14 s	0%	3360 ^a	16 m 7 s	23,112	217	34,338	520, 360, 240, 200, 400, 770	6 m 23 s	0%	3360 ^a	22.55 s	86.77%/97.67%
L5	42,588	757	62,068	520, 360, 240, 200, 400, 770	2 h	4%	3000 ^b	8 m 19 s	27,468	253	43,084	520, 360, 240, 200, 400, 770	2 h	4%	3000 ^b	1 m 54 s	0%/77.15%
L6	38,304	793	56,874	520, 360, 240, 200, 400, 770	2 h	3.57%	3360 ^b	6 m 21 s	23,184	217	38,658	520, 360, 240, 200, 400, 770	1 h 47 m 55 s	0%	3360 ^a	24.77 s	10.07%/93.50%
L7	37,368	649	51,475	320, 360, 320, 440, 480, 490, 200	2 h	1.61%	3720 ^b	16 m 32 s	26,568	217	37,795	320, 360, 320, 440, 480, 490, 200	1 h 29 m 2 s	0%	3720 ^a	1 m 15 s	25.81%/92.44%
L8	46,620	757	66,101	320, 360, 320, 440, 480, 490, 200	2 h	3.57%	3360 ^b	10 m 36 s	31,500	253	47,117	320, 360, 320, 440, 480, 490, 200	2 h	3.57%	3360 ^b	2 m 24 s	0%/77.36%
L9	41,760	793	60,331	320, 360, 320, 440, 480, 490, 200	2 h	3.23%	3720 ^b	6 m 9 s	26,640	217	42,115	320, 360, 320, 440, 480, 490, 200	1 h 18 m 41 s	0%	3720 ^a	1 m 31 s	34.43%/75.34%
															Ave	erage	31.60%/82.06%

Table 12. Computational results of the large problems using Model 1 and Model 2.

^a Optimal solution. ^b The incumbent solution that could be found within the time limit. The better values are displayed in boldface.

Furthermore, from Table 12, Model 1 could solve only three large problems optimally within the time limit (Problem L1, L2, and L4), whereas Model 2 could optimally solve these problems using a smaller CPU time. For Problem L6, L7, and L9, Model 1 could not solve them to reach optimality within the time limit since the %gap was not equal to 0%. However, Model 2 could solve these problems to reach optimality within the time limit (%gap = 0%). For Problem L3, L5, and L8, neither models could solve them to reach optimality. The makespan of the incumbent solution from Model 2 was equal to that from Model 1 (the incumbent solutions from both models have the same quality), and the values of %gap from both models were also the same in each problem. However, Model 2 used a smaller time_{inc} than Model 1 to reach the incumbent solution for these problems.

Figure 5d,e shows the CPU time and time_{inc}, respectively, for solving each large problem using Model 1 and Model 2. From Figure 5d, solving Problem L3, L5, and L8 using Model 1 and Model 2 was not achieved within the 2-h time limit, but a significant difference in the CPU times between the two models was evident for the other problems. From Figure 5e, moreover, Model 2 used a smaller time_{inc} to reach the incumbent solution for all the large problems. The average RPI_{time} and RPI^{inc}_{time} values were 31.60% and 82.06%, respectively. These show that Model 2 outperformed Model 1 in terms of computational complexity for the large problems.

In a real-life application, any PCB manufacturing company may not need to solve a problem to reach optimality if it takes a long computational time, but the company needs to find a promising solution for the problem as fast as possible. The time_{inc} of each large problem using Model 2 was small and acceptable in a real-life application. The maximum value of time_{inc} to reach the incumbent for large problems using Model 2 is only 2 min 50 s (in Problem L2). Hence, Model 2 can fulfill this preference and can be a practical option to provide a promising schedule for the pressing process in any PCB manufacturing industry.

4.2.4. Computational Results of the Additional Test Problems

Table 13 shows the NBV, NCV, NC, and the results for each additional test problem using Model 1 and Model 2. Model 2 had lower NBV, NCV, and NC in each problem than Model 1. Figure 6a–c shows the NBV, NCV, and NC, respectively, that each model generated in each additional test problem. This revealed that Model 2 clearly outperformed Model 1 in terms of size complexity for the additional test problems.

In addition, from Table 13, neither models could solve all the additional test problems optimally within the 2-h time limit, except for Model 2, which could solve Problem A1 optimally. For the problems that could not be solved optimally, both models could find the incumbent solution with the same value of makespan and %gap, except for Problem A2 and A8. For Problem A2, each model could find the incumbent solution with the same value of makespan, but the %gap from Model 2 (4.39%) was smaller than that from Model 1 (5.26%). In Problem A8, Model 2 could find a better incumbent solution ($C_{max} = 4440$ min) than that by Model 1 ($C_{max} = 4560$ min). For Problem A3–A7 and A9, the incumbent solutions from both models have the same quality, but Model 2 used less time_{inc} than Model 1 to reach the incumbent solution for these problems.

	Results Using Model 1								Results Using Model 2								
No.	Model Size			Results				Model Size					Results				- PPL. /PPIinc
	NBV	NCV	NC	Outputs	CPU Time	%gap	C _{max} (Min)	Time _{inc}	NBV	NCV	NC	Outputs	CPU Time	%gap	C _{max} (Min)	Time _{inc}	- Miltime/Miltime
A1	46,368	865	70,937	680, 560, 760, 880, 480	2 h	4.55%	5280 ^b	12 m 49 s	27,168	289	46,937	680, 560, 760, 880, 480	1 h 2 m 58 s	0%	5160 ^a	1 m 13 s	47.53%/90.51%
A2	59,472	1009	93,511	680, 560, 760, 880, 480	2 h	5.26%	4560 ^b	48 m 35 s	32,592	337	60,135	680, 560, 760, 880, 480	2 h	4.39%	4560 ^b	3 m 43 s	0%/92.35%
A3	54,144	1057	86,585	680, 560, 760, 880, 480	2 h	2.33%	5160 ^b	16 m 4 s	27,264	289	54,617	680, 560, 760, 880, 480	2 h	2.33%	5160 ^b	1 m 19 s	0%/91.80%
A4	50,976	865	75,546	400, 520, 800, 640, 720, 840	2 h	2.17%	5520 ^b	10 m 30 s	31,776	289	51,546	400, 520, 800, 640, 720, 840	2 h	2.17%	5520 ^b	1 m 44 s	0%/83.49%
A5	64,848	1009	98 <i>,</i> 888	400, 520, 800, 640, 720, 840	2 h	2.50%	4800 ^b	1 h 18 m 26 s	37,968	337	65,512	400, 520, 800, 640, 720, 840	2 h	2.50%	4800 ^b	2 m 32 s	0%/96.77%
A6	58,752	1057	91,194	400, 520, 800, 640, 720, 840	2 h	2.17%	5520 ^b	14 m 59 s	31,872	289	59,226	400, 520, 800, 640, 720, 840	2 h	2.17%	5520 ^b	1 m 11 s	0%/92.10%
A7	55,584	865	80,115	520, 440, 600, 400, 360, 700, 600	2 h	2.33%	5160 ^b	16 m 9 s	36,384	289	56,155	520, 440, 600, 400, 360, 700, 600	2 h	2.33%	5160 ^b	1 m 18 s	0%/91.95%
A8	70,224	1009	104,265	520, 440, 600, 400, 360, 700, 600	2 h	5.26%	4560 ^b	44 m 4 s	43,344	337	70,889	520, 440, 600, 400, 360, 700, 600	2 h	2.70%	4440 ^b	14 m 16 s	0%/67.62%
A9	63,360	1057	95,803	520, 440, 600, 400, 360, 700, 600	2 h	2.33%	5160 ^b	13 m 30 s	36,480	289	63,835	520, 440, 600, 400, 360, 700, 600	2 h	2.33%	5160 ^b	1 m 7 s	0%/91.72%
																Average	5.28%/88.70%

Table 13. Computational results of the additional test problems using Model 1 and Model 2.

^a Optimal solution. ^b The incumbent solution that could be found within the time limit. The better values are displayed in boldface.



Figure 6. Comparison of the model size, as the (**a**) NBV (number of binary variables), (**b**) NCV (number of continuous variables), and (**c**) NC (number of constraints) and the (**d**) CPU time and (**e**) time_{inc} (computational time to reach the incumbent solution) for solving the additional test problems between Model 1 and Model 2.

Figure 6d demonstrates the CPU time for solving each additional test problem using Model 1 and Model 2. The CPU time for solving Problem A1 using Model 2 was smaller than that using Model 1, but the CPU times for solving the other problems using both models were 2 h because they could not verify optimality within the time limit. However, Model 2 used a smaller time_{inc} to reach the incumbent solution for all additional test problems (Figure 6e). Thus, Model 2 saved a large amount of computational time to reach the incumbent solution, compared with Model 1, for all problems, where the quality of the incumbent solution from Model 2 was no worse than that from Model 1. For example, in Problem A8, Model 2 used a time_{inc} of 14 min 16 s to reach the incumbent solution $(C_{max} = 4440 \text{ min})$ compared to 44 min 4 s $(C_{max} = 4560 \text{ min})$ for Model 1. The time_{inc} used in the other problems using Model 2 varied in the range of only 1-4 min. The difference in time_{inc} between Model 1 and Model 2 is particularly prominent in Problem A2 and A5. Model 2 used a time_{inc} of only 3 min 43 s and 2 min 32 s to reach the incumbent solution of Problem A2 and A5, respectively, compared to 48 min 35 s and 78 min 26 s, respectively, for Model 1 with the same makespan. In addition, the average RPI_{time} and RPI^{inc}_{time} were 5.28% and 88.70%, respectively. These show that Model 2 is very efficient and effective at finding a good solution within a small computational time and so can be a practical option in the real PCB manufacturing industry.

Moreover, the results from Model 2 can give useful information. For example, from Problem A1–A3 in Table 9, all the parameters in the problems were the same except for the number of press machines and ovens, where the number of press machines and ovens were increased (from Problem A1) by one in Problem A2 and A3, respectively. The results of these problems (Table 13) showed that the makespan of Problem A1–A3 were 5160 min, 4560 min, and 5160 min, respectively. This means that increasing the number of press machines by one from Problem A1 could reduce the makespan from 5160 to 4560 min, but increasing the number of ovens by one from Problem A1 could not reduce the makespan. Thus, Model 2 can help in deciding which resources should be increased to reduce the production time.

5. Discussions

Model 2 was found to outperform Model 1 in terms of all three structural elements of size complexity (the NBV, NCV, and NC). This is because Model 2 used the binary variable $Y_{ptp't'}$ and the continuous variable B_{pt} instead of $Y_{ptp't'_0}$ and B_{pt_0} in Model 1, respectively, which can reduce the NBV and NCV in the model, but the meaning of both models remained the same. It can also reduce the NC that are related to the index of the oven. In addition, the dimensionalities of the two decision variables were also reduced because the variables $Y_{ptp't'_0}$ and B_{pt_0} have five and three indices, but the variables $Y_{ptp't'}$ and B_{pt} have only four and two indices, respectively. Moreover, Model 2 did not use the variables that were related to the completion time and could reduce the NC to handle the non-overlapping conditions in an oven. Some examples of these replacements and eliminations are provided in Section 3. Since the most deciding factor that affects the performance of a MILP is the NBV, and the next influential factor is the NC [11], reducing these in Model 2 can help to solve large-sized problems more efficiently.

Furthermore, Model 2 used less computational time than Model 1 for solving all the problems that could be solved to reach optimality. It could also newly verify an optimal solution for some large problems and additional test problems that Model 1 could not. The average RPI_{time} of the small, medium, large, and additional test problems were 36.84%, 69.98%, 31.60%, and 5.28%, respectively, where the average RPI_{time} of all problems is $\frac{5(36.84)+8(69.98)+9(31.60)+9(5.28)}{5+8+9+9} = 34.71\%$. In addition, Model 2 used a smaller computational time to find the incumbent solution for all the problems that could not be solved optimally within the 2-h time limit, where the incumbent solution from Model 1. The average RPI_{time} of the large and additional test problems were 82.06% and 88.70%, respectively. Thus, Model 2 also outperformed Model 1 in terms of computational complexity. Note that, although the average RPI_{time} of the additional test problems is small, because most of them could not be solved optimally, the average RPI_{time} is still very high. This shows that Model 2 is very effective at finding the incumbent solution for large-sized problems within a relatively short time.

The computational times of Model 2 to reach the incumbent solution of large problems with a planning horizon of 3 d and additional test problems with a planning horizon of 4 d are small and practicable. In the real situation, a PCB company needs to find a good schedule for the pressing process scheduling within a small computational time, and an incumbent solution can be just what they need. Hence, in practice, a PCB company can use Model 2 to find an optimal solution or a good feasible solution for pressing process scheduling by setting a small time limit, such as 5 min, or by setting a small %gap, such as 5%.

Moreover, the proposed Model 2 has implications for both theory and practice. In theory, the techniques to reduce the model size in this paper can be applied to other similar scheduling problems. In practice, the following are examples of the real-world situations that can be beneficial from Model 2.

1. A PCB company which manually schedules the pressing process can use the proposed model to find an optimal schedule or a good feasible schedule for the pressing process

n d te c d

scheduling. The manager of the PCB company can compare the solution from the proposed model with the manual schedule and choose the better schedule to proceed in the real situation.

2. In case of a rush order, a PCB company can use the proposed model to decide which resources should be increased to reduce the production time before proceeding in the real situation. For example, the manager of the PCB company can try to increase the parameters in the proposed model (e.g., the number of press machines or the number of ovens) and find how much the makespan can be reduced.

6. Conclusions

This study investigated pressing process scheduling, which is a real-world application in the PCB manufacturing industry. An application of MILP for pressing process scheduling is presented. The objective of the problem was to minimize the makespan. This objective can imply maximizing the utilization of the resources, which is usually the main objective of any PCB manufacturing enterprise.

The aim of this study was to improve the existing MILP model (Model 1 [10]) for solving pressing process scheduling. The obtained improvement (Model 2) revealed the possible application of MILP to handle an application in the real-world industry. The development of MILP models is interesting because the MILP model has the benefit that it can guarantee an optimal solution if one exists. MILP is also an approach that is widely used to cope with problems coming from organizations or industries. In this study, the previous Model 1 [10] and the improved Model 2 of this study were used to solve problems in the literature and additional test problems that were newly generated, and then, the computational results from both models were compared. The models were compared in terms of both size and computational complexities. The results show that Model 2 outperformed Model 1 in terms of both complexities. For the problems that could be solved optimally by Model 1, Model 2 could also solve them optimally but using less computational time. Furthermore, Model 2 could newly verify the optimal solution for some large problems and some additional test problems within the 2-h time limit. Model 2 also used an acceptable computational time (and less than Model 1) to find the incumbent solution for all the problems that could not be solved to reach optimality, where the makespan of the incumbent solution from Model 2 was less than or equal to the makespan of that from Model 1. In actual practice, a PCB company needs to use a small computational time to provide an effective schedule for the problem at hand. Therefore, Model 2 is suitable to use in actual practice.

The limitation of this approach is that the MILP model is not effective for solving pressing process scheduling problems with large planning horizons, such as, a period of a month. This is because the size of the model will become very large, which leads to a very long computational time. To be practical, it needs to use other methods, such as heuristic and metaheuristic algorithms, for solving long-term pressing process scheduling.

For future research, some extensions of the MILP model for the pressing process with additional constraints could be of interest. For example, the panel types have different cycle times and different due dates, one cycle of a press machine can press a group of compatible panel types, and press machines or ovens require machine maintenance. These additional constraints increase the complication of the problem, but it can be very challenging for further research. Another future research line is to develop heuristic or metaheuristic algorithms for solving pressing process scheduling problems using the solutions from the MILP model in this paper as references.

Author Contributions: Conceptualization, T.L., B.I., and C.J.; methodology, T.L., B.I., and C.J.; software, T.L.; 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 the 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. 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]
- 2. Qin, W.; Zhuang, Z.; Liu, Y.; Tang, O. A two-stage ant colony algorithm for hybrid flow shop scheduling with lot sizing and calendar constraints in printed circuit board assembly. *Comput. Ind. Eng.* **2019**, *138*, 106115. [CrossRef]
- 3. Khandpur, R.S. Printed Circuit Boards: Design, Fabrication, Assembly and Testing; McGraw-Hill: New York, NY, USA, 2006.
- 4. Pezzella, F.; Morganti, G.; Ciaschetti, G. A genetic algorithm for the flexible job-shop scheduling problem. *Comput. Oper. Res.* **2008**, *35*, 3202–3212. [CrossRef]
- 5. 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]
- 6. Demir, Y.; Isleyen, S.K. Evaluation of mathematical models for flexible job-shop scheduling problems. *Appl. Math. Model.* **2013**, 37, 977–988. [CrossRef]
- 7. Roshanaei, V.; Azab, A.; ElMaraghy, H. Mathematical modelling and a meta-heuristic for flexible job shop scheduling. *Int. J. Prod. Res.* **2013**, *51*, 6247–6274. [CrossRef]
- 8. Naderi, B.; Gohari, S.; Yazdani, M. Hybrid flexible flowshop problems: Models and solution methods. *Appl. Math. Model.* 2014, 38, 5767–5780. [CrossRef]
- 9. Meng, L.; Zhang, C.; Shao, X.; Zhang, B.; Ren, Y.; Lin, W. More MILP models for hybrid flow shop scheduling problem and its extended problems. *Int. J. Prod. Res.* 2020, *58*, 3905–3930. [CrossRef]
- 10. 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. [CrossRef]
- 11. Pan, C.H. A study of integer programming formulations for scheduling problems. Int. J. Syst. Sci. 1997, 28, 33-41. [CrossRef]
- 12. Unlu, Y.; Mason, S.J. Evaluation of mixed integer programming formulations for non-preemptive parallel machine scheduling problems. *Comput. Ind. Eng.* **2010**, *58*, 785–800. [CrossRef]
- 13. Naderi, B.; Azab, A. An improved model and novel simulated annealing for distributed job shop problems. *Int. J. Adv. Manuf. Technol.* **2015**, *81*, 693–703. [CrossRef]
- 14. Chen, J.S.; Yang, J.S. Model formulations for the machine scheduling problem with limited waiting time constraints. *J. Inf. Optim. Sci.* **2006**, *27*, 225–240. [CrossRef]
- 15. Naderi, B.; Fatemi Ghomi, S.M.T.; Aminnayeri, M.; Zandieh, M. A study on open shop scheduling to minimise total tardiness. *Int. J. Prod. Res.* 2011, *49*, 4657–4678. [CrossRef]
- 16. Mousakhani, M. Sequence-dependent setup time flexible job shop scheduling problem to minimise total tardiness. *Int. J. Prod. Res.* **2013**, *51*, 3476–3487. [CrossRef]
- 17. Mehrabad, M.S.; Fattahi, P. Flexible job shop scheduling with tabu search algorithms. *Int. J. Adv. Manuf. Technol.* **2007**, *32*, 563–570. [CrossRef]
- 18. Fattahi, P.; Mehrabad, M.S.; Jolai, F. Mathematical modeling and heuristic approaches to flexible job shop scheduling problems. *J. Intell. Manuf.* 2007, *18*, 331–342. [CrossRef]
- 19. Roshanaei, V.; ElMaraghy, H.; Azab, A. Sequence-based MILP modeling for flexible job shop scheduling. In Proceedings of the IIE Conference & Expo 2012, Orlando, FL, USA, 19–23 May 2012.
- 20. Naderi, B.; Azab, A. Modeling and heuristics for scheduling of distributed job shops. *Exp. Syst. Appl.* **2014**, *41*, 7754–7763. [CrossRef]
- 21. Wang, F.; Tang, Q.H.; Rao, Y.Q.; Zhang, C.Y.; Zhang, L.P. Efficient Estimation of Distribution for Flexible Hybrid Flow Shop Scheduling. *Acta. Autom. Sin.* 2017, 43, 280–292.
- 22. Meng, L.; Zhang, C.; Ren, Y.; Zhang, B.; Lv, C. Mixed-integer linear programming and constraint programming formulations for solving distributed flexible job shop scheduling problem. *Comput. Ind. Eng.* **2020**, *142*, 106347. [CrossRef]
- 23. Pinedo, M.L. Scheduling: Theory, Algorithms, and Systems, 3rd ed.; Springer: New York, NY, USA, 2008.