HEC MONTREAL

Improved formulations for a printing planning problem using symmetry breaking constraints

By

Zhao Li

A Thesis Submitted
In Partial Fulfillment of Requirements
for a Master of Science in Global Supply Chain Management

August, 2017

Abstract

This paper analyses a common planning problem in the printing industry. In this problem, the planner first allocates designs to different slots of the printing plates. Then the number of plates and the number of rotations for each plate need to be carefully decided to meet customer demands and reduce costs. A general optimization formulation has been introduced for this problem (Baumann et al. 2014^{a}, 2014^{b}). However, the issue of symmetry is significant in these formulations, due to identical printing plates, as well as identical slots on an individual plate. This symmetry causes the existence of alternative (optimal or not) solutions which generally increases the search process and the CPU time required to solve the problem. The aim of this thesis is to further study the printing planning problem and improve these formulations by adding symmetry breaking constraints, which can decrease the number of equivalent solutions and thus the computation time. We use optimisation programming language (OPL) to formulate the models, and the CPLEX solver to solve them and compare CPU time. We analysed the impact of various parameters on the total cost in the printing planning problem through a sensitivity analysis. Furthermore, we tested several types of symmetry breaking constraints, and although they are not all equally effective, some of them substantially decrease the CPU time. Such an innovation enhances the efficient utilisation of production equipment, and thus allows a company to increase its competitive advantage.

Table of Contents

List of tables

List of figures

Chapter 1: Introduction

1.1 Context

Rapid changes in economic development in the twenty-first century have created fiercely competitive markets in which manufacturing companies increasingly seek to rationalize their production planning to survive. Optimal use of production machinery, labour power, and materials can provide more options to customers, faster response to customer orders, and better assure quality of service. Furthermore, using production equipment efficiently is one of the main ways to reduce costs and maximize profit. A good production planning system is hence very important for manufacturing companies. In this thesis, we specifically consider the production planning for a firm in the custom design printing industry. Thus, this thesis aims to help companies in the custom design printing industry employ their production facilities more efficiently.

1.2 Problem

The problem discussed in this thesis is a common one in the custom design printing industry. The printing company being examined produces napkin pouches for different customers, a process which involves several steps. The first step is printing the inked image to the napkin using a rotating metal printing plate. For the second step, the printed napkin is folded and glued into napkin pouches; however, this thesis only considers the first step. The metal printing plates are made using a special process so that only specific areas of a plate are covered by the ink. Because of this process, the cost of manufacturing custom printing plates is quite high, involving production costs, setup costs, and variable costs (Baumann et al. 2014^{a}). These expenses provide companies with the incentive to find a cost-effective production plan required for a printing order while still satisfying customer demands. The printing plates are a standard size and each printing plate has seven slots on which designs can be allocated. Because of the identical plates and identical slots, the symmetry issue of the printing-planning problem is significant. Furthermore, although the production planning of custom designed napkins pouches is complex, most of the planning work is still done manually using a spread-sheet, a method which is highly inaccurate and time-consuming. Additionally, one of the main drawbacks of the spread-sheet approach is its tendency to result in an inefficient utilisation of the production machinery.

1.3 Objective

In response to such real-world printing problems, Baumann et al. (2014^{a}, 2014^{b}) wrote two papers, presenting two mixed integer programming（MIP） models for efficient printing planning. In this thesis, we further study this specific planning problem, improving on these two MIP models by adding new symmetry breaking constraints to reduce the necessary CPU time. Furthermore, we analyse the value of flexibility in this specific problem.

1.4 Contributions of the thesis

This paper improves on the two MIP models presented in Baumann et al. $(2014^{a}, 2014^{b})$ with the aim of obtaining better results with respect to CPU times as well as the quality of the solutions found. We propose three new types of symmetry breaking constraints which provide better results than those presented in Baumann et al. (2014^{a}). We further study a special case of symmetry breaking constraints, i.e., fixing a variable. The computational results indicate that while fixing a variable can improve the model, the impact of this is limited when the model already has other symmetry breaking constraints. Furthermore, we propose new symmetry breaking constraints based on another criterion, i.e., the presence or the absence of a color on the plate and the presence or the absence of a design in a slot on plate. Finally, we analyse via a sensitivity analysis the impact of various parameters on the total cost in the printing planning problem.

1.5 Methodology

This thesis utilises an existing data set used by Baumann et al. (2014^{a}) in the article 'Planning of a make-to-order production process in the printing industry'. This data set is used in their computational experiments as a benchmark test set. We use optimisation programming language (OPL) to formulate the models, to which we add new symmetry-breaking constraints, using the state-of-the-art general-purpose commercial optimization software CPLEX to solve them. Next, we compare the results of the various formulations with the models proposed by Baumann et al. $(2014^a,\ 2014^b)$. Furthermore, the thesis also studies the value of different types of process flexibility in this planning problem proposed by Baumann et al. (2014^a).

1.6 Structure of the thesis

The remainder of this thesis is organized as follows: In Chapter 2, we discuss the issue of symmetry breaking in MIP formulations and review the related literature. In Chapter 3, we discuss the printing planning problem and review the literature on similar problems. Chapter 4 presents new symmetry breaking constraints. In Chapter 5, we report on the design and the results of the computational analysis. In Chapter 6, we analyse the value of flexibility in production planning problem. Finally, in Chapter 7, we formulate conclusions and directions for future research.

Chapter 2 Symmetry breaking

This chapter is organized as follows. Section 2.1 discusses the issue of symmetry in Mixed Integer Programming (MIP), while Section 2.2 reviews the literature on approaches used to deal with symmetry in MIP.

2.1 Issue of symmetry in mixed integer programming formulations

In the context of this paper, symmetry refers to objects which are identical and can be interchanged. This widespread phenomenon causes difficulties in MIP. When variables can be permuted, many alternative solutions can be created for a specific problem formulation. However, when we bring these solutions into reality, we find that they are essentially the equivalent solution because the structure of the solution is the same. When the symmetric solutions group is large, it is difficult to use traditional branch-and-bound algorithms to solve the MIP formulation since the search pool is huge and the required CPU time is too long. Fortunately, there are several ways of dealing with symmetries. Adding symmetry breaking constraints to the MIP formulation is one of the most efficient approaches that can eliminate symmetric solutions (Margot 2010).

2.1.1 Illustrated example using the bin packing problem

The issue of symmetry can be illustrated using the classical bin packing problem. Given the set $L_n = \{1, 2, 3, \ldots, n\}$ of n items, each item i has a size S_i. Each item must be allocated exactly to one bin with a capacity of C. We want to find the minimum number of bins needed so that all items are allocated. We assume that n bins are available.

If we use a linear program to describe this problem, then the bin packing problem is expressed as follows:

$$Min \quad \sum_{j=1}^{n} Y_j \qquad (1)$$

s.t.

$$\sum_{i=1}^{n} S_i X_{ij} \leq CY_j \qquad \forall j \in \{1, \dots, n\} \quad (2)$$

$$\sum_{j=1}^{n} X_{ij} = 1 \qquad \forall i \in \{1, \dots, n\} \quad (3)$$

$$X_{ij} \in \{0,1\} \qquad \forall i \in \{1, \dots, n\}, \forall j \in \{1, \dots, n\} \quad (4)$$

$$Y_j \in \{0,1\} \qquad \forall j \in \{1, \dots, n\} \quad (5)$$

The decision variables have the following definition:

Y_j=1, if the bin j is used; =0, otherwise;

X_{ij} =1, if we put item i into bin j; =0 otherwise;

The objective function (1) is to minimize the number of bins used. Constraint (2) imposes that the total of the items' sizes in each bin should not exceed the capacity of a used bin. Constraint (3) imposes that each item must be put in exactly one bin. Constraints (4) and (5) impose the binary conditions.

We illustrate this with an example. As we can see from the Table 1 below, there are 5 items. *L*= {1,2,3,4,5} and the corresponding volume size is 0.6, 0.4, 0.4, 0.8, 0.9. The bin capacity is 1.

Table 1: Bin packing problem example data

item	**1**	**2**	**3**	**4**	**5**
volume	0.6	0.4	0.4	0.8	0.9

The following Figure 1 presents one of the feasible solutions. We can see from it that we can put each item in different bins because the item volume satisfies the capacity constraint. Then we use 5 bins.

Figure 1: A feasible solution of the bin packing problem

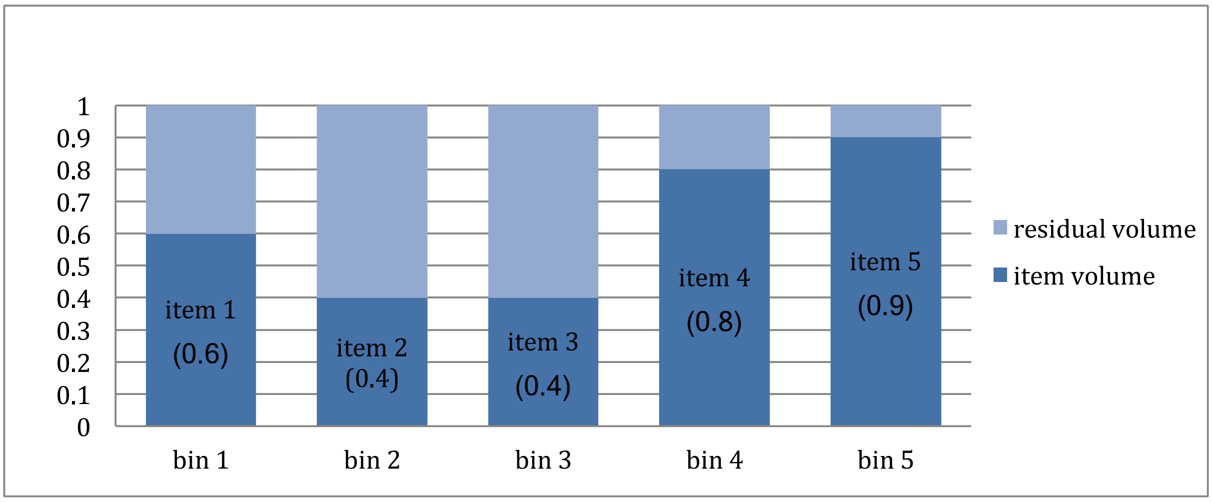
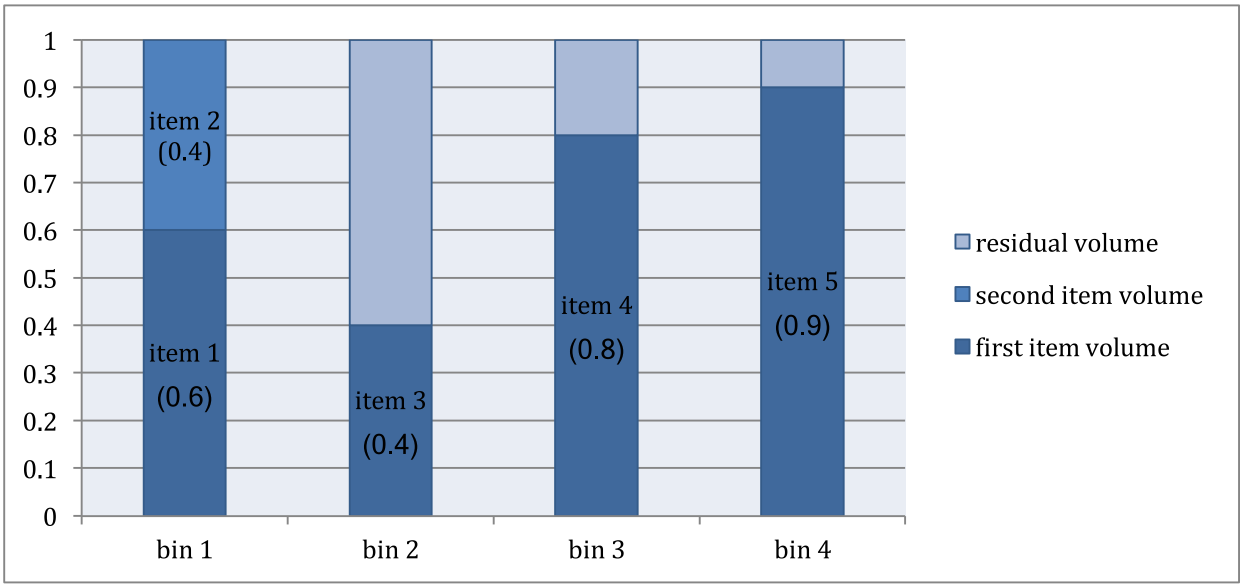


Figure 2 presents one of the optimal solutions for minimizing the number of bins. We can see from it that we can allocate item 1 and 2 to bin 1, item 3 to bin 2, item 4 to bin 3, and item 5 to bin 4, producing an optimal solution that uses only 4 bins in total.

Figure 2: An optimal solution of the bin packing problem

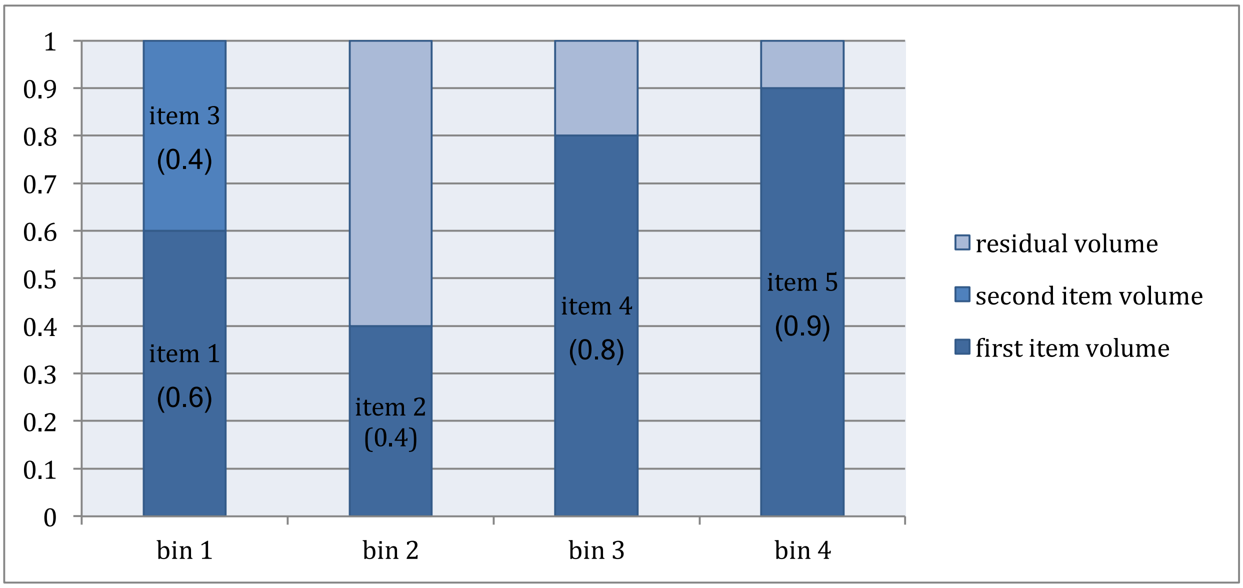


However, there are also other optimal solutions. As we can see from Figure 3, we can allocate item 3 to bin 1, item 1 and 2 to bin 2, item 4 to bin 3, and item 5 to bin 4. Compared with the original optimal solution, we just interchange the items in the bins 1 and 2. This illustrates symmetry among bins since they all have the same capacity.

Figure 3: An alternative optimal solution obtained by interchanging bins 1 and 2

Furthermore, as we can see from Figure 4, we can allocate item 1 and 3 to bin 1, item 2 to bin 2, item 4 to bin 3, and item 5 to bin 4. Compared with the original optimal solution, we just interchange the position of items 2 and 3, since the size of item 2 and item 3 are same.

Figure 4: An alternative optimal solution obtained by interchanging items 2 and 3



This example shows that alternative optimal solutions that are equivalent to the original optimal solution can be obtained due to symmetry. With respect to the formulation, they are in fact different solutions because the optimal values of the variables are different. However, when we bring these solutions into reality, we find that the solutions are equivalent. When the symmetric solutions group is large, it is difficult to use traditional branch-and-bound algorithms to solve the MIP formulation, because the algorithm needs more time to review all the alternative solutions, typically leading to long CPU times.

2.2 Literature review: Approaches used to deal with symmetry in MIP

The Mathematical Programming community has developed several approaches to deal with symmetry in MIP formulations. This section briefly discusses these approaches, which can be broadly subdivided into 3 categories: *1) adding static symmetry breaking inequalities to the original formulation, 2) problem reformulation, and 3) dynamic symmetry breaking inequalities during the search process.* Margot (2010) provides an excellent review on several approaches for dealing with symmetry problems.

2.2.1 Adding static symmetry breaking constraints to the original formulation

Adding static symmetry breaking constraints is one of the most useful techniques for solving symmetric MIP problems. This approach has been explored for many specific problems. Sherali and Smith (2001) identify the symmetry difficulty in three applications: a telecommunications network design problem, a noise dosage problem, and a machine procurement and operation problem. In the noise dosage problem, the aim is to minimize the dosage of noise workers are exposed to when they operate machines. Because workers are identical, alternative solutions can be obtained by reallocating the workers to the different machines. To eliminate this symmetry, the authors add lexicographic ordering constraints related to allocating workers to machines. Denton et al. (2010) applied this approach in the problem of allocating surgeries to operating rooms. In this problem, the operating rooms are the same, which leads to symmetry issues. Denton et al. (2010) add lexicographic constraints to mitigate the effect of indistinguishable operating rooms, which is similar to the solution used by Sherali and Smith (2001). The lexicographic order constraints require that surgeries be assigned to operating rooms in lexicographic order. Sherali et al. (2013) propose a mixed integer programming model for an integrated airline operational planning process that involves schedule design, fleet assignment, and aircraft routing. In this model, symmetry arises because aircraft of the same type are interchangeable. Sherali et al. (2013) use suitable hierarchical symmetry breaking constraints to differentiate identical aircraft. Degraeve et al. (2002) apply symmetry breaking constraints to eliminate symmetry in a fashion layout and cutting problem. Jans (2009) compares the effect of eight different symmetry-breaking constraints to lot-sizing problems on parallel identical machines.

Fixing a variable is a special case of adding static symmetry breaking constraints. We use the above-mentioned bin packing problem to illustrate this. In the bin packing problem, we need to allocate items to bins while minimizing the number of bins. When we think about a solution, we know we use at least one bin, and that we need to put item one into one bin. Therefore, we can always put item one into bin one without losing optimality. In this mathematical modeling, we can fix the binary variable X_{11} to one, which means we always put item 1 in bin 1. Normally, this approach can reduce the search space and reduce the number of alternative solutions. As we can see from the literature, this approach has been successfully applied to many problems. Degraeve and Jans (2002) applied this approach to a fashion layout problem to reduce the computational time, later also successfully used the technique in a lottery problem (Jans and Degraeve 2008). Margot (2003) demonstrates how such an approach can easily be combined with other techniques for partly eliminating symmetry and reducing the overall solution time.

Adding static constraints to the initial formulation is a simple and convenient solution to symmetry issues. It is a more flexible method compared with other solutions, and can be applied in a standard solver without interfering with the internal solution algorithm. Users do not need to learn specific computer science algorithms, and it is easy to use this approach. However, this approach also has some limitations. These constraints are specific to the problem, and authors need to identify the specific symmetry of the problem by themselves. Furthermore, it is difficult to estimate the impact of static symmetry breaking constraints without empirical evaluation, as many factors affect this approach, such as the structure of the problem, the choice of the algorithm, the input order, etc.

2.2.2 Problem reformulation

Problem reformulation is an important approach to deal with symmetry issues. By defining new specific decision variables, problem reformulation can avoid part of the inherent symmetry. This technique has been successfully applied to some specific problems, such as the printing planning problem (Baumann and Trautmann 2014[b]), the cutting stock problem (Ben Amor et al. 2005), and the fashion layout and cutting problem (Degraeve and Jans 2002). This approach is more complicated and less flexible than adding static constraints to the initial formulation, because of the need to recreate a new formulation for each specific problem. However, the general reformulation approach can usually improve the lower bound and reduce the solution time, as indicated by several papers such as Barnhart et al. (1998) and Teghem et al. (1995). Dantzig-Wolfe decomposition (Barnhart et al. 1998) is a special case of reformulation, which reduces the symmetry issue for some specific problems.

2.2.3 Dynamically breaking symmetry during the search process

For this approach, instead of improving the original formulation, the algorithm deals with the symmetry during the search process. During the branch-and-bound process, a filtering algorithm identifies whether a new node is symmetrically equivalent to a previously visited node during the search process. If these two nodes are equivalent, the algorithm stops exploring the node and the tree is pruned. Margot (2002) dynamically adds isomorphism inequalities to the branch-and-cut algorithms for pruning the enumeration tree, while Margot (2003) uses orbits of the symmetry group for pruning the enumeration tree and generating cuts. Both articles apply their respective techniques to three classical problems: covering designs, error correction codes, and hard covering problems. Margot (2007) generalizes the approach mentioned in the previous two papers to handle the symmetry issues in integer linear programming. Compared with the previous two approaches, this approach often requires a large number of constraints to be handled. Moreover, adding dynamic symmetry breaking constraints in the algorithm requires expert knowledge, which presents an obstacle for many users.

Chapter 3 The printing-planning problem

This chapter is organized as follows. Section 3.1 describes the printing planning problem in detail. Section 3.2 discusses the symmetry issues in this printing planning due to the presence of identical plates and slots. Section 3.3 presents the MIP formulation for the printing planning problem as proposed in Baumann et al. (2014^a), while Section 3.4 presents another MIP formulation for this problem used in Baumann et al. (2014^b). Finally, Section 3.5 reviews the literature on related problems.

3.1 The planning problem in the printing industry

This section describes the printing-planning problem in detail, which was previously analysed by Baumann et al. $(2014^a, 2014^b)$. The printing company being discussed produces napkin pouches. As mentioned in the introduction, each customer has his or her own requirement for the napkin pouches' design. Customers can choose any color and decide whether to put a white stripe at the edge of the pouches. This white stripe is referred to as a white border. Therefore, the requirements for napkin pouches include color and white borders. Each week, customers send orders with different requirements to the printing company, which uses a "make-to-order" strategy to fulfill these demands. For the remainder of the thesis, we refer to this type of demand as "customer-specific design demand". This thesis focuses mainly on this type of demand, but there is another type, which we call standard design demand. Standard designs have different colors and always have a white border. For the company, this kind of demand is easy to plan for and satisfy. The company can use a "make-to-stock" strategy to satisfy standard design demand. In this problem, it is assumed that the demand for standard designs is 0 and some customer specific designs can be produced together with the standard design.

This application involves several plates, each with seven slots to which designs must be assigned. Since all the designs (customer specific designs and standard designs) are the same size, we can at most allocate seven different designs to the seven slots of a single plate. Production machinery imposes some technological constraints on the printing planning problem. Each slot must be filled with a design; a slot cannot be left empty. Due to the color constraint, each plate can contain at most two different colors. When more than two colors are allocated to a plate, it is not feasible for the printing machine to run. Furthermore, there is a small gap between plates due to some technical constraints. When the production machine is running, a white stripe shows up at the edge of paper. Because of this white border constraint, each plate needs to have at least two customer specific designs with white border; otherwise it needs to add a standard design to avoid the white border. At most one standard design can be allocated to a plate. There are also some organizational constraints which need to be met. In reality, the packing space is limited. When we allocate a customer specific design to two or more plates, packing would require more space, which increases both the costs and the labour power required. Therefore, in this problem, a customer specific design cannot be allocated to more than one plate.

In the printing planning problem, the planner first allocates designs to the various slots of the plates. A design can be allocated to more than one slot on the same plate. Then the number of plates, as well as the number of rotations (i.e., the number of impressions using the plate for all designs allocated to that specific plate) is decided upon to best meet customer demands. We must satisfy this demand in a way that makes it possible to over-produce but impossible to under-produce. Our objective function is to minimize the total cost, which consists of two parts: set up costs and overproduction costs. Set up costs are incurred whenever we use a new plate because the new plate needs to be produced, cleaned and installed. Overproduction costs is incurred when the production quantity exceeds customer demand. The number of units produced of one specific design is equal to the number of slots used for this specific design multiplied by the corresponding number of rotations of the plate. For the customer-specific designs, overproduction happens if more units are produced than the demand. For the standard designs, any production is counted as overproduction, but the associated cost is lower than for the overproduction of customer-specific designs.

Illustrative example: We use an example from Baumann et al. (2014[a]) to better illustrate the printing planning problem. As can be seen from Table 2, in one week there is an order for three customer-specific designs: D1, D2, and D3. The company also has three standard designs that can be produced. For each design, we have information on whether it is a standard design or not, whether it requires a white border or not, the color, the demand and the overproduction cost. The setup cost per plate is $540, while the overproduction cost per unit of customer-specific designs is $0.0035 and the overproduction cost per unit of standard design is $0.001. These costs are the same as the existing data set used in Baumann et al. (2014[a]).

Table 2: A customer order for illustrative example

Designs	Standard design	White border	Color	Demand (units)	Overproduction cost per unit ($)
D1	No	No	Blue	30,000	0.0035
D2	No	No	Pink	25,000	0.0035
D3	No	Yes	Green	40,000	0.0035
D4	Yes	No	Blue	0	0.001
D5	Yes	No	Pink	0	0.001
D6	Yes	No	Green	0	0.001

An optimal solution to this example, which has already been solved by the model presented in Baumann et al. (2014^{a}), is given in Figure 5. Each plate contains 7 slots. Because each plate can only contain at most two different colors, we need at least two plates (plate A and plate B). We can then decide which design is allocated to which plate. We allocate designs D1 and D3 to plate A, and D2 to plate B. Due to the white border constraint, we also need to add a standard design to plate B.

Figure 5: An optimal solution to the illustrative example

Plate A	Plate B
D1-slot 1	D2-slot 1
D1-slot 2	D2-slot 2
D1-slot 3	D2-slot 3
D3-slot 4	D2-slot 4
D3-slot 5	D2-slot 5
D3-slot 6	D2-slot 6
D3-slot 7	D4-slot 7
Rotation: 10,000 times	Rotation: 4,167 times

In plate A, D1 is allocated to slots 1, 2, and 3, while D3 is allocated to slots 4, 5, 6 and 7. For plate B, D2 is allocated from slot 1 to slot 6. A standard design of D4 is allocated to slot 7.

The number of units printed of one specific design is equal to the number of slots used for this specific design multiplied by the corresponding number of rotations of the plate. Plate A needs to rotate 10,000 times. Total production for D1 is 30,000 units (=3*10,000), which satisfies customer demand exactly. Total production for D3 is 40,000 units (= 4*10,000), which also exactly satisfies the customer demand. Plate B needs to rotate 4,167 times. Therefore, total production for D2 is 25,002 units (=6*4,167). The overproduction cost for the customer specific design D2 is $0.007(=0.0035$*(25,002-25,000)). Total production for the standard design D4 is 4,167 units (= 1*4,167), with an overproduction cost of $4.167 (=4167*$0.001). The setup cost is \$1,080, since there are two plates used at a cost of \$540 each. Total cost is the setup cost (\$1,080) plus the overproduction cost (\$0.007+\$4.167), which is equal to \$1,084.174.

3.2 Symmetry in the printing-planning problem

Because of the identical slots and plates involved in printing, the printing-planning problem has many alternative (optimal) solutions, which all have the same structure. When we bring these solutions into reality, we find the solutions are equivalent. This symmetry can be explained using the above illustrative example. Figure 5 presents one of the optimal solutions from the example, in which designs D1 and D3 are allocated to plate A, and design D2 and the standard design D4 are allocated to plate B. But there are also other optimal solutions to this problem.

3.2.1 Symmetry due to identical plates

One of the alternative optimal solutions is given in Figure 6. We can see from Figure 6 that plates A and B are interchanged in this solution. This alternative optimal solution is therefore a result of the symmetry of the plates.

Figure 6: Symmetry due to identical plates

Plate A	Plate B
D2-slot 1	D1-slot 1
D2-slot 2	D1-slot 2
D2-slot 3	D1-slot 3
D2-slot 4	D3-slot 4
D2-slot 5	D3-slot 5
D2-slot 6	D3-slot 6
D4-slot 7	D3-slot 7

3.2.2 Symmetry due to identical slots

Two other alternative optimal solutions are given in Figures 7 and 8. We can see from these that the slot positions within a plate can be interchanged due to the symmetry of the slots. In Figure 7, D3 is allocated to slots 1, 2, 3 and 4; D1 is allocated to slots 5, 6 and 7; the standard design D4 is allocated to slot 1; and D2 is allocated to the rest of the slots. In Figure 8, D3 is allocated to slots 1, 3, 5 and 7; D1 is allocated to slots 2, 4 and 6; D4 is allocated to slot 1; and D2 to the rest of the slots.

Figure 7: Symmetry due to identical slots: one alternative optimal solution

D3-slot 1	D4-slot 1
D3-slot 2	D2-slot 2
D3-slot 3	D2-slot 3
D3-slot 4	D2-slot 4
D1-slot 5	D2-slot 5
D1-slot 6	D2-slot 6
D1-slot 7	D2-slot 7
Plate A	Plate B

Figure 8: Symmetry due to identical slots: another alternative optimal solution

D3-slot 1	D4-slot 1
D1-slot 2	D2-slot 2
D3-slot 3	D2-slot 3
D1-slot 4	D2-slot 4
D3-slot 5	D2-slot 5
D1-slot 6	D2-slot 6
D3-slot 7	D2-slot 7
Plate A	Plate B

Each optimal solution can be transformed into several alternative optimal solutions. However, because each alternative solution is associated to a different set of values of the decision variables (using the formulation proposed by Baumann et al. (2014[a])), these alternative solutions represents distinct solutions and need to be explored in a branch-and-bound process. This thesis attempts to use symmetry-breaking constraints to eliminate these symmetry issues and thus reduce the number of equivalent solutions, reducing in turn the required CPU time.

In this application, we assume there is only one machine used to produce all the products. However, in reality printing manufacturers may have several identical machines, so there is also a symmetry issue due to identical machinery. This particular kind of symmetry is not considered in this thesis.

3.3 MIP formulation in Baumann et al. (2014^a)

As described in the previous section, Baumann et al. (2014^a) present a MIP formulation for the printing planning problem. We first introduce the notation they used for this formulation.

We define the sets and parameters:

$C = \{1, \ldots, |C|\}$ Set of colors

$I = \{1, \ldots, |I|\}$ Set of designs

$I^o \subset I$ Set of customer-specific designs

$I^s \subset I$ Set of standard designs

$I_c \subseteq I$ Set of designs with color c

$I_w \subseteq I$ Set of designs with white border

$J = \{1, \ldots, |J|\}$ Set of slots

$P = \{1, \ldots, |P|\}$ Set of plates

$\bar{c}$ Maximum number of different colors per plate

c_i^o Overproduction cost per unit of customer-specific design *i*

c_i^s Overproduction cost per unit of standard design *i*

c^p Setup cost for a plate

d_i Demand for design *i*

M Sufficient large number, = $max_{i \in I} d_i$

Next, we define the decision variables:

u_{ijp} Units of design *i* produced in slot *j* of plate *p*

v_i Overproduced units of design *i*

W_p =1, if plate *p* is used; 0, otherwise

X_{ijp} =1, if design *i* is allocated to the slot *j* of plate *p*; 0, otherwise

Z_{cp} =1, if plate *p* contains color *c*; 0, otherwise

Y_{ip} =1, if design *i* is allocated to plate *p*; 0, otherwise

The objective function is to minimize the overproduction and setup costs.

$$Min \sum_{i \in I^o} c_i^o v_i + \sum_{i \in I^s} c_i^s v_i + \sum_{p \in P} c^p W_p$$

Constraint (1) calculates the number of overproduced units of a particular design.

s.t.

$$v_i = \sum_{j \in J; p \in P} u_{ijp} - d_i \qquad \forall i \in I \quad (1)$$

Constraint (2) assures that if a plate is used, then each slot on that plate must be used.

$$\sum_{i \in I} X_{ijp} = W_p \qquad \forall j \in J; \forall p \in P \quad (2)$$

Constraint (3) ensures that a specific design can only be produced in the slot of a plate when it is allocated to this slot.

$$u_{ijp} \leq M X_{ijp} \qquad \forall i \in I; \forall j \in J; \forall p \in P \quad (3)$$

Constraint (4) guarantees that each slot on the plate should produce the same number of units.

$$\sum_{i \in I} u_{ijp} = \sum_{i \in I} u_{i,j-1,p} \qquad \forall j \in J: j > 1; \forall p \in P \quad (4)$$

Constraint (5) and constraint (6) limit the maximum number of colors each plate can contain.

$$\sum_{c \in C} Z_{cp} \leq \overline{c} \qquad \forall p \in P \quad (5)$$

$$|J| Z_{cp} \geq \sum_{i \in I_c; j \in J} X_{ijp} \qquad \forall c \in C; \forall p \in P \quad (6)$$

Constraint (7) imposes a white border constraint. This requires that either two customer-specific designs with a white border or a standard design must be allocated to each plate.

$$W_p \leq \sum_{i \in I_{w}; j \in J} \frac{1}{2} X_{ijp} + \sum_{i \in I^S; j \in J} X_{ijp} \qquad \forall p \in P \quad (7)$$

Constraint (8) ensures that each plate contains at most one slot with a standard design.

$$\sum_{i \in I^S; j \in J} X_{ijp} \leq 1 \qquad \forall p \in P \quad (8)$$

Constraint (9) and constraint (10) impose the no-split constraint, i.e., each customer-specific design must be allocated to a single plate.

$$\sum_{p \in P} Y_{ip} = 1 \qquad \forall i \in I^o \quad (9)$$

$$|J|Y_{ip} \geq \sum_{j \in J} X_{ijp} \qquad \forall i \in I^o; \forall p \in P \quad (10)$$

Integrality constraints:

$$u_{ijp} \geq 0 \qquad \forall i \in I;\ \forall j \in J; \forall p \in P$$
$$v_i \geq 0 \qquad \forall i \in I$$
$$W_p \in \{0,1\} \qquad \forall p \in P$$
$$X_{ijp} \in \{0,1\} \qquad \forall i \in I; \forall j \in J; \forall p \in P$$
$$Y_{ip} \in \{0,1\} \qquad \forall i \in I; \forall p \in P$$
$$Z_{cp} \in \{0,1\} \qquad \forall c \in C; \forall p \in P$$

Hereafter, we refer to this formulation as the basic formulation.

3.4 MIP in Baumann et al. $(2014^{\mathbf{b}})$

In the same year, Baumann et al. (2014^b) presented another MIP formulation for the printing planning problem, which we refer to as the alternative formulation. This formulation uses some new variables to avoid part of the inherent symmetry of the printing planning problem.

The new variables are:

r_p Number of rotations of plate *p*

q_{inp} Units of design *i* produced in *n* slots of plate *p*

K_{inp} =1, if design *i* is allocated to *n* slots of plate *p*; 0, otherwise

The alternative formulation has the same objective function as the basic formulation.

$$Min \sum_{i \in I^o} c_i^o v_i + \sum_{i \in I^s} c_i^s v_i + \sum_{p \in P} c^p W_p$$

The constraints (11) - (13) confine the range of the number of units printed of each design. When K_{inp} equals 1, then constraint (11) and (12) ensure that q_{inp} is equal to the exact amount produced, i.e., n slots multiplied by the number of rotations. When K_{inp} equals zero, then (12) ensures that q_{inp} is also set to zero, and the right-hand side of (13) becomes a negative number.

s.t.

$$q_{inp} \leq nr_p \qquad \forall i \in I;\ n = 1, \dots, |J|; \forall p \in P \quad (11)$$

$$q_{inp} \leq nMK_{inp} \qquad \forall i \in I; n = 1, \dots, |J|; \forall p \in P \quad (12)$$

$$q_{inp} \geq nr_p - nM\left(1 - K_{inp}\right) \quad \forall i \in I; n = 1, \dots, |J|; \forall p \in P \quad (13)$$

Constraint (14) calculates the number of overproduced design units.

$$\sum_{n \in J;\ p \in P} q_{inp} = d_i + v_i \qquad \forall i \in I \quad (14)$$

Constraint (15) assures that if a plate is used, each slot on that plate must be used.

$$\sum_{i \in I; n \in J} nK_{inp} = |J|W_p \qquad \forall p \in P \quad (15)$$

Constraint (16) imposes a white border constraint. This requires that either two customer-specific designs with a white border or a standard design must be allocated to each plate.

$$\sum_{i \in I_{W}; n \in J} \frac{1}{2} nK_{inp} + \sum_{i \in I^S; n \in J} K_{inp} \geq W_p \qquad \forall p \in P \quad (16)$$

Constraint (17) ensures that each plate contains at most one slot with a standard design.

$$\sum_{i \in I^S; n \in J} nK_{inp} \leq 1 \qquad \forall p \in P \quad (17)$$

Constraint (18) and constraint (19) limit the maximum number of colors each plate can contain.

$$\sum_{c \in C} Z_{cp} \leq \overline{c} \qquad \forall p \in P \qquad (18)$$

$$\sum_{i \in I_c; n \in J} K_{inp} \leq |J| Z_{cp} \qquad \forall c \in C; \forall p \in P \qquad (19)$$

Constraint (20) imposes the no-split constraint, i.e., each customer-specific design must be allocated to a single plate.

$$\sum_{n \in J; p \in P} K_{inp} = 1 \qquad \forall i \in I^o \qquad (20)$$

Constraint (21) eliminates the symmetry between the used and unused plates. Plate *p* is not used before plate *p-1*.

$$W_{p-1} \geq W_p \qquad \forall p \in P : p > 1 \qquad (21)$$

Non-negativity and integrality constraints:

$$r_p \geq 0 \qquad \forall p \in P$$

$$q_{inp} \geq 0 \qquad \forall i \in I; n = 1, \dots, |J|; \forall p \in P$$

$$v_i \geq 0 \qquad \forall i \in I$$

$$W_p \in \{0,1\} \qquad \forall p \in P$$

$$K_{inp} \in \{0,1\} \qquad \forall i \in I; n = 1, \dots, |J|; \forall p \in P$$

$$Z_{cp} \in \{0,1\} \qquad \forall c \in C; \forall p \in P$$

3.5 Literature review on related problems

This section reviews some combinatorial or optimization problems similar to the printing planning problem, more specifically, the fashion layout and cutting problem and the cover printing problem. These problems often contain large symmetry groups, which are difficult for branch-and-bound algorithms to solve.

3.5.1 Fashion layout and cutting problem

The fashion layout and cutting problem was first presented by Degraeve and Vandebroek (1998). This problem occurred in the Belgian high fashion industry. For haute couture, the clothes are cut and sewed according to orders that have specific designs and sizes. Cutting the fabric is the most time-consuming and costly work involved in this industry. Fabric is first placed on the cutting table. The templates which we are also called stencils indicate the most economical way to place the parts of the article. Each stencil used has a corresponding size. The combined set of stencils, also called the pattern, is then fixed onto the fabric. Next, the tailor cuts the fabric according to the pattern. As with the printing planning problem, in which designs need to be allocated to specific slots on a plate, different types of stencils need to be allocated to the positions in a pattern. The number of layers of fabric in the fashion layout problem is similar to the number of rotations in the printing planning problem. However, the layer of fabric has a limitation due to the length of the knife and the thickness of the fabric. The aim of the fashion layout problem is analogous to the printing planning problem, which is to find the best patterns (combination of stencils) and decide on the number of layers of fabric for each pattern in order to satisfy customer demand while meeting equipment constraints. The objective function of the fashion layout problem is the same as the printing planning problem, which is to minimize the setup (number of pattern) and overproduction (fabric waste) costs. The printing planning problem is more difficult than the fashion layout problem, however, because the printing planning problem has more constraints, such as the color and white border constraints. Degraeve and Vandebroek (1998) used a non-liner programming model and a MIP model to solve this problem. Furthermore, they also tried the complete enumeration approach to solve the problem, and compared the results of these two approaches. The results indicated that a larger size of this problem was linked to better performance of the MIP approach. Based on the two models presented in Degraeve and Vandebroek (1998), Martens (2004) develops two genetic algorithms to solve the fashion layout problem, particularly in cases where the problem case has become large and complicated, such as a large order of jeans. Degraeve et al. (2002) present two alternative mixed linear programming formulations, which are superior to the models presented in Degraeve and Vandebroek (1998).

There are also some variants of the fashion layout problem considered by other authors. Rose and Shier (2007) simplify the assumption of the problem, using less layers of fabric and cutting tables to represent minimizing the setup cost. The overproduction cost can be avoided by satisfying the exact demand. This simplified assumption makes the exact enumeration approach more efficient. Compared with previous articles, Nascimento et al. (2010) add additional costs to the objective function, such as inventory costs, cutting costs, folding costs, and spreading costs. They propose several state-space based heuristics to solve this problem. Yang et al. (2011) use integer programming (IP) to find the optimal solution of a variant of the fashion layout problem, which incorporates the stack cost of a layer of cloth to be considered. They combined IP with ant colony optimization to obtain a heuristic solution.

3.5.2 Cover-printing problem

The book-cover printing problem has many similarities with the napkin printing problem, since both involve printing using plates. Both problems need to decide the number of required plates and the number of prints per plate. Book-cover designs should be assigned to the compartment of a plate which is similar to the napkin printing problem that napkin designs should be allocated to the slot of a plate. There are four compartments on the plate for printing book-covers, so there are less positions available than that for printing napkins. The objective of both problems is similar since the goal is to minimize the setup cost of the plate and the overproduction costs while still satisfying customer demands and constraints. The cover-printing problem is simpler than the printing planning problem, however, because covers lack the color and white border constraints. Teghem et al. (1995) present both a mixed integer non-linear programming model and a linear model dealing with this problem. They first attempt to use the linear model, combined with a simulated annealing algorithm, to solve this problem. In the following years, many authors proposed different approaches in an attempt to solve this problem. Elaoud et al. (2007) solve it in two ways. They first attempt to use single objective genetic algorithms to obtain a near-optimal solution. The results indicate that this approach is better than the simulated annealing algorithm. They then use multi-objective genetic algorithms to find a compromise solution between minimization of setup costs and minimization of overproduction costs. This approach proved to be more efficient than the simulated annealing algorithm and single objective genetic algorithm. Ekici et al. (2010) present a linear integer programming model, and by adding some simple cuts, are able to obtain a better lower bound, solving 14 out of 32 real world instances by two efficient heuristics approaches. Other approaches have also been proven to produce positive results for the cover-printing problem, such as a greedy random adaptive search procedure (Tuyttens and Vandaele 2010, 2014), and an ad-hoc heuristic approach (Romero and Alonso 2012).

Chapter 4 New symmetry breaking constraints

This chapter is organized as follows. Section 4.1 presents the new sets of symmetry breaking constraints for the basic formulation, while Section 4.2 presents the symmetry breaking constraints for the alternative formulation. In Section 4.3, we fix a variable and add it to the basic formulation. In Section 4.4, we impose a hierarchical order based on another criterion, i.e., the presence or absence of a color on the plate. Finally, in Section 4.5, we provide a clear summary of all the models we proposed.

4.1 Symmetry breaking constraints for the basic formulation

This section introduces the set of symmetry breaking constraints used in Baumann et al. (2014^{a}), as well as three new sets of symmetry breaking constraints, all of which can be applied in the basic formulation presented in Baumann et al. (2014^{a}).

4.1.1 Symmetry breaking constraints used in Baumann et al. $(\mathbf{2014^{a}})$

In Baumann et al. (2014^{a}), the authors propose to add three symmetry-breaking sets of constraints to their basic formulation. The first is set (22), which is used to eliminate the symmetry of slots within the plate. The index of the design allocated to slot *(j-1)* must be no larger than the index of the design allocated to slot *j*. The authors impose this hierarchical ordering by assigning a specific number to each slot. In set (22), this specific number is equal to the index of the design allocated to the specific slot. For a given plate, the designs are allocated to slots according to a non-decreasing order of the design index. Figure 9 depicts two equivalent plates, each with seven slots. The only difference between these plates is that the designs are in different slots. Plate 1 follows the order imposed by set (22). Plate 2, however, does not follow this order, and so is eliminated by (22).

$$\sum_{i \in I} i\, X_{i,j-1,p} \leq \sum_{i \in I} i\, X_{ijp} \qquad \forall j \in J : j > 1, \forall p \in P \quad (22)$$

Figure 9: The order imposed by constraint set (22)

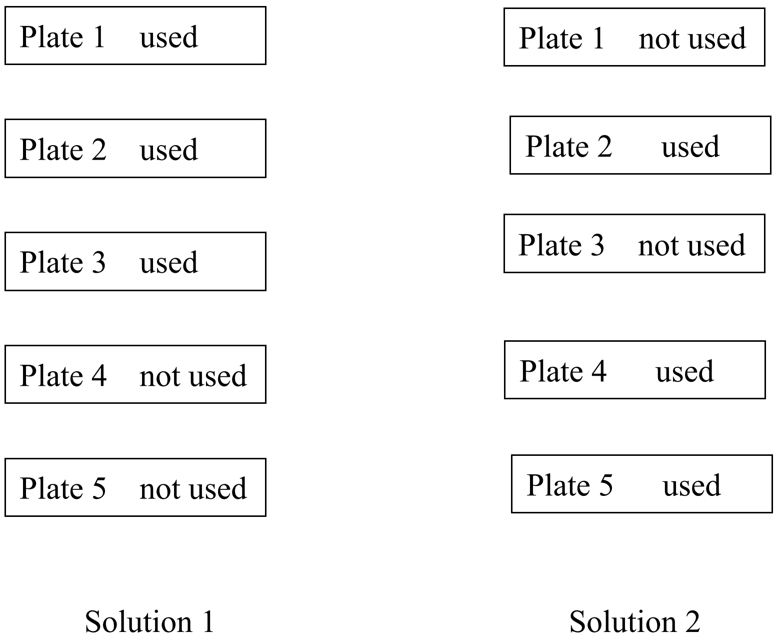
Plate 1	Value of $\sum_{i \in I} i\, X_{i,j,p}$	Plate 2	Value of $\sum_{i \in I} i\, X_{i,j,p}$
Design 1-slot 1	1	Design 1-slot 1	1
Design 1-slot 2	1	Design 2-slot 2	2
Design 2-slot 3	2	Design 1-slot 3	1
Design 3-slot 4	3	Design 4-slot 4	4
Design 4-slot 5	4	Design 4-slot 5	4
Design 4-slot 6	4	Design 7-slot 6	7
Design 7-slot 7	7	Design 3-slot 7	3

Plate 1 Plate 2

The second symmetry-breaking constraint is set (23), which is responsible for distinguishing which plates should be used and which should not. The specific ordering we impose is that plate *p* is not used before plate *p-1*. Figure 10 presents two solutions, each using only 3 out of 5 plates. Solution 1 uses the plates according to the order imposed by (23). Solution 2 does not follow this order, and so is eliminated.

$$W_{p-1} \geq W_p \qquad \forall p \in P: p > 1 \quad (23)$$

Figure 10: The order imposed by constraint set (23)



There is, however, still a symmetry issue between the used plates. Constraint set (24) imposes a hierarchical ordering based on the index of the design allocated to the first slot in each plate, resulting in a non-increasing ordering of the plates. Figure 11 presents two solutions with 3 plates each. The solutions are equivalent and differ only by the ordering of the plates. Solution 1 follows the order imposed by constraint set (24). Since solution 2 fails to follow this order, it is eliminated.

$$\sum_{i \in I} i\, X_{i,j,p-1} \geq \sum_{i \in I} i\, X_{ijp} \qquad j = 1, \forall p \in P: p > 1 \quad (24)$$

Constraints (22) and (24) follow a similar logic, i.e., use the index of an allocated design to impose some ordering. For this thesis, constraints following this type of logic (i.e. constraints (22), (24)) are denoted as symmetry-breaking constraint 0 ($\mathbf{SBC0}$), while the basic formulation augmented with these constraints is referred to as $\mathrm{SBC0}^{a}$.

Figure 11: The order imposed by constraint set (24)

Value of $\sum_{i \in I} i X_{i1p}$

Solution 1:

Plate 1

Design 7 Slot 1	Design 7 Slot 2	Design 7 Slot 3	Design 8 Slot 4	Design 8 Slot 5	Design 9 Slot 6	Design 9 Slot 7	7

Plate 2

Design 4 Slot 1	Design 4 Slot 2	Design 5 Slot 3	Design 5 Slot 4	Design 6 Slot 5	Design 6 Slot 6	Design 6 Slot 7	4

Plate 3

Design 1 Slot 1	Design 2 Slot 2	Design 2 Slot 3	Design 2 Slot 4	Design 2 Slot 5	Design 2 Slot 6	Design 3 Slot 7	1

Solution 2:

Plate 1

Design 4 Slot 1	Design 4 Slot 2	Design 5 Slot 3	Design 5 Slot 4	Design 6 Slot 5	Design 6 Slot 6	Design 6 Slot 7	4

Plate2

Design 1 Slot 1	Design 2 Slot 2	Design 2 Slot 3	Design 2 Slot 4	Design 2 Slot 5	Design 2 Slot 6	Design 3 Slot 7	1

Plate 3

Design 7 Slot 1	Design 7 Slot 2	Design 7 Slot 3	Design 8 Slot 4	Design 8 Slot 5	Design 9 Slot 6	Design 9 Slot 7	7

4.1.2 First new set of symmetry-breaking constraints

Two papers (Coelho and Laporte (2014) and Albareda-Sambola et al. (2011)) inspired the first new type of symmetry breaking constraint. Coelho and Laporte (2014) studied the inventory-routing problem, which combines inventory control and vehicle routing problems. They present a formulation to minimize transportation and inventory-holding costs. In solving this problem, they also encountered the symmetry issue. The vehicles used to transport inventory are homogeneous, since the capacity of each vehicle is the same. Although interchanging the vehicles produces alternative feasible solutions, we know these solutions are equivalent to each other. Coelho and Laporte (2014) add the following set of constraints to solve this symmetry issue:

$$y_i^{kt} \leq \sum_{j<i} y_j^{k-1,t} \qquad \forall\, i \in V, \forall k \in K\backslash\{1\}, \forall t \in T$$

These constraints use the following sets and decision variables. Let V be the set of customers, K the set of identical vehicles, and T the set of time periods. The binary variable y_i^{kt} is equal to 1 when customer i is assigned to vehicle k in period t, otherwise y_i^{kt} is equal to 0. According to this symmetry breaking constraint, if customer i is assigned to vehicle k in period t (i.e. y_i^{kt}=1), then for the same period, some customer with an index smaller than i should be allocated to the previous vehicle. We adopt the logic of this symmetry breaking constraint for our printing planning problem. The following constraints are identified as symmetry breaking constraints (25), which reduces the symmetry of slots within the plate. According to (25), if design i is assigned to slot j of plate p, then a design with an index equal to or less than i should be assigned to slot *j-1* of plate *p.* In this way, the designs are assigned to the slots in a non-decreasing order of the design index for any given plate. Figure 12 depicts two equivalent plates with seven slots that only differ by having the designs in different slots of each plate. Plate 1 follows the order imposed by (25). Since plate 2 fails to follow this order, it is eliminated.

$$X_{ijp} \leq \sum_{l \leq i} X_{l,j-1,p} \qquad \forall i \in I, \forall j \in J : j > 1, \forall p \in P \quad (25)$$

Figure 12: The order imposed by constraint set (25)

Design 1-slot 1
Design 1-slot 2
Design 2-slot 3
Design 3-slot 4
Design 4-slot 5
Design 4-slot 6
Design 7-slot 7

Plate 1

Design 4-slot 1
Design 1-slot 2
Design 7-slot 3
Design 1-slot 4
Design 4-slot 5
Design 2-slot 6
Design 3-slot 7

Plate 2

Constraints (23) are still kept to eliminate the symmetry between the plates that are used and those that are not used.

$$W_{p-1} \geq W_p \qquad \forall p \in P: p > 1 \quad (23)$$

The following set is the symmetry breaking constraints (26), which eliminates the symmetry between the used plates according to the design index allocated to the first slot of the plate. Such a constraint requires that if design *i* is assigned to the first slot of plate *p,* then a design with an index that is equal to or smaller than *i* should be assigned to the first slot of plate *p-1*. This results in plates ordered in a non-decreasing way. Figure 13 presents two solutions with 3 plates each. The solutions are equivalent and differ only in their ordering of the plates. While solution 1 follows the order imposed by (26), solution 2 does not, and as such is eliminated.

$$X_{ijp} \leq \sum_{l \leq i} X_{l,j,p-1} \qquad \forall i \in I, j = 1, \forall p \in P: p > 1 \quad (26)$$

Constraints (25) and (26) follow a similar logic, and are hereafter referred to as symmetry-breaking constraint 1 (**SBC1**), with the basic formulation augmented with constraints (25), (23) and (26) referred to as $\mathrm{SBC1}^{a}$.

Figure 13: The order imposed by constraint set (26)

Solution 1:

Plate 1

Design 1 Slot 1	Design 2 Slot 2	Design 2 Slot 3	Design 2 Slot 4	Design 2 Slot 5	Design 2 Slot 6	Design 3 Slot 7

Plate 2

Design 4 Slot 1	Design 4 Slot 2	Design 5 Slot 3	Design 5 Slot 4	Design 6 Slot 5	Design 6 Slot 6	Design 6 Slot 7

Plate 3

Design 7 Slot 1	Design 7 Slot 2	Design 7 Slot 3	Design 8 Slot 4	Design 8 Slot 5	Design 9 Slot 6	Design 9 Slot 7

Solution 2:

Plate 1

Design 7 Slot 1	Design 7 Slot 2	Design 7 Slot 3	Design 8 Slot 4	Design 8 Slot 5	Design 9 Slot 6	Design 9 Slot 7

Plate2

Design 1 Slot 1	Design 2 Slot 2	Design 2 Slot 3	Design 2 Slot 4	Design 2 Slot 5	Design 2 Slot 6	Design 3 Slot 7

Plate 3

Design 4 Slot 1	Design 4 Slot 2	Design 5 Slot 3	Design 5 Slot 4	Design 6 Slot 5	Design 6 Slot 6	Design 6 Slot 7

4.1.3 Second new set of symmetry-breaking constraints

The second new type of symmetry breaking constraint is inspired by Degraeve et al. (2002) in which they study a fashion layout problem. Several layers of cloth are placed on a cutting table, with stencils of different sizes fixed on the cloth. The authors of the study created a formulation to find the best combination of stencils (called a pattern), and to determine the height of the stack of cloth that minimizes setup and overproduction costs for a given demand. Symmetry is a significant issue in this problem. In one pattern, the positions in a pattern can be interchanged, while the patterns themselves can also be interchanged because the cutting tables are identical. To overcome this symmetry issue, Degraeve et al. (2002) present the following symmetry-breaking constraints, based on the logic that if a size which is at most size i is allocated to the position p of pattern j, then a size which is greater than i cannot be allocated to position $p-1$ of pattern j. Let P be the set of different sizes, T be the set of different positions in a pattern, and R_s be the set of patterns. The binary variable a_{wpj} is equal to 1 when the size w is in position p in pattern j, otherwise a_{wpj} is equal to 0. The symmetry breaking constraints are then as follow:

$$\sum_{w=1}^{i} a_{wpj} + \sum_{w=i+1}^{|P|} a_{w,p-1,j} \leq 1 \qquad \forall i \in P, \forall p \in T \backslash \{1\}, \forall j \in R_s$$

This logic can also be applied to our problem. The following set aims to differentiate the slots on a plate. Constraints (27) specify that if a design with an index of at most i is allocated to slot j of plate p, then a design with an index greater than i cannot be allocated to the previous slot on the same plate. In this way, for any given plate, the designs are assigned to the slots in a non-decreasing order of the design index. Figure 14 depicts two equivalent plates with seven slots each, that differ only by having the designs in different slots of the plates. Plate 1 follows the order imposed by (27), while plate 2 does not, resulting in its elimination.

$$\sum_{l=1}^{i} X_{ljp} + \sum_{l=i+1}^{|I|} X_{l,j-1,p} \leq 1 \qquad \forall i \in I, \forall j \in J: j > 1, \forall p \in P \quad (27)$$

Figure 14: The order imposed by constraint set (27)

Design 1-slot 1	Design 4-slot 1
Design 1-slot 2	Design 1-slot 2
Design 2-slot 3	Design 7-slot 3
Design 3-slot 4	Design 1-slot 4
Design 4-slot 5	Design 4-slot 5
Design 4-slot 6	Design 2-slot 6
Design 7-slot 7	Design 3-slot 7
Plate 1	Plate 2

Constraint set (23) is still used to order the plates and determine which are used and which are not.

$$\mathrm{W}_{\mathrm{p}-1} \geq \mathrm{W}_{\mathrm{p}} \qquad \forall \mathrm{p} \in \mathrm{P}: \mathrm{p} > 1 \quad (23)$$

The following set eliminates the symmetry between the used plates according to the design index allocated to the first slot of each plate. Such a constraint in (28) requires that when a design, which is at most *i*, is allocated to the first slot of plate *p,* then the first slot of the previous plate cannot have a design greater than *i*. With this constraint in place, the used plates are ordered in a non-decreasing way. Figure 15 presents two solutions with 3 plates. The solutions are equivalent and differ only according to the plates' ordering. Solution 1 follows the order imposed by constraint set (28). Solution 2, in contrast, fails to follow this order and as such is eliminated.

$$\sum_{l=1}^{i} X_{ljp} + \sum_{l=i+1}^{|I|} X_{l,j,p-1} \leq 1 \qquad \forall i \in I, \forall j = 1, \forall p \in P{:}\, p > 1 \quad (28)$$

Constraints (27) and constraints (28) follow a similar logic, and are indicated by symmetry-breaking constraint 2 ($\mathbf{SBC2}$). This thesis indicates the basic formulation, augmented with constraints (27), (23) and (28), as SBC2^{a}.

Figure 15: The order imposed by constraint set (28)

Solution 1:

Plate 1

Design 1	Design 2	Design 2	Design 2	Design 2	Design 2	Design 3
Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7

Plate 2

Design 4	Design 4	Design 5	Design 5	Design 6	Design 6	Design 6
Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7

Plate 3

Design 7	Design 7	Design 7	Design 8	Design 8	Design 9	Design 9
Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7

Solution 2:

Plate 1

Design 7	Design 7	Design 7	Design 8	Design 8	Design 9	Design 9
Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7

Plate2

Design 1	Design 2	Design 2	Design 2	Design 2	Design 2	Design 3
Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7

Plate 3

Design 4	Design 4	Design 5	Design 5	Design 6	Design 6	Design 6
Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7

4.1.4 Third new set of symmetry-breaking constraints

For the third new type of symmetry breaking constraints used in our model, we use a similar idea to that of Jans (2009). Jans (2009) studied the parallel machine lot-sizing problem and uses symmetry breaking constraints to solve the symmetry of identical machines, ultimately putting forward eight different symmetry-breaking constraints. In comparing these, computational tests indicate that the following set is the most powerful for imposing a lexicographic ordering of machines according to the items being set up.

$$\sum_{i=1}^{n} 2^{(n-i)}\, y_{i,k-1,t} \geq \sum_{i=1}^{n} 2^{(n-i)}\, y_{ikt} \qquad \forall k \in M\backslash\{1\}, \forall t \in T$$

These constraints use the following sets and decision variables. Let *M* be the set of identical machines and *T* the set of time periods. There are *n* different items. y_{ikt} is a binary variable indicating that a setup has been done for item *i* in period *t* on machine *k*. The $2^{(n-i)}$ is used as the coefficient related to the setup of item *i* to ensure that a unique number is assigned to each machine, according to its setup schedule. Machines will be ordered in a non-decreasing way according to this unique number.

This logic can usefully be applied to our problem. The following set is used to eliminate the symmetry of slots within a plate. We impose the lexicographic ordering by associating a specific number to each slot. For a constraint in (29), this specific number is equal to $2^{(|I|-i)}$, if design *i* is allocated to the slot. For a given plate, the designs are allocated to slots to ensure that the slots follow a non-decreasing order of this value. Figure 16 depicts two equivalent plates with seven slots, which only differ by having the designs in different slots of each plate. The plate 1 follows the order imposed by constraint set (29). Since plate 2 does not follow this order, it is therefore eliminated.

$$\sum_{i=1}^{|I|} 2^{(|I|-i)} X_{i,j-1,p} \geq \sum_{i=1}^{|I|} 2^{(|I|-i)} X_{ijp} \qquad \forall j \in J: j > 1, \forall p \in P \quad (29)$$

Figure 16: The order imposed by constraint set (29)

Plate 1	Value of $\sum_{i=1}^{\|I\|} 2^{(\|I\|-i)} X_{ijp}$	Plate 2	Value of $\sum_{i=1}^{\|I\|} 2^{(\|I\|-i)} X_{ijp}$
Design 1-slot 1	$2^{(5-1)} = 16$	Design 4-slot 1	$2^{(5-4)} = 2$
Design 1-slot 2	$2^{(5-1)} = 16$	Design 1-slot 2	$2^{(5-1)} = 16$
Design 2-slot 3	$2^{(5-2)} = 8$	Design 5-slot 3	$2^{(5-5)} = 1$
Design 3-slot 4	$2^{(5-3)} = 4$	Design 1-slot 4	$2^{(5-1)} = 16$
Design 4-slot 5	$2^{(5-4)} = 2$	Design 4-slot 5	$2^{(5-4)} = 2$
Design 4-slot 6	$2^{(5-4)} = 2$	Design 2-slot 6	$2^{(5-2)} = 8$
Design 5-slot 7	$2^{(5-5)} = 1$	Design 3-slot 7	$2^{(5-3)} = 4$

Constraint set (23) is kept the same to differentiate which plates are used and which are not.

$$W_{p-1} \geq W_p \qquad \forall p \in P: p > 1 \quad (23)$$

The following set eliminates the symmetry between used plates based on the designs allocated to the first slot in each plate. Plates are ordered lexicographically by assigning a unique number to each plate. The unique number in constraints (30) is equal to $2^{(|I|-i)}$, if design *i* is allocated to the first slot of the plate.

$$\sum_{i=1}^{|I|} 2^{(|I|-i)} X_{i,j,p-1} \geq \sum_{i=1}^{|I|} 2^{(|I|-i)} X_{ijp} \qquad j = 1, \forall p \in P: p > 1 \quad (30)$$

When the total number of designs exceeds 40, the coefficient 2 to the power 40 or higher is a very large number, which can lead to internal calculation issues due to numerical instability. In light of this, if the number of designs exceeds 40, we will only consider the first 40 designs, leading to a partial lexicographic order. Constraints in (30) are hence only applied to the first 40 designs, which result in the following constraint:

$$\sum_{i=1}^{40} 2^{(40-i)} X_{i,j,p-1} \geq \sum_{i=1}^{40} 2^{(40-i)} X_{ijp} \qquad j = 1, \forall p \in P: p > 1$$

In Figure 17, two solutions with 3 plates are presented. The solutions are equivalent, differing only in the ordering of the plates. Solution 1 follows the order imposed by constraint set (30). Because solution 2 fails to follow this order, it is eliminated.

Constraints (29) and (30) follow a similar logic and are henceforth indicated as symmetry-breaking constraint 3 (**SBC3**) while the basic formulation augmented with constraints (29), (23) and (30) is indicated as SBC3^{a}.

Figure 17: The order imposed by constraint set (30)

Value of $\sum_{i=1}^{|I|} 2^{(|I|-i)} X_{i1p}$

Solution 1:

Plate 1

Design 1	Design 2	Design 2	Design 2	Design 2	Design 2	Design 3
Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7

$2^{(9-1)} = 256$

Plate 2

Design 4	Design 4	Design 5	Design 5	Design 6	Design 6	Design 6
Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7

$2^{(9-4)} = 32$

Plate 3

Design 7	Design 7	Design 7	Design 8	Design 8	Design 9	Design 9
Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7

$2^{(9-7)} = 4$

Solution 2:

Plate 1

Design 7	Design 7	Design 7	Design 8	Design 8	Design 9	Design 9
Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7

$2^{(9-7)} = 4$

Plate2

Design 1	Design 2	Design 2	Design 2	Design 2	Design 2	Design 3
Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7

$2^{(9-1)} = 256$

Plate 3

Design 4	Design 4	Design 5	Design 5	Design 6	Design 6	Design 6
Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7

$2^{(9-4)} = 32$

4.2 New symmetry breaking constraints for the alternative formulation

As discussed in Chapter 3, Baumann et al. (2014^b) reformulated the printing-planning problem using new decision variables to eliminate part of its inherent symmetry. Instead of using variables that specify which design is allocated to a specific slot on the plate (the binary variable X_{ijp}, equal to 1 if design i is allocated to slot j on plate p, otherwise being equal to 0), the authors use a new variable that indicates the number of slots on the plate occupied by a specific design. This is the binary variable K_{inp}, which is equal to 1 if design i is allocated to n slots on plate p; otherwise, it is equal to 0. In this way, the inherent symmetry of the slots is avoided altogether. In addition, the authors use constraints (21) to eliminate the symmetry between the used and unused plates. However, there is still a symmetry issue between the used plates, which can be overcome by adding new symmetry-breaking constraints. This thesis proposes four new types of symmetry breaking constraints for the alternative formulation proposed by Baumann et al.(2014^b).

It should be noted that there is a significant difference between the basic formulation and the alternative formulation. The new variable indicates, via index n, to how many slots a specific design is allocated. The original idea was to order the used plates based on the design allocated to the first slot. Since this new variable K_{inp} leads to that the specific information of the first slot is no longer present in the alternative formulation, new strategies are needed to impose an order between the used plates. These new symmetry breaking constraints are based on the presence (or absence) of a design on the whole plate, not just the design in the first slot. We therefore cannot compare the first slot on each plate when trying to eliminate symmetry between the used plates, but must instead consider all slots on the plate.

Thus, the logic of SBC2 (constraints (27) and constraints (28)) cannot be applied to this alternative formulation, since SBC2's logic would result in the condition that when a design whose index is at most *i* is allocated to plate *p,* a design with an index greater than *i* cannot be allocated to plate *p-1*. However, according to the condition of the printing-planning problem, we know that when design *i* is allocated to plate *p*, we can still allocate a design with an index greater than *i* to plate *p-1*. In light of this, this paper cannot use the logic of SBC2 in the alternative formulation, but instead uses the logic of SBC0, SBC1 and SBC3 adopted for the alternative formulation to eliminate the symmetry between the used plates.

Symmetry breaking constraint set (31) imposes a non-increasing order to the plate by associating a specific number to each plate, equal to the sum of the indices of the designs allocated to the plate. Symmetry breaking constraint set (31) is inspired by a similar logic as that of SBC0. We indicate the alternative formulation augmented with symmetry breaking constraint set (31) as SBC0^b. Figure 18 presents two solutions with 3 plates; the solutions are equivalent and differ only in the ordering of the plates. Solution 1 follows the order constraint set (31) imposed. Since solution 2 fails to follow this order, it is eliminated.

$$\sum_{i\in I}\sum_{n\in J} i * K_{i,n,p-1} \geq \sum_{i\in I}\sum_{n\in J} i * K_{i,n,p} \qquad \forall p \in P: p > 1 \quad (31)$$

Figure 18: The order imposed by constraint set (31)

Solution 1:

		Value of $\sum_{i\in I}\sum_{n\in J} i * K_{i,n,p}$
Plate 1	3 slots for design 7，2 slots for design 8，2 slots for design 9	7+8+9=24
Plate 2	2 slots for design 4，2 slots for design 5，3 slots for design 6	4+5+6=15
Plate 3	1 slot for design 1，5 slots for design 2，1 slot for design 3	1+2+3=6

Solution 2:

Plate 1	2 slots for design 4，2 slots for design 5，3 slots for design 6	4+5+6=15
Plate2	1 slot for design 1，5 slots for design 2，1 slot for design 3	1+2+3=6
Plate 3	3 slots for design 7，2 slots for design 8，2 slots for design 9	7+8+9=24

Symmetry breaking constraints set (32) eliminate symmetry between used plates using the index of designs allocated to each plate. According to this constraint, if design i is assigned to plate p (i.e., $\sum_{n\in J} K_{i,n,p} = 1$), then a design with an index equal to or smaller than i should be assigned to plate p-1 (i.e.$\sum_{l\leq i}\sum_{n\in J} K_{l,n,p-1} = 1$). This results in an ordering of the plates according to the lowest index design present in each plate. Symmetry breaking constraint set (32) uses a similar logic to that found in SBC1. The alternative formulation augmented with symmetry breaking constraints set (32) is denoted as $\mathrm{SBC1}^{b}$. Figure 19 presents two solutions with 3 plates each. Since solution 1 follows the order imposed by constraint set (32), while solution 2 does not, solution 2 is eliminated by (32) in favour of solution 1.

$$\sum_{n\in J} K_{inp} \leq \sum_{l\leq i}\sum_{n\in J} K_{l,n,p-1} \qquad \forall i \in I, \forall p \in P: p > 1 \quad (32)$$

Figure 19: The order imposed by constraint set (32)

Solution 1:

Plate 1

1 slot for design 1， 5 slots for design 2， 1 slot for design 3

Plate 2

2 slots for design 4， 2 slots for design 5， 3 slots for design 6

Plate 3

3 slots for design 7， 2 slots for design 8， 2 slots for design 9

Solution 2:

Plate 1

5 slots for design 1， 2 slots for design 4

Plate2

3 slots for design 2， 2 slots for design 3， 2 slots for design 6

Plate 3

1 slot for design 5， 2 slot for design 7， 1 slots for design 8, 3 slots for design 9

Constraint set (33) eliminates symmetry issues between used plates by imposing a lexicographic ordering on the plates. Plates are ordered in a non-decreasing way by assigning a unique number to each plate. Each design i contributes a value of $2^{(|I|-i)}$ to the total of this unique number if design i is allocated to the plate.

$$\sum_{i=1}^{|I|} 2^{(|I|-i)} * \sum_{n \in J} K_{i,n,p-1} \geq \sum_{i=1}^{|I|} 2^{(|I|-i)} * \sum_{n \in J} K_{i,n,p} \qquad \forall p \in P: p > 1 \quad (33)$$

As with constraint set (30), when the total number of designs exceeds 40, the coefficient 2 to the power of 40 or higher produces a very large number, which could lead to numerical instability. Therefore, if the number of designs exceeds 40, we will only consider the first 40 designs, leading to a partial lexicographic order.

$$\sum_{i=1}^{40} 2^{(40-i)} * \sum_{n \in J} K_{i,n,p-1} \geq \sum_{i=1}^{40} 2^{(40-i)} * \sum_{n \in J} K_{i,n,p} \qquad \forall p \in P: p > 1$$

Symmetry breaking constraint set (33) is inspired by a logic similar to that of SBC3. We indicate the alternative formulation augmented with symmetry breaking constraint set (33) as $\mathrm{SBC3}^b$. In Figure 20, two solutions with 3 plates are presented. These solutions are equivalent and differ only by the ordering of the plates. Solution 1 follows the order imposed by constraint set (33). Solution 2 does not and so is eliminated.

Figure 20: The order imposed by constraint set (33)

Solution 1:

Value of $\sum_{i=1}^{|I|} 2^{(|I|-i)} * \sum_{n \in J} K_{i,n,p}$

Plate 1

1 slot for design 1， 5 slots for design 2， 1 slot for design 3

$2^{(9-1)} + 2^{(9-2)} + 2^{(9-3)} = 448$

Plate 2

2 slots for design 4， 2 slots for design 5， 3 slots for design 6

$2^{(9-4)} + 2^{(9-5)} + 2^{(9-6)} = 56$

Plate 3

3 slots for design 7， 2 slots for design 8， 2 slots for design 9

$2^{(9-7)} + 2^{(9-8)} + 2^{(9-9)} = 7$

Solution 2:

Plate 1

3 slots for design 7， 2 slots for design 8， 2 slots for design 9

$2^{(9-7)} + 2^{(9-8)} + 2^{(9-9)} = 7$

Plate2

1 slot for design 1， 5 slots for design 2， 1 slot for design 3

$2^{(9-1)} + 2^{(9-2)} + 2^{(9-3)} = 448$

Plate 3

2 slots for design 4， 2 slots for design 5， 3 slots for design 6

$2^{(9-4)} + 2^{(9-5)} + 2^{(9-6)} = 56$

4.3 Fixing a variable

As pointed out in the literature review, fixing a variable is a special case of adding symmetry breaking constraints to a formulation. This section therefore attempts to figure out how fixing a variable affects the overall formulation, and how fixing a variable interacts with other symmetry breaking constraints.

Regarding the basic formulation and its alternatives, $\mathrm{SBC0^a} - 3^{\mathrm{a}}$, we fix a binary variable X_{ijp} (which is equal to 1, if design *i* is allocated to slot *j* of plate *p*; otherwise it is equal to 0). The fixed variable X_{111} in (34) is added to the basic formulation and to $\mathrm{SBC0^a} - 3^{\mathrm{a}}$. The constraint means that design 1 must be allocated to slot 1 of plate 1. Figure 21 presents two solutions with 1 plate, with solution 1 following the order imposed by constraint (34), while solution 2, failing to follow this order, is eliminated by constraint (34).

$$X_{111} = 1 \quad (34)$$

However, we find that constraint $X_{111} = 1$ conflicts with the $\mathrm{SBC0^a}$, since constraints (24) requires that design 1 should be allocated to the first slot of the last used plate and the constraint $X_{111} = 1$ requires that design 1 should be allocated to the first slot of the first plate.

Furthermore, we cannot fix the variable in the alternative formulation, as we cannot fix the binary variable K_{inp}, which indicates that *n* slots on the plate p are occupied by a specific design i. The reason is that we do not know a priori how many slots will be occupied by a specific design. When we fix $K_{111} = 1$, this means that design 1 only occupies 1 slot on plate 1, which eliminates some feasible solutions.

Figure 21: The order imposed by constraint set (34)

Solution 1:

Plate 1

Design 1 Slot 1	Design 2 Slot 2	Design 2 Slot 3	Design 2 Slot 4	Design 2 Slot 5	Design 2 Slot 6	Design 3 Slot 7

Solution 2:

Plate 1

Design 3 Slot 1	Design 2 Slot 2	Design 2 Slot 3	Design 2 Slot 4	Design 2 Slot 5	Design 2 Slot 6	Design 1 Slot 7

4.4 Imposing a hierarchical order based on the colors

All the previous symmetry breaking constraints impose orders based on the presence or absence of a design in a slot on a plate to eliminate the symmetry among both the slots and the plates. However, we are curious as to how imposing the order based on another criterion would impact these results. In the printing problem, each design has a specific color. It is therefore possible to consider using the presence or absence of a color as the criterion to impose an order to eliminate symmetry among the slots and plates. In both formulations, there is only one variable, Z_{cp}, which is related to the color. The binary variable Z_{cp} equals one if any design with color c is allocated to plate p, and is otherwise zero. The variable Z_{cp} is similar to the variable K_{inp} used in the alternative formulation, in the sense that it applies to all the slots of the plate as a whole. Therefore, this new criterion can only be applied to eliminate the symmetry between the used plates.

Since the alternative formulation already eliminates the inherent symmetry between slots using new variables and eliminates the symmetry between the plates which are used and those which are not used using constraints (21), we first add the symmetry breaking constraints that impose the hierarchical order based on the color index to the alternative formulation in order to eliminate the symmetry between the used plates. Three new symmetry breaking constraints, adopted from the logics of SBC0, SBC1 and SBC3, are presented here, since the logic of SBC2 is no longer applicable with color as the criterion.

Symmetry breaking constraints (35) - (37), which impose a hierarchical order based on the color index, are adopted from the logics of SBC0, SBC1 and SBC3, respectively.

Symmetry breaking constraint set (35) imposes a non-increasing order to the plate by associating a specific number to each plate. The specific number in a constraint of (35) is equal to the sum of the indices of the colors allocated to the plate. Symmetry breaking constraint set (35) is inspired by a similar logic to that of SBC0. We indicate the alternative formulation augmented with symmetry breaking constraints (35) as SBC0^{c}.

Figure 22 shows two solutions with 3 plates; the solutions are equivalent, and differ only in the ordering of the plates. Solution 1 follows the order imposed by constraint set (35), while solution 2 does not and is therefore eliminated.

$$\sum_{c \in C} c * Z_{c,p-1} \geq \sum_{c \in C} c * Z_{c,p} \; \forall p \in P: p > 1 \quad (35)$$

Figure 22: The order imposed by constraint set (35)

Solution 1:

		Value of $\sum_{c \in C} c * Z_{c,p}$
Plate 1	4 slots for color 5, 3 slots for color 6	5+6=11
Plate 2	2 slots for color 3, 5 slots for color 4	3+4=7
Plate 3	6 slots for color 1, 1 slot for color 2	1+2=3

Solution 2:

Plate 1	2 slots for color 3, 5 slots for color 4	3+4=7
Plate2	6 slots for color 1, 1 slot for color 2	1+2=3
Plate 3	4 slots for color 5, 3 slots for color 6	5+6=11

Symmetry breaking constraint set (36) eliminates symmetry amongst the used plates using the index of color allocated to each plate. Such a constraint imposes the rule that if color *c* is assigned to plate *p*, then a color with an index equal to or smaller than *c* should be assigned to plate *p-1.* Plates are thus ordered in a non-decreasing way based on the smallest index of the color in each plate.

$$Z_{c,p} \leq \sum_{l \leq c} Z_{l,p-1} \qquad \forall c \in C, \forall p \in P: p > 1 \quad (36)$$

Symmetry breaking constraint set (36) is inspired by a logic similar to that found in SBC1. The paper refers to the alternative formulation augmented with symmetry breaking constraints (36) as $\mathrm{SBC1}^{c}$. Figure 23 presents two solutions with 3 plates each. Solution 1 follows the order constraint set (36) imposed. Solution 2, in contrast, does not, and as such, is eliminated.

Figure 23: The order imposed by constraint set (36)

Solution 1:

Plate 1

6 slots for color 1, 1 slot for color 2

Plate 2

2 slots for color 3, 5 slots for color 4

Plate 3

4 slots for color 5, 3 slots for color 6

Solution 2:

Plate 1

5 slots for color 1, 2 slots for color 6

Plate2

3 slots for color 2, 4 slots for color 3

Plate 3

4 slots for color 4, 3 slots for color 5

Constraint set (37) eliminates the symmetry between the used plates by imposing a lexicographic ordering on the plates using the colors. Plates are ordered in a non-decreasing way by associating a unique number to each plate. Each color c contributes a total value of $2^{(|C|-c)}$ to the total value of this unique number if color c is assigned to the plate.

$$\sum_{c \in C} 2^{(|C|-c)} * Z_{c,p-1} \geq \sum_{c \in C} 2^{(|C|-c)} * Z_{c,p} \qquad \forall p \in P: p > 1 \quad (37)$$

Symmetry breaking constraint set (37) is inspired by a similar logic to that found in SBC3, and the alternative formulation augmented with symmetry breaking constraints (37) is therefore referred to as $\mathrm{SBC3}^{c}$. Figure 24 displays two solutions with 3 plates. Solution 1 follows the order constraint set (37) imposed, while solution 2 does not and is eliminated.

We also add constraints (35) - (37) to the basic formulation respectively to create three new formulations. Because the binary variable Z_{cp} can only be used to eliminate symmetry between the used plates, the symmetry of slots is eliminated by constraints (22), (25), and (29) in the three respective formulations. Symmetry between those plates that are used and those that are not is eliminated by constraints (23). The basic formulation augmented with symmetry breaking constraints (22), (23), and (35) is indicated as $\mathrm{SBC0^d}$, while the basic formulation augmented with symmetry breaking constraints (25), (23), and (36) is $\mathrm{SBC1^d}$, and the basic formulation augmented with symmetry breaking constraints (29), (23), and (37) is $\mathrm{SBC3^d}$. It now becomes possible to compare the results of models that eliminate symmetries by the same criterion (the presence or absence of a design in a slot on a plate to deal with both the slot and plate symmetry) and models that eliminate different symmetries using different criteria (the presence or absence of a design in a slot on a plate to deal with the slot symmetry and the presence or absence of a color on the plate to deal with the plate symmetry).

Figure 24: The order imposed by constraint set (37)

Solution 1:

Value of $2^{(|C|-c)}$

Plate 1	6 slots for color 1, 1 slot for color 2	$2^{(6-1)} + 2^{(6-2)} = 48$
Plate 2	2 slots for color 3, 5 slots for color 4	$2^{(6-3)} + 2^{(6-4)} = 12$
Plate 3	4 slots for color 5, 3 slots for color 6	$2^{(6-5)} + 2^{(6-6)} = 3$

Solution 2:

Plate 1	5 slots for color 4, 2 slots for color 6	$2^{(6-4)} + 2^{(6-6)} = 5$
Plate2	3 slots for color 2, 4 slots for color 1	$2^{(6-2)} + 2^{(6-1)} = 48$
Plate 3	4 slots for color 3, 3 slots for color 5	$2^{(6-3)} + 2^{(6-5)} = 10$

4.5 Summary of all the models we proposed

In Section 4.5, we provide a clear summary of all the models we proposed. We can see from Table 3 that the first column indicates the name of each model used. The second column indicates the models we proposed is based on the basic formulation or alternative formulation and which constraints are used in each model. The third column indicates the new symmetry breaking constraints impose orders based on the presence of the designs, or on the presence of colors, or both. The fourth column indicates whether the formulation is new or it was already proposed in one of the Baumann papers.

Table 3: Summary of all the models we proposed

Name	Components	Criteria	New formulation or not
Basic formulation	Basic formulation	the presence of the designs	Proposed in Baumann (2014^{a})
$\mathbf{SBC0^{a}}$	Basic formulation +(22), (23) and (24)	the presence of the designs	Proposed in Baumann (2014^{a})
$\mathbf{SBC1^{a}}$	Basic formulation +(25), (23) and (26)	the presence of the designs	New
$\mathbf{SBC2^{a}}$	Basic formulation +(27), (23) and (28)	the presence of the designs	New
$\mathbf{SBC3^{a}}$	Basic formulation +(29), (23) and (30)	the presence of the designs	New
Alternative formulation	Alternative formulation	the presence of the designs	Proposed in Baumann (2014^{b})
$\mathbf{SBC0^{b}}$	Alternative formulation +(31)	the presence of the designs	New
$\mathbf{SBC1^{b}}$	Alternative formulation +(32)	the presence of the designs	New
$\mathbf{SBC3^{b}}$	Alternative formulation +(33)	the presence of the designs	New
Basic formulation + $\mathbf{X_{111} = 1}$	Basic formulation + (34)	the presence of the designs	New
$\mathbf{SBC1^{a}+X_{111} = 1}$	Basic formulation +(25), (23), (26) and (34)	the presence of the designs	New
$\mathbf{SBC2^{a}+X_{111} = 1}$	Basic formulation +(27), (23), (28) and (34)	the presence of the designs	New
$\mathbf{SBC3^{a}+X_{111} = 1}$	Basic formulation +(29), (23), (30) and (34)	the presence of the designs	New
$\mathbf{SBC0^{c}}$	Alternative formulation +(35)	the presence of the colors	New
$\mathbf{SBC1^{c}}$	Alternative formulation +(36)	the presence of the colors	New
$\mathbf{SBC3^{c}}$	Alternative formulation +(37)	the presence of the colors	New
$\mathbf{SBC0^{d}}$	Basic formulation +(22), (23) and (35)	Both	New
$\mathbf{SBC1^{d}}$	Basic formulation +(25), (23) and (36)	Both	New
$\mathbf{SBC3^{d}}$	Basic formulation +(29), (23) and (37)	Both	New

Chapter 5 Computational results

We use OPL to formulate the MIP models, and use IBM ILOG CPLEX Optimization Studio 12.6.3 as the solver. All computations were performed on a standard computer with Intel(R) Core™ i5-3317U CPU with 1.70GHz clock speed and 4 GB of RAM.

5.1 Test data set

This thesis uses an existing data set used by Baumann et al. $(2014^a,\ 2014^b)$. There are 72 instances in the data file used here, which were created based on the original data provided by the company. The authors generated these 72 instances based on several parameters, including the number of designs, white-border ratio, color-code ratio, and demand ratio. We next shortly discuss these 4 aspects. Baumann et al. (2014^a) added these 72 instances in order to test a heuristic. Baumann et al. (2014^b) applied the first 40 instances in order to test the basic formulation augmented with symmetry breaking constraints and the alternative formulation. The structure of data is discussed in more detail in Baumann et al. (2014^a). This thesis uses only the first 56 instances for its test, since none of the models used here are able to find a feasible solution for the last 16 instances.

Number of designs N: We classify the size of instances based on this parameter. Small-size instances have 5, 10, and 15 different customer-specific designs. Medium instances have 20 and 25 customer-specific designs, and large instances have 30 and 50 customer-specific designs. Each level of the number of designs has 8 instances. Because the number of designs is the most important factor influencing the size of the model, with respect to the number of decision variables and constraints, we analyse the results based on the number of designs.

Color ratio CR: There are two levels used to set the maximum number of colors. The lower level of color ratio is 0.15, while the higher level is 0.3. Baumann et al. (2014^{a}) multiply these ratios by the number of customer-specific designs to get the maximum number of colors. A random color is assigned to each design.

White-border ratio WR: There are two levels used to set the customer-specific design with or without a white-border. The lower level of white-border ratio is 0.33, and the higher level ratio is 0.66. A random number between 0 and 1 was generated for each design. If this random number is greater than the white-border ratio, the design is without white-border and otherwise it has a white border. The standard design is set to always have a white-border.

Demand ratio DR: There are two levels for the demand ratio: a lower level of 0.2 and a higher level of 0.4. A maximum number of different demand values for each instance is set by multiplying the demand ratio by the number of designs. The demand value in each instance is then chosen randomly from the range of admissible demand values, which ranges from 5,000 to 80,000, derived from the original data provided by the company.

In the data set used here, each level of design numbers has eight instances. The data set therefore includes 24 small-size instances, 16 medium-sized instances, and 16 large-size instances. The goal of all formulations is to minimize the total cost, including set up costs and overproduction costs. In all instances, the setup cost per plate is \$540. The overproduction cost per unit of customer-specific design is \$0.0035, and the overproduction cost per unit of standard design is \$0.001.

We set a CPU time limit of 1800 seconds per instance for the CPLEX solver, and an upper limit on the working memory of 4096 megabytes. When the size of the search tree exceeds the working memory, the program transfers the node file to the disk.

We compare computational results on five aspects: the value of the best feasible solution (B), the MIP gap (G), the required CPU time (T), the number of optimal solution (O), and the number of feasible solutions generated (F). The MIP gap, calculated as a percentage, is equal to the value of the best feasible solution minus the best lower bound at the end of the optimization process, divided by the value of the best feasible solution (Pochet and Wolsey 2006). When the value of the best feasible solution is equal to the best lower bound at the end of the optimization process, the optimal solution is found. The required CPU time (T) is measured in seconds. As the number of designs is the main driver which affects the complexity of the model, we aggregate the results according to the number of designs. We average the instance results, including the value of the best feasible solution (B), the MIP gap (G), and the required CPU time (T), on all instances which have the same number of designs. We also add up the number of optimal solutions (O) and the number of feasible solution (F) that the model obtained, again, when the number of designs is the same. The entry ‘na’ indicates that no feasible solution was found within the time limit, for at least one instance in that class. To ensure fairness, we only compare the results when the model obtains all the feasible solutions within each level of the number of designs.

5.2 Numerical results

5.2.1 Results of the basic formulation and the basic formulations augmented with different symmetry breaking constraints

The results of the basic formulation, $\mathrm{SBC0}^{a}$, $\mathrm{SBC1}^{a}$, $\mathrm{SBC2}^{a}$, and $\mathrm{SBC3}^{a}$ are reported in Table 4. The first column is the number of designs (N). Columns 2-26 present average results for the best feasible solution (B), the MIP gap (G), the required CPU time (T), the number of optimal solutions (O), and the number of feasible solutions (F), generated by the basic formulation, the $\mathrm{SBC0}^{a}$, the $\mathrm{SBC1}^{a}$, the $\mathrm{SBC2}^{a}$, the $\mathrm{SBC3}^{a}$, with the number of designs held to be the same. For average instance results, with the number of designs being the same, the best results obtained for the best feasible solution (B), the MIP gap (G), and the required CPU time (T), are marked in bold.

Comparing the results of the basic formulation augmented without SBCs with those produced by an SBC-augmented model produces the same conclusion as Baumann et al. (2014)[a]. For small size instance, the basic formulation augmented with SBC is superior to the basic formulation without SBCs, since SBCs significantly reduce the CPU time required to find an optimal solution. When the number of designs is 10, the basic formulation without SBCs only finds 2 optimal solutions out of 8 instances within the time limit. The basic formulations augmented with SBCs, in contrast, are able to find all the optimal solutions within the same time limit. As the number of designs increases, the basic formulation augmented without SBCs becomes incapable of finding any optimal solution whatsoever. However, for medium and large-size instances, while the basic formulations augmented with SBCs cannot find the optimal solution within the time limit, the value of the best feasible solution obtained is better than that produced by the non-SBC augmented model. Therefore, it seems safe to assume that the symmetry breaking constraints do in fact help the model cut the nodes of the searching tree earlier, allowing it to use less time finding the optimal solution, but also increase the difficulty of finding a feasible solution.

Next, we compare in detail the basic formulation with different symmetry breaking constraints. When the number of designs is 5, $\mathrm{SBC2}^{a}$ takes the least time to find the optimal solution, with $\mathrm{SBC1}^{a}$ and $\mathrm{SBC3}^{a}$ only taking slightly longer. $\mathrm{SBC0}^{a}$ is the worst performer in terms of required CPU time, taking 40% more CPU time compared with $\mathrm{SBC2}^{a}$. When the number of designs increases to 10， $\mathrm{SBC1}^{a}$ has the lowest CPU time, while $\mathrm{SBC0}^{a}$ takes 4.7 times longer than $\mathrm{SBC1}^{a}$ to find the optimal solution. When the number of designs is 15， $\mathrm{SBC3}^{a}$ has the lowest CPU time, saving 68% CPU time compared with $\mathrm{SBC0}^{a}$, while $\mathrm{SBC1}^{a}$ and $\mathrm{SBC2}^{a}$ saved almost 45% CPU time compared with $\mathrm{SBC0}^{a}$. When the number of designs increases to 20, $\mathrm{SBC0}^{a}$ is unable to find the optimal solution at all. However, $\mathrm{SBC1}^{a}$， $\mathrm{SBC2}^{a}$ and $\mathrm{SBC3}^{a}$ can still find some optimal solutions. Therefore, $SBC0^a$ takes more time than $SBC1^a$, $SBC2^a$ and $SBC3^a$ to find the optimal solution within the time limit. As the number of designs continues to increase, $SBC0^a$ has more difficulties of finding feasible solutions compared with $SBC1^a$, $SBC2^a$ and $SBC3^a$. $SBC1^a$ performs the best in finding feasible solutions since it is able to find the most feasible solutions out of all the basic models with different SBCs. $SBC3^a$ has the best objective function and MIP gap when the number of designs is 25. $SBC1^a$ has the best objective function and MIP gap when the number of designs is 30. When the number of designs increases to 50, none of the formulations are able to find feasible solutions for all instances within the time limit.

To summarize, $SBC1^a$, $SBC2^a$ and $SBC3^a$ can significantly reduce the required CPU time compared with $SBC0^a$, with $SBC1^a$ performing the best in terms of finding feasible solutions.

Table 4: Results of the basic formulation, SBC0[a], SBC1[a], SBC2[a], and SBC3[a]

	Basic formulation					SBC0[a]					SBC1[a]					SBC2[a]					SBC3[a]				
N	B	G	T	O	F	B	G	T	O	F	B	G	T	O	F	B	G	T	O	F	B	G	T	O	F
5	762	0.0%	0.86	8	8	762	0.0%	0.22	8	8	762	0.0%	0.18	8	8	762	0.0%	**0.14**	8	8	762	0.0%	0.15	8	8
10	1301	9.6%	1586	2	8	1301	0.0%	11.84	8	8	1301	0.0%	**2.51**	8	8	1301	0.0%	3.66	8	8	1301	0.0%	2.90	8	8
15	1919	15.2%	1800	0	8	**1915**	1.1%	671	7	8	1924	2.8%	400	7	8	**1915**	**0.0%**	337	8	8	**1915**	**0.0%**	**219**	8	8
20	**2644**	37.6%	1800	0	8	3140	45.7%	1800	0	8	2887	**26.9%**	1576	2	8	2756	28.1%	**1531**	2	8	2964	35.0%	1755	1	8
25	**3541**	**42.4%**	1800	0	8	4469	52.2%	1800	0	8	4359	51.1%	1800	0	8	4926	57.0%	1800	0	8	3844	42.9%	1800	0	8
30	**4543**	**47.6%**	1800	0	8	na	na	1800	0	6	**7180**	**62.7%**	1800	0	8	9477	68.6%	1800	0	8	14602	67.0%	1800	0	8
50	**14139**	**69.9%**	1800	0	8	na	na	1800	0	2	na	na	1800	0	7	na	na	1800	0	4	na	na	1800	0	2

5.2.2 Results of the alternative formulation and the alternative formulations augmented with different symmetry breaking constraints

The result of the alternative formulation, $\mathrm{SBC0}^b$, $\mathrm{SBC1}^b$, and $\mathrm{SBC3}^b$ are reported in Table 5.

The first column of Table 5 is the number of designs (N). Columns 2-21 present average results of the value of the best feasible solution (B), the MIP gap (G), the required CPU time (T), the number of optimal solutions (O) and the number of feasible solutions (F) for the new variable model, $\mathrm{SBC0}^b$, $\mathrm{SBC1}^b$, and $\mathrm{SBC3}^b$, with the number of designs held the same across the rows. The entry 'na' indicates that no feasible solution was found within the time limit for at least one instance in that class. From these results, for a given number of designs, the best results obtained for the best feasible solution (B), the MIP gap (G), and the required CPU time (T) are marked in bold.

First, we compare the regular alternative formulation with the alternative formulations augmented with different symmetry breaking constraints ($\mathrm{SBC0}^b$, $\mathrm{SBC1}^b$ and $\mathrm{SBC3}^b$), based on the level of the number of designs, since this is the primary factor affecting the complexity of the model. When the number of designs is 5 or 10, there are only minor differences between these models. $\mathrm{SBC3}^b$ works best with respect to the required CPU time when the number of designs is 5, while $\mathrm{SBC1}^b$ is the fastest to find the optimal solution for 10 designs. When the designs increase to 15, a much larger difference in CPU time emerges between $SBC1^b$, $SBC3^b$ and the alternative formulation, and $\mathrm{SBC0}^b$. On average, $\mathrm{SBC1}^b$ and $\mathrm{SBC3}^b$ use only approximatively

20% of the CPU time required by either the alternative formulation or $\mathrm{SBC0}^b$. When the number of designs is 20, $\mathrm{SBC1}^b$ and $\mathrm{SBC3}^b$ find 5 and 4 optimal solutions, respectively, which is better than either the alternative formulation or $\mathrm{SBC0}^b$, which find only 2 optimal solutions. At 25 designs, none of the four models can find the optimal solution. With respect to the value of the best feasible solution and MIP gaps, $\mathrm{SBC1}^b$ and $\mathrm{SBC3}^b$ are superior to the alternative formulation. A similar situation arose when considering 30 designs. Unfortunately, it is impossible to compare the results from $\mathrm{SBC0}^b$ at these levels as it was unable to find all the feasible solutions for 25 and 30 designs. When the number of designs rises to 50, both the alternative formulation and $\mathrm{SBC1}^b$ find more feasible solutions than either $\mathrm{SBC0}^b$ or $\mathrm{SBC3}^b$.

Next, we compare the alternative formulation augmented with SBCs in Table 5 with the basic formulation augmented with SBCs in Table 4. We can see from the results that the alternative formulations, augmented with different symmetry breaking constraints, take less CPU time and find more optimal solutions than the basic formulation augmented with SBCs. Furthermore, the solutions found by the alternative formulation always have the same or better value for the best feasible solution (B) and the MIP gap (G) than those generated by the basic formulation. However, the basic formulation did find more feasible solutions than the alternative formulation.

To summarize, the alternative formulation is superior to the basic formulation, while $\mathrm{SBC1^b}$ and $\mathrm{SBC3^b}$ outperform the alternative formulation and $\mathrm{SBC0^b}$ in terms of reducing CPU time and optimality gaps for instances with 30 designs or less. SBC1^{b} finds more feasible solutions than other alternative formulations augmented with SBCs.

Table 5: Results of the alternative formulation, $SBC0^b$, $SBC1^b$, and $SBC3^b$

	Alternative formulation					**$SBC0^b$**					**$SBC1^b$**					**$SBC3^b$**				
N	B	G	T	O	F	B	G	T	O	F	B	G	T	O	F	B	G	T	O	F
5	762	0.0%	0.13	8	8	762	0.0%	0.15	8	8	762	0.0%	0.15	8	8	762	0.0%	**0.12**	8	8
10	1301	0.0%	1.71	8	8	1301	0.0%	1.97	8	8	1301	0.0%	**1.25**	8	8	1301	0.0%	1.42	8	8
15	1915	0.0%	121	8	8	1915	0.0%	131	8	8	1915	0.0%	**24**	8	8	1915	0.0%	28	8	8
20	2496	12.8%	1371	2	8	2490	12.9%	1375	2	8	**2473**	**3.5%**	**815**	5	8	2511	6.5%	988	4	8
25	3343	30.6%	1803	0	8	na	na	1598	0	7	**3297**	18.8%	1800	0	8	3302	**18.7%**	1800	0	8
30	4417	41.8%	1806	0	8	na	na	1396	0	5	4327	**36.5%**	1801	0	8	**4290**	36.9%	1800	0	8
50	na	na	1800	0	7	na	na	1800	0	0	na	na	1800	0	6	na	na	1800	0	1

5.2.3 Results of models with variable fixing

Table 6 reports the results of both the basic formulation and the basic formulation augmented with $X_{111} = 1$.

The first column of Table 6 is the number of designs (N). Columns 2-11 present the average results of the best feasible solution (B), the MIP gap (G), the required CPU time (T), the number of optimal solutions (O), and the number of feasible solutions (F), for both the basic formulation and the basic formulation augmented with $X_{111} = 1,$ for N number of designs. The best results obtained for each of these values for both the basic and augmented formulation is marked in bold.

Overall, $X_{111} = 1$ helps the basic formulation reduce the CPU time needed to find the optimal solution by more than 40%. When the number of designs is 10, the basic formulation with $X_{111} = 1$ finds 3 more optimal solutions than the basic formulation. The solutions found by the basic formulation with $X_{111} = 1$ also have the same or better value for the best feasible solution (B) and the MIP gap (G) than those produced by the basic formulation, except for the value of the best feasible solution when the number of design is 25.

Table 6: Results of the basic formulation and the basic formulation with $X_{111} = 1$

	Basic formulation					**Basic formulation with $X_{111} = 1$**				
N	B	G	T	O	F	B	G	T	O	F
5	762	0.0%	0.86	8	8	762	0.0%	**0.44**	8	8
10	1301	9.6%	1586	2	8	1301	**3.6%**	**875**	5	8
15	1919	15.2%	1802	0	8	**1915**	**15.0%**	1802	0	8
20	2644	37.6%	1800	0	8	**2640**	**37.3%**	1800	0	8
25	**3541**	42.4%	1801	0	8	3592	**41.1%**	1800	0	8
30	4543	47.6%	1800	0	8	**4495**	**47.0%**	1800	0	8
50	14139	69.9%	1800	0	8	**12610**	**65.9%**	1800	0	8

Table 7 reports the $\mathrm{SBC1^a - 3^a}$ results, as well as the $\mathrm{SBC1^a - 3^a}$ with $X_{111} = 1$ results. Using this Table, we can figure out whether $X_{111} = 1$ can still help the model obtain better results when the model already has other symmetry breaking constraints. Table 7 has a similar structure to Table 6, in that results are compared in the same five categories: the value of the best feasible solution (B), the MIP gap (G), the required CPU time (T), the number of optimal solutions (O), and the number of feasible solutions (F). The best result for each is marked in bold.

As can be seen from Table 7, results from $\mathrm{SBC1^a - 3^a}$ with $X_{111} = 1$ are not always better than the $\mathrm{SBC1^a - 3^a}$ results when compared respectively. The impact of $X_{111} = 1$ on $\mathrm{SBC1^a - 3^a}$ varied instance by instance. When the number of designs is 5， $\mathrm{SBC2^a}$ with $X_{111} = 1$ takes less time to find the optimal solution compared with $\mathrm{SBC2^a}$. When the number of designs is 10, $\mathrm{SBC2^a}$ with $X_{111} = 1$ and $\mathrm{SBC3^a}$ with $X_{111} = 1$ work better than $\mathrm{SBC2^a}$ and $\mathrm{SBC3^a}$ respectively with respect to the required CPU time. When the number of designs is 15, $\mathrm{SBC1^a}$ with $X_{111} = 1$ and $\mathrm{SBC3^a}$ with $X_{111} = 1$ use less time to find the optimal solution than $\mathrm{SBC1^a}$ and $\mathrm{SBC3^a}$ respectively. A similar situation arose when considering 20 designs, with respect to the value of the best feasible solution and MIP gaps, $\mathrm{SBC1^a}$ with $X_{111} = 1$ and $\mathrm{SBC3^a}$ with $X_{111} = 1$ are superior to $\mathrm{SBC1^a}$ and $\mathrm{SBC3^a}$ respectively. At 25 designs, $\mathrm{SBC2^a}$ with $X_{111} = 1$ obtain the smaller value of the best feasible solution and MIP gaps than $\mathrm{SBC2^a}$. When the number of designs is 30, $\mathrm{SBC1^a - 3^a}$ with $X_{111} = 1$ work better than $\mathrm{SBC1^a - 3^a}$ with respect to the value of the best feasible solution and MIP gaps. When the number of designs rises to 50, none of the models are able to find feasible solutions for all instances within the time limit. Therefore, approximatively 60% of results from $\mathrm{SBC1^a - 3^a}$ are improved by adding $X_{111} = 1$. Furthermore, $X_{111} = 1$ increases the difficulty of finding the feasible solution, because $\mathrm{SBC1^a - 3^a}$ with $X_{111} = 1$ always finds less feasible solutions than $\mathrm{SBC1^a - 3^a}$, when compared respectively. In conclusion, it seems that fixing a variable has not a clear impact for formulations that already contain symmetry breaking constraints.

Table 7: Results of the $\mathrm{SBC1^a - 3^a}$ and the $\mathrm{SBC1^a - 3^a}$ with $\mathrm{X_{111} = 1}$

	$SBC1^a$					**$SBC1^a$ with $X_{111} = 1$**					**$SBC2^a$**					**$SBC2^a$ with $X_{111} = 1$**					**$SBC3^a$**					**$SBC3^a$ with $X_{111} = 1$**				
N	B	G	T	O	F	B	G	T	O	F	B	G	T	O	F	B	G	T	O	F	B	G	T	O	F	B	G	T	O	F
5	762	0.0%	0.18	8	8	762	0.0%	0.22	8	8	762	0.0%	0.14	8	8	762	0.0%	**0.09**	8	8	762	0.0%	0.15	8	8	762	0.0%	0.16	8	8
10	1301	0.0%	**2.51**	8	8	1301	0.0%	4.85	8	8	1301	0.0%	3.66	8	8	1301	0.0%	3.62	8	8	1301	0.0%	2.90	8	8	1301	0.0%	2.65	8	8
15	1924	2.8%	400	7	8	**1915**	**0.0%**	253	8	8	**1915**	**0.0%**	337	8	8	**1915**	**0.0%**	377	8	8	**1915**	**0.0%**	219	8	8	**1915**	**0.0%**	**211**	8	8
20	2887	26.9%	1576	2	8	2807	25.9%	1669	1	8	2756	28.1%	**1531**	2	8	2937	37.3%	1709	1	8	2964	35.0%	1755	1	8	**2727**	**23.8%**	1640	2	8
25	4359	51.1%	1800	0	8	4496	51.6%	1800	0	8	4926	57.0%	1800	0	8	4869	53.6%	1800	0	8	**3844**	**42.9%**	1800	0	8	4170	49.4%	1800	0	8
30	7180	62.7%	1800	0	8	**6305**	**60.9%**	1800	0	8	9477	68.6%	1800	0	8	6966	63.2%	1800	0	8	14602	67.0%	1800	0	8	7024	61.8%	1800	0	8
50	na	na	1800	0	7	na	na	1800	0	5	na	na	1800	0	4	na	na	1800	0	3	na	na	1800	0	2	na	na	1800	0	1

5.2.4 Results of imposing hierarchical order based on the colors

Table 8 reports $\mathrm{SBC0^c}$, $\mathrm{SBC1^c}$, and $\mathrm{SBC3^c}$ results, compared with $\mathrm{SBC0^b}$, $\mathrm{SBC1^b}$, and $\mathrm{SBC3^b}$ results. From this table, we can determine whether using the presence or absence of a color on the plate is better than using the presence or absence of a design in a slot on a plate as a criterion to eliminate symmetry in the printing planning problem. Table 8 shows that $\mathrm{SBC0^c}$ outperforms $\mathrm{SBC0^b}$ with respect to the value of the best feasible solution (B), the MIP gap (G), and the required CPU time (T). Moreover, $\mathrm{SBC0^c}$ finds more optimal solutions and feasible solutions than $\mathrm{SBC0^b}$. When we check $\mathrm{SBC1^c}$ and $\mathrm{SBC3^c}$, we find out that $\mathrm{SBC1^c}$ or $\mathrm{SBC3^c}$ results are typically worse than those generated by either $\mathrm{SBC1^b}$ or $\mathrm{SBC3^b}$ respectively. However, for instances where the number of designs is 30, $\mathrm{SBC1^c}$ performs better than other models.

Table 8: Results of $SBC0^c$, $SBC1^c$, and $SBC3^c$

	$\mathbf{SBC0^b}$					$\mathbf{SBC0^c}$					$\mathbf{SBC1^b}$					$\mathbf{SBC1^c}$					$\mathbf{SBC3^b}$					$\mathbf{SBC3^c}$				
N	B	G	T	O	F	B	G	T	O	F	B	G	T	O	F	B	G	T	O	F	B	G	T	O	F	B	G	T	O	F
5	762	0.0%	0.15	8	8	762	0.0%	0.15	8	8	762	0.0%	0.15	8	8	762	0.0%	0.14	8	8	762	0.0%	**0.12**	8	8	762	0.0%	0.16	8	8
10	1301	0.0%	1.97	8	8	1301	0.0%	1.64	8	8	1301	0.0%	**1.25**	8	8	1301	0.0%	1.53	8	8	1301	0.0%	1.42	8	8	1301	0.0%	1.62	8	8
15	1915	0.0%	131	8	8	1915	0.0%	59	8	8	1915	0.0%	**24**	8	8	1915	0.0%	56	8	8	1915	0.0%	28	8	8	1915	0.0%	57	8	8
20	2490	12.9%	1375	2	8	2482	7.8%	1008	4	8	**2473**	**3.5%**	**815**	5	8	2502	7.1%	990	5	8	2511	6.5%	988	4	8	2535	7.7%	1053	4	8
25	na	na	1598	0	7	3386	25.4%	1803	0	8	**3297**	18.8%	1800	0	8	3469	25.4%	1803	0	8	3302	**18.7%**	1800	0	8	3298	20.3%	1803	0	8
30	na	na	1396	0	5	4648	44.6%	1802	0	8	4327	36.5%	1801	0	8	**4232**	**34.5%**	1801	0	8	4290	36.9%	1800	0	8	4656	42.5%	1804	0	8
50	na	na	1800	0	0	na	na	1800	0	2	na	na	1800	0	6	na	na	1800	0	5	na	na	1800	0	1	na	na	1800	0	3

Table 9 reports $\mathrm{SBC0^d}$, $\mathrm{SBC1^d}$, and $\mathrm{SBC3^d}$ results, compared with those from $\mathrm{SBC0^a}$, $\mathrm{SBC1^a}$, and $\mathrm{SBC3^a}$. This table allows us to evaluate for the basic formulation whether using different criteria to eliminate the symmetry in the printing planning problem, i.e. designs for the slot symmetry and colors for the plate symmetry, is better than using the same criteria to eliminate the symmetry, i.e. designs for both the slot and plate symmetry.

We can see from the Table 9 that for small-size instances (where the design number is 5, 10 and 15), models with SBCs in the same criteria always use less CPU time than models with SBCs in different criteria. For medium-size instances (20 and 25 designs), $\mathrm{SBC0^d}$ outperforms $\mathrm{SBC0^a}$, but $\mathrm{SBC1^d}$ performs poorly compared to $\mathrm{SBC1^a}$. When the number of designs is 30, the situation is similar to the medium-size instances.

Overall, only on a small set of instances (n =20 or 25 for $\mathrm{SBC0^d}$, n=15 for $\mathrm{SBC1^d}$, n=20 for $\mathrm{SBC3}^d$) is there an improvement in the objective function value when considering different criteria instead of the same criterion.

Table 9: Results of $SBC0^d$, $SBC1^d$, and $SBC3^d$

	$SBC0^a$					**$SBC0^d$**					**$SBC1^a$**					**$SBC1^d$**					**$SBC3^a$**					**$SBC3^d$**				
N	B	G	T	O	F	B	G	T	O	F	B	G	T	O	F	B	G	T	O	F	B	G	T	O	F	B	G	T	O	F
5	762	0.0%	0.22	8	8	762	0.0%	0.36	8	8	762	0.0%	0.18	8	8	762	0.0%	0.23	8	8	762	0.0%	**0.15**	8	8	762	0.0%	0.22	8	8
10	1301	0.0%	11.84	8	8	1301	0.0%	13.58	8	8	1301	0.0%	**2.51**	8	8	1301	0.0%	5.82	8	8	1301	0.0%	2.90	8	8	1301	0.0%	4.54	8	8
15	**1915**	1.1%	671	7	8	1916	2.8%	830	6	8	1924	2.8%	400	7	8	**1915**	**0.0%**	503	8	8	**1915**	**0.0%**	**219**	8	8	**1915**	**0.0%**	428	8	8
20	3140	45.7%	1800	0	8	2993	43.3%	1800	0	8	2887	**26.9%**	**1576**	2	8	2963	33.9%	1713	1	8	2964	35.0%	1755	1	8	**2769**	30.7%	1625	1	8
25	4469	52.2%	1800	0	8	4126	46.7%	1800	0	8	4359	51.1%	1800	0	8	4446	52.1%	1800	0	8	**3844**	**42.9%**	1800	0	8	4076	47.4%	1800	0	8
30	na	na	1800	0	6	7855	64.5%	1800	0	8	**7180**	**62.7%**	1800	0	8	11822	75.5%	1800	0	8	14602	67.0%	1800	0	8	7218	66.3%	1800	0	8
50	na	na	1800	0	2	na	na	1800	0	2	na	na	1800	0	7	66281	89.4%	1800	0	8	na	na	1800	0	2	na	na	1800	0	0

5.3 Result summary of all the models

In conclusion to this chapter, we provide a clear summary of all the models that were evaluated. We can see from the Table 10 that the first column indicates the name of each model used. The second column indicates which constraints are used in each model. The third column indicates the total number of optimal solutions each model found, and the fourth column indicates the total number of feasible solutions each model found. The fifth, sixth, and seventh columns indicate the average CPU time required of each model when the number of designs is 5, 10, and 15, respectively. The eighth and ninth columns indicate the value of the best feasible solutions and the MIP gap of each model, respectively, when the number of designs is 20. The tenth and eleventh columns indicate the value of the best feasible solution and the MIP gap of each model when the number of designs is 25. The twelfth and thirteenth columns indicate the value of the best feasible solution and the MIP gap of each model when the number of designs is 30. The fourteenth and fifteenth columns indicate the value of the best feasible solution and the MIP gap of each model when the number of designs is 50. The best results obtained in each column are marked in bold.

From Table 10, we observe that adding symmetry breaking constraints to the original model can reduce the required CPU time to find the optimal solution, but it also increases the difficulty of finding feasible solutions.

In terms of finding feasible solutions, we observe that the basic formulation (with $X_{111} = 1$) finds feasible solutions for all the instances. The only formulation augmented with symmetry breaking constraints that also finds a feasible solution for all instances is $\mathrm{SBC1}^{d}$. When considering only the formulations augmented with symmetry breaking constraints, we observe that all of the newly proposed formulations find more feasible solutions than the augmented formulations originally proposed by Baumann et al. (2014^{a}, 2014^{b}), i.e. $\mathrm{SBC0}^{a}$ and $\mathrm{SBC0}^{b}$.

In terms of finding optimal solutions, we observe that two of the augmented alternative formulations, $\mathrm{SBC1}^{b}$ and $\mathrm{SBC1}^{c}$, are able to find 29 optimal solutions, which is more than any other formulation. $\mathrm{SBC1}^{b}$ has furthermore the lowest overall CPU times for the small instances with 5 to 15 designs. It also finds the best bound for instances with 20 and 25 designs, and it is very close to the best found bound for instances with 30 designs.

Overall, it is clear that SBC0 is inferior to SBC1 – SBC3 in all aspects. Although there is no formulation that wins on all the criteria and instances, it seems that our newly proposed $\mathrm{SBC1}^{b}$ provides an excellent overall result.

Table 10: Result summary of all the models

Name	Components	O	F	T(5)	T(10)	T(15)	B(20)	G(20)	B(25)	G(25)	B(30)	G(30)	B(50)	G(50)
Basic formulation	Basic formulation	10	**56**	0.86	1586	1802	2644	37.6%	3541	42.4%	4543	47.6%	14139	69.9%
$SBC0^a$	Basic formulation +(22), (23) and (24)	23	48	0.22	11.84	671	3140	45.7%	4469	52.2%	na	na	na	na
$SBC1^a$	Basic formulation +(25), (23) and (26)	25	55	0.18	2.51	400	2887	26.9%	4359	51.1%	7180	62.7%	na	na
$SBC2^a$	Basic formulation +(27), (23) and (28)	26	52	0.14	3.66	337	2756	28.1%	4926	57.0%	9477	68.6%	na	na
$SBC3^a$	Basic formulation +(29), (23) and (30)	25	50	0.15	2.90	219	2964	35.0%	3844	42.9%	14602	67.0%	na	na
Alternative formulation	Alternative formulation	26	55	0.13	1.71	121	2496	12.8%	3343	30.6%	4417	41.8%	na	na
$SBC0^b$	Alternative formulation +(31)	26	44	0.15	1.97	131	2490	12.9%	na	na	na	na	na	na
$SBC1^b$	Alternative formulation +(32)	**29**	54	0.15	**1.25**	**24**	**2473**	**3.5%**	**3297**	18.8%	4327	36.5%	na	na
$SBC3^b$	Alternative formulation +(33)	28	49	0.12	1.42	28	2511	6.5%	3302	**18.7%**	4290	36.9%	na	na
Basic formulation+$X_{111} = 1$	Basic formulation + (34)	13	**56**	0.44	875	1802	2640	37.3%	3592	41.1%	4495	47.0%	**12610**	**65.9%**
$SBC1^a$+$X_{111} = 1$	$SBC1^a$+ (34)	25	53	0.22	4.85	253	2807	25.9%	4496	51.6%	6305	60.9%	na	na
$SBC2^a$+$X_{111} = 1$	$SBC2^a$ + (34)	25	51	**0.09**	3.62	377	2937	37.3%	4869	53.6%	6966	63.2%	na	na
$SBC3^a$+$X_{111} = 1$	$SBC3^a$+ (34)	26	49	0.16	2.65	211	2727	23.8%	4170	49.4%	7024	61.8%	na	na
$SBC0^c$	Alternative formulation +(35)	28	50	0.15	1.64	59	2482	7.8%	3386	25.4%	4648	44.6%	na	na
$SBC1^c$	Alternative formulation +(36)	**29**	53	0.14	1.53	56	2502	7.1%	3469	25.4%	**4232**	**34.5%**	na	na
$SBC3^c$	Alternative formulation +(37)	28	51	0.16	1.62	57	2535	7.7%	3298	20.3%	4656	42.5%	na	na
$SBC0^d$	Basic formulation +(22), (23) and (35)	22	50	0.36	13.58	830	2993	43.3%	4126	46.7%	7855	64.5%	na	na
$SBC1^d$	Basic formulation +(25), (23) and (36)	25	**56**	0.23	5.82	503	2963	33.9%	4446	52.1%	11822	75.5%	66281	89.4%
$SBC3^d$	Basic formulation +(29), (23) and (37)	25	48	0.22	4.54	428	2769	30.7%	4076	47.4%	7218	66.3%	na	na

Chapter 6 Analysis of the value of flexibility

6.1 The value of flexibility

This chapter analyses the value of flexibility in the printing planning problem. This analysis indicates the impact of constraints on the total cost. Regarding the printing planning problem, there are many specific constraints, such as the color constraint, the white border constraint, and the split constraint. We study how much the cost of printing changes when we increase the flexibility of these constraints separately.

Constraint (2) assures that if a plate is used, no slot on that plate can be empty; each slot of that plate must be occupied by a design. In order to analyse the value of flexibility with respect to constraint (2), we allow that a given slot can be empty. We change the mathematical symbols of constraint (2) from “equal to” to “less than or equal to”.

$$\sum_{i \in I} X_{ijp} \leq W_p \qquad \forall j \in J; \forall p \in P \quad (2')$$

Constraint (5) imposes the color constraint, requiring that each plate cannot contain more than two different colors simultaneously. In order to analyse the value of flexibility with respect to constraint (5), we gradually increase the maximum number of colors that can be allocated to each plate from 2 to 7. Each plate can contain at most 7 different colors, since each plate has 7 slots and each design has exactly one color.

$$\sum_{c \in C} Z_{cp} \leq \overline{c} \qquad \forall p \in P \quad (5)$$

Constraint (7) imposes the white border constraint, requiring that either two customer-specific designs with a white border or a standard design must be allocated to each plate. To analyse the value of flexibility with respect to constraint (7), we will delete this constraint from the basic formulation to observe how this changes the total cost.

$$W_p \leq \sum_{i\in I_{w};j\in J}\frac{1}{2}X_{ijp} + \sum_{i\in I^s;j\in J}X_{ijp} \qquad \forall p \in P \quad (7)$$

Constraint (8) requires that each plate can contain at most one slot with a standard design. Similar to constraint (7), we will delete this constraint from the basic formulation to see how the total cost changes.

$$\sum_{i\in I^s;j\in J}X_{ijp} \leq 1 \qquad \forall p \in P \quad (8)$$

Constraint (9) imposes the split constraint, requiring that a customer-specific design be allocated to a single plate. We will delete constraint (9) from the basic formulation, again to observe how the total cost changes. At the same time, we will delete constraint (10), since deleting constraint (9) means constraint (10) is no longer needed.

$$\sum_{p\in P}Y_{ip} = 1 \qquad \forall i \in I^o \quad (9)$$

$$|J|Y_{ip} \geq \sum_{j\in J}X_{ijp} \qquad \forall i \in I^o; \forall p \in P \quad (10)$$

6.2 Result of the flexibility analysis

The model $\mathrm{SBC3^a}$ is used to test the value of flexibility, since only it is capable of finding the optimal solutions for all of the first 24 instances.

Table 11 indicates the impact of constraint (2) on the total cost. We can see from the Table that there is no change in the total cost when the slot is allowed to remain empty.

Therefore, constraint (2) has no impact on the total cost for the instances we test. However, in some situation, constraint (2) does impact the total cost. For example, each plate has 4 slots. We have an order that includes 2 designs: design 1 has a demand of 1000; design 2 has a demand of 2000. A solution with no overproduction would be to have 1 slot for design 1, 2 slots for design 2, and 1 slot for empty with a rotation of 1000. If we enforce to use all the slots on the plate, it will certainly give rise to the overproduction costs. The reason for the small impact of constraint (2) to the total cost is that each instance we test has at least 5 designs and the set up cost for a new plate is high compared to the overproduction cost. In order to minimize the total cost, all the designs are allocated to the minimal number of required plates, which reduces the chance of leaving the slot empty.

Table 11: The impact of constraint (2) on the total cost

	$\mathbf{SBC3^a}$		$\mathbf{SBC3^a}$+ constraint (2) changed	
N	B	T	B	T
5	762	0.15	762	0.19
10	1301	2.90	1301	3.33
15	1915	219	1915	291

Table 12 indicates the impact of constraint (5) on the total cost. When the maximum number of colors that can be allocated to a single plate is increased from 2 to 3, the average value of the optimal solutions decreases by $124 (6.5%) for instances where the number of designs is 15; the average value of the optimal solutions decreases by $31 (2.4%) for 10 designs; and the average value remains the same for instances with 5 designs. When we increase the maximum number of colors that can be allocated to a single plate from 3 to 4, the average value of the optimal solutions for instances where the number of designs is 5 or 10 stays the same as when the maximum number of colors allocated to a single plate is 3. The average value of the optimal solutions for instances of 15 designs decreases by $31 (1.7%). Continuing to increase the maximum number of colors that can be allocated to a plate one by one until the upper limit of 7 is reached produces no more change in the value of the optimal solution. We observe diminishing marginal improvements when we relax this constraint. When the maximum number of colors is relaxed from 2 to 3, the number of required plates for each instance will considerably decrease, which also leads to a decrease of the set up cost. When the maximum number of colors per plate continue increase, the influence on the number of required plates will become less important, which also explains the diminishing marginal improvements when we relax this constraint.

Table 12: The impact of constraint (5) on the total cost

	Max 2 colors		Max 3 colors		Max 4 colors		Max 5 colors		Max 6 colors		Max 7 colors	
N	B	T	B	T	B	T	B	T	B	T	B	T
5	762	0.15	762	0.13	762	0.13	762	0.14	762	0.13	762	0.13
10	1301	2.90	1270	3.24	1270	3.17	1270	3.50	1270	3.27	1270	3.23
15	1915	219	1791	458	1760	398	1760	302	1760	303	1760	302

Table 13 indicates the impact of constraint (7), the white border constraint, on the total cost. We can see from Table 13 that the impact of constraint (7) is small. Deleting this constraint from the model, the average value of the optimal solution for 5 designs decreases by $11 (1.4%); for 15 designs, the value decreases by only $1 (0.1%). In the Appendix, Exhibit 8 presents detailed results showing that only 2 instances out of 24 change the value of the optimal solution. The impact of constraint (7) is small because each instance has at least 5 designs, which means that there are many combinations of designs to increase the possibility of satisfying the white border constraint.

Table 13: The impact of constraint (7) on the total cost

	$\mathbf{SBC3^a}$		$\mathbf{SBC3^a}$ without constraint (7)	
N	B	T	B	T
5	762	0.15	751	0.11
10	1301	2.90	1301	3.21
15	1915	219	1914	154

Table 14 indicates the impact of constraint (8), which holds that each plate can contain at most one slot with a standard design, on the total cost. We can see from Table 14 that this constraint has more influence on those instances where the number of designs is small. When we delete constraint (8) from the model, the average value of the optimal solutions for instances of 5 designs decreases by $91 (11.9%); for 10 designs the average value decreases by $14 (1.1%); and for 15 designs, the average value remains constant. Why the constraint (8) has more impact on instances with small number of designs? One of reasons we think is that the small number of designs leads to the small number of combinations of designs, which also makes it more likely to fill the slot with a standard design since the overproduction cost of a standard design is lower than the overproduction cost of a customer specific design.

Table 14: The impact of constraint (8) on the total cost

	$\mathbf{SBC3^a}$		$\mathbf{SBC3^a}$ without constraint (8)	
N	B	T	B	T
5	762	0.15	671	0.16
10	1301	2.90	1287	4.12
15	1915	219	1915	382

Table 15 indicates the impact of constraints (9) and (10) on the total cost. We can see from Table 15 that if we delete the non-split constraint from the model, the value of the best feasible solutions does not significantly change; the complexity of the model, however, rises considerably. The average value of the optimal solutions for instances with 5 designs remains the same. The average value for 10 designs decreases by $21 (1.6%), while for 15 designs, the value increases by $11 (0.6%). The increase in this last case is explained by the fact that for 4 instances out of 8 no optimal solution was found within the time limit, which also explains why the required CPU time rises sharply.

Table 15: The impact of constraints (9) and (10) on the total cost

	SBC3[a]		**SBC3[a]** without constraint (9) and (10)	
N	B	T	B	T
5	762	0.15	762	0.19
10	1301	2.90	1280	11.20
15	1915	219	1926	1068

Chapter 7 Conclusion

This thesis has improved the two MIP models presented in Baumann et al. ($2014^a, 2014^b$) by adding new symmetry breaking constraints to solve the issues of symmetry due to both identical slots and identical plates. Computational results indicate that the three types of symmetry breaking constraints proposed (SBC1, SBC2, SBC3) clearly perform better than the symmetry breaking constraints ($\mathrm{SBC0}$) presented in Baumann et al. ($2014^a, 2014^b$). SBC1 in particular has been demonstrated to be superior to other symmetry breaking constraints with respect to finding both optimal and feasible solutions within the time limit. Although there is no formulation which wins on all the criteria and instances, $\mathrm{SBC1}^b$ has an overall very good performance. Computational results also indicate that fixing a variable can reduce the required CPU time when the issue of symmetry is significant. However, the effect of fixing a variable is not clear when the model already employs other symmetry breaking constraints. The symmetry breaking constraints presented in Baumann et al.($2014^a, 2014^b$) use the presence or absence of a design in a slot on a plate as criterion for eliminating symmetry. We also propose to use the presence or absence of a color on the plate as this criterion instead. The alternative formulation, augmented with SBC1, based on this new criterion ($\mathrm{SBC1}^{\mathrm{c}}$) performs better than all other models for instances where the number of designs is 30. The basic formulation, augmented with SBC1, based on this new criterion ($\mathrm{SBC1}^{\mathrm{d}}$) finds feasible solutions for all the instances while other formulations augmented with symmetry breaking constraints cannot find feasible solutions for all the instances. This thesis studies further the printing planning problem, analysing the impact of constraints on total costs. Color constraints have been found to have the biggest impact on the total cost, especially when the number of designs is large. Constraint (8) has also a big impact for instances with a small number of designs. The non-split constraint exercises considerable influence on the complexity of the printing planning problem.

The research presented in this thesis also has its limitations. In our thesis, we use an existing data set used by Baumann et al. (2014^{a}). According to the results we obtained, all the formulations we proposed begin to have difficulty in finding feasible solutions when the number of design is 30 or more and the formulations augmented with symmetry breaking constraints only find a few of feasible solutions when the number of design is 50. We notice that in the data set, after the number of design is 30, the number of design turns into 50. Therefore, we lack the results of how well the symmetry breaking constraints perform when number of design is 35, 40, and 45. It would be interesting to generate the new instances with 35, 40, and 45 designs and test it. Furthermore, we set the time limit to 1800 seconds. It would be interesting to observe how well the symmetry breaking constraints perform if we set a longer time limit.

An interesting area for future research is the exploration of other symmetry issues in extensions of the printing planning problem. As mentioned in previous chapters, the issue of symmetry is a very common phenomenon in the printing industry. For example, printing manufacturers use several identical machines to produce the napkin pouches, while workers allocated to the different machines are also identical. Another extension would be to look at multi-periods, where the printing problem can include the possibility of holding inventory. Then the printing planning problem can incorporate more costs such as inventory cost. Another interesting line of research would be to reduce the symmetry issue by other approaches such as problem reformulation with Dantzig-Wolfe decomposition and developing further heuristic or optimal algorithms.

Appendix

We will give the detailed results for all the formulations tested. There are 56 instances tested. In the following Exhibits, N, WR, CR and DR refer to the number of designs, the white border ratio, the color ratio and the demand ratio, as explained in Section 5.1. B, G and T refer to the value of the best feasible solution (B), the MIP gap (G) and the required CPU time (T), which we also explained in Section 5.1.

Exhibit 1：Detailed results of the basic formulation, $SBC0^a$, $SBC1^a$, $SBC2^a$, and $SBC3^a$

					Basic formulation			**$SBC0^a$**			**$SBC1^a$**			**$SBC2^a$**			**$SBC3^a$**		
Instance	N	WR	CR	DR	B	G	T	B	G	T	B	G	T	B	G	T	B	G	T
1	5	0.33	0.15	0.4	731	0.0%	0.69	731	0.0%	0.22	731	0.0%	0.19	731	0.0%	0.05	731	0.0%	0.21
2	5	0.33	0.15	0.2	665	0.0%	0.84	665	0.0%	0.30	665	0.0%	0.23	665	0.0%	0.13	665	0.0%	0.30
3	5	0.33	0.3	0.4	731	0.0%	1.19	731	0.0%	0.22	731	0.0%	0.14	731	0.0%	0.06	731	0.0%	0.06
4	5	0.33	0.3	0.2	1079	0.0%	0.92	1079	0.0%	0.22	1079	0.0%	0.03	1079	0.0%	0.16	1079	0.0%	0.03
5	5	0.66	0.15	0.4	731	0.0%	0.63	731	0.0%	0.06	731	0.0%	0.19	731	0.0%	0.06	731	0.0%	0.23

6	5	0.66	0.15	0.2	611	0.0%	0.59	611	0.0%	0.30	611	0.0%	0.17	611	0.0%	0.20	611	0.0%	0.09
7	5	0.66	0.3	0.4	731	0.0%	1.14	731	0.0%	0.22	731	0.0%	0.23	731	0.0%	0.20	731	0.0%	0.17
8	5	0.66	0.3	0.2	817	0.0%	0.86	817	0.0%	0.22	817	0.0%	0.27	817	0.0%	0.22	817	0.0%	0.13
9	10	0.33	0.15	0.4	1355	19.7%	1801	1355	0.0%	5.52	1355	0.0%	2.39	1355	0.0%	2.25	1355	0.0%	2.17
10	10	0.33	0.15	0.2	1311	7.3%	1800	1311	0.0%	31.97	1311	0.0%	2.11	1311	0.0%	3.15	1311	0.0%	3.70
11	10	0.33	0.3	0.4	1202	0.0%	576	1202	0.0%	8.17	1202	0.0%	1.75	1202	0.0%	2.74	1202	0.0%	2.14
12	10	0.33	0.3	0.2	1376	21.5%	1801	1376	0.0%	8.17	1376	0.0%	3.00	1376	0.0%	4.30	1376	0.0%	5.83
13	10	0.66	0.15	0.4	1223	2.8%	1800	1223	0.0%	13.10	1223	0.0%	2.35	1223	0.0%	6.36	1223	0.0%	2.80
14	10	0.66	0.15	0.2	1243	7.2%	1800	1243	0.0%	8.83	1243	0.0%	3.68	1243	0.0%	3.52	1243	0.0%	1.95
15	10	0.66	0.3	0.4	1334	0.0%	1306	1334	0.0%	6.86	1334	0.0%	2.26	1334	0.0%	2.52	1334	0.0%	2.50
16	10	0.66	0.3	0.2	1363	18.4%	1801	1363	0.0%	12.09	1363	0.0%	2.56	1363	0.0%	4.42	1363	0.0%	2.11
17	15	0.33	0.15	0.4	1904	14.9%	1803	1904	0.0%	1184	1904	0.0%	324	1904	0.0%	344	1904	0.0%	261
18	15	0.33	0.15	0.2	2083	22.2%	1803	2051	8.6%	1800	2121	22.4%	1800	2051	0.0%	835	2051	0.0%	727
19	15	0.33	0.3	0.4	2018	19.7%	1804	2018	0.0%	254	2018	0.0%	68.94	2018	0.0%	177	2018	0.0%	95.29
20	15	0.33	0.3	0.2	2028	20.1%	1800	2028	0.0%	444	2028	0.0%	235	2028	0.0%	561	2028	0.0%	213
21	15	0.66	0.15	0.4	1676	3.4%	1801	1676	0.0%	195	1676	0.0%	48.73	1676	0.0%	128	1676	0.0%	36.27
22	15	0.66	0.15	0.2	1803	10.1%	1803	1803	0.0%	283	1803	0.0%	345	1803	0.0%	164	1803	0.0%	143
23	15	0.66	0.3	0.4	1819	10.9%	1802	1819	0.0%	303	1819	0.0%	246	1819	0.0%	311	1819	0.0%	146
24	15	0.66	0.3	0.2	2023	19.9%	1801	2023	0.0%	905	2023	0.0%	131	2023	0.0%	172	2023	0.0%	134

25	20	0.33	0.15	0.4	2053	21.1%	1800	2464	34.2%	1800	2053	0.0%	527	2053	0.0%	843	3329	52.7%	1800
26	20	0.33	0.15	0.2	2616	38.1%	1800	2849	43.1%	1800	2746	39.8%	1800	2564	31.1%	1800	2616	21.6%	1802
27	20	0.33	0.3	0.4	2888	43.9%	1800	3380	52.1%	1800	3659	41.0%	1800	3016	28.4%	1800	3393	36.3%	1800
28	20	0.33	0.3	0.2	3010	46.2%	1800	3516	38.6%	1800	3865	58.1%	1800	3039	30.0%	1800	3448	55.0%	1800
29	20	0.66	0.15	0.4	2173	25.4%	1800	2998	46.0%	1800	2173	0.0%	1284	2173	0.0%	603	2173	0.0%	1439
30	20	0.66	0.15	0.2	2734	40.7%	1800	3426	55.0%	1800	2734	23.5%	1800	2820	43.4%	1800	2520	35.7%	1800
31	20	0.66	0.3	0.4	2588	37.4%	1800	3081	47.4%	1800	2828	23.6%	1800	3129	44.3%	1800	2965	27.2%	1800
32	20	0.66	0.3	0.2	3094	47.6%	1800	3408	49.6%	1800	3040	29.0%	1800	3254	47.9%	1800	3266	51.2%	1800
33	25	0.33	0.15	0.4	3162	31.7%	1800	4420	51.1%	1800	3825	43.5%	1800	3348	35.5%	1800	3164	31.7%	1800
34	25	0.33	0.15	0.2	3500	44.9%	1800	3571	39.5%	1800	3567	45.5%	1800	4542	52.4%	1800	4178	53.8%	1800
35	25	0.33	0.3	0.4	3681	47.6%	1800	4116	53.1%	1800	6128	68.2%	1800	4728	59.1%	1800	3833	43.6%	1800
36	25	0.33	0.3	0.2	3831	43.6%	1801	5176	62.7%	1800	4123	53.0%	1800	5047	61.8%	1800	5150	58.1%	1800
37	25	0.66	0.15	0.4	3091	30.1%	1801	3565	39.4%	1800	3902	49.4%	1800	4696	58.9%	1802	3160	31.6%	1800
38	25	0.66	0.15	0.2	3192	39.6%	1806	4677	58.8%	1800	3383	36.2%	1800	4446	56.7%	1800	3200	32.5%	1800
39	25	0.66	0.3	0.4	4163	53.7%	1800	3812	43.3%	1800	4677	53.8%	1800	5079	57.9%	1800	4025	45.6%	1800
40	25	0.66	0.3	0.2	3706	48.0%	1800	6420	70.0%	1800	5263	59.0%	1800	7522	74.0%	1800	4040	46.5%	1800
41	30	0.33	0.15	0.4	3940	41.3%	1800	5558	58.4%	1800	5847	60.4%	1800	5434	57.4%	1800	5134	47.4%	1800
42	30	0.33	0.15	0.2	4323	46.5%	1800	7468	69.0%	1800	15158	84.7%	1800	6195	62.7%	1800	6008	61.5%	1800
43	30	0.33	0.3	0.4	5474	50.7%	1800	7211	62.6%	1800	6877	60.7%	1800	16384	85.0%	1800	16384	84.9%	1800

44	30	0.33	0.3	0.2	4569	49.3%	1800	na	na	1800	6137	62.3%	1800	15942	85.5%	1800	5555	54.2%	1800
45	30	0.66	0.15	0.4	3941	41.3%	1800	10763	78.5%	1800	5637	58.5%	1800	4657	50.4%	1800	64198	96.4%	1800
46	30	0.66	0.15	0.2	4497	48.5%	1800	5784	53.3%	1800	4494	48.5%	1800	5895	60.8%	1800	5929	61.0%	1800
47	30	0.66	0.3	0.4	4453	48.0%	1800	10326	77.6%	1800	5890	57.6%	1800	15245	84.8%	1800	5948	61.1%	1800
48	30	0.66	0.3	0.2	5147	55.0%	1800	na	na	1800	7400	68.7%	1800	6062	61.9%	1800	7661	69.8%	1800
49	50	0.33	0.15	0.4	12215	64.6%	1800	11650	62.9%	1800	18886	78.1%	1800	na	na	1800	na	na	1800
50	50	0.33	0.15	0.2	11115	65.3%	1800	na	na	1800	18668	79.3%	1800	25988	85.5%	1800	22557	82.9%	1801
51	50	0.33	0.3	0.4	15562	75.2%	1800	na	na	1800	108225	96.4%	1800	na	na	1800	na	na	1800
52	50	0.33	0.3	0.2	15176	71.5%	1800	na	na	1800	na	na	1800	na	na	1800	na	na	1800
53	50	0.66	0.15	0.4	13914	72.3%	1800	na	na	1800	17261	77.7%	1800	na	na	1800	na	na	1800
54	50	0.66	0.15	0.2	8172	52.8%	1800	12887	70.1%	1800	17392	77.8%	1800	22451	82.8%	1800	111656	96.5%	1800
55	50	0.66	0.3	0.4	21231	81.8%	1800	na	na	1800	21950	82.4%	1800	27057	85.8%	1800	na	na	1800
56	50	0.66	0.3	0.2	15724	75.5%	1800	na	na	1800	21739	82.3%	1800	22186	82.7%	1800	na	na	1800

Exhibit 2: Detailed results of alternative formulation $SBC0^b$, $SBC1^b$, and $SBC3^b$

					Alternative formulation			$SBC0^b$			$SBC1^b$			$SBC3^b$		
Instance	N	WR	CR	DR	B	G	T	B	G	T	B	G	T	B	G	T
1	5	0.33	0.15	0.4	731	0.0%	0.14	731	0.0%	0.19	731	0.0%	0.15	731	0.0%	0.11
2	5	0.33	0.15	0.2	665	0.0%	0.14	665	0.0%	0.17	665	0.0%	0.13	665	0.0%	0.16
3	5	0.33	0.3	0.4	731	0.0%	0.13	731	0.0%	0.16	731	0.0%	0.11	731	0.0%	0.09
4	5	0.33	0.3	0.2	1079	0.0%	0.09	1079	0.0%	0.14	1079	0.0%	0.10	1079	0.0%	0.08
5	5	0.66	0.15	0.4	731	0.0%	0.11	731	0.0%	0.11	731	0.0%	0.09	731	0.0%	0.11
6	5	0.66	0.15	0.2	611	0.0%	0.14	611	0.0%	0.16	611	0.0%	0.19	611	0.0%	0.11
7	5	0.66	0.3	0.4	731	0.0%	0.14	731	0.0%	0.13	731	0.0%	0.17	731	0.0%	0.13
8	5	0.66	0.3	0.2	817	0.0%	0.16	817	0.0%	0.17	817	0.0%	0.23	817	0.0%	0.14
9	10	0.33	0.15	0.4	1355	0.0%	0.92	1355	0.0%	1.36	1355	0.0%	1.49	1355	0.0%	0.86
10	10	0.33	0.15	0.2	1311	0.0%	1.50	1311	0.0%	1.52	1311	0.0%	0.92	1311	0.0%	1.53
11	10	0.33	0.3	0.4	1202	0.0%	1.58	1202	0.0%	1.72	1202	0.0%	1.12	1202	0.0%	1.08
12	10	0.33	0.3	0.2	1376	0.0%	2.66	1376	0.0%	3.18	1376	0.0%	1.78	1376	0.0%	2.03
13	10	0.66	0.15	0.4	1223	0.0%	1.03	1223	0.0%	1.63	1223	0.0%	1.19	1223	0.0%	1.14

14	10	0.66	0.15	0.2	1243	0.0%	1.14	1243	0.0%	1.91	1243	0.0%	1.38	1243	0.0%	1.91
15	10	0.66	0.3	0.4	1334	0.0%	1.44	1334	0.0%	2.70	1334	0.0%	0.78	1334	0.0%	1.08
16	10	0.66	0.3	0.2	1363	0.0%	3.43	1363	0.0%	1.79	1363	0.0%	1.39	1363	0.0%	1.77
17	15	0.33	0.15	0.4	1904	0.0%	92.89	1904	0.0%	156	1904	0.0%	26.98	1904	0.0%	34.46
18	15	0.33	0.15	0.2	2051	0.0%	265	2051	0.0%	193	2051	0.0%	65.74	2051	0.0%	62.46
19	15	0.33	0.3	0.4	2018	0.0%	51.64	2018	0.0%	78.92	2018	0.0%	11.64	2018	0.0%	14.39
20	15	0.33	0.3	0.2	2028	0.0%	141	2028	0.0%	140	2028	0.0%	22.00	2028	0.0%	35.11
21	15	0.66	0.15	0.4	1676	0.0%	29.48	1676	0.0%	19.50	1676	0.0%	4.70	1676	0.0%	4.38
22	15	0.66	0.15	0.2	1803	0.0%	218	1803	0.0%	146	1803	0.0%	32.77	1803	0.0%	43.47
23	15	0.66	0.3	0.4	1819	0.0%	12.51	1819	0.0%	89.60	1819	0.0%	8.24	1819	0.0%	6.25
24	15	0.66	0.3	0.2	2023	0.0%	156	2023	0.0%	228	2023	0.0%	19.93	2023	0.0%	25.42
25	20	0.33	0.15	0.4	2053	0.0%	43.91	2053	0.0%	30.84	2053	0.0%	15.33	2053	0.0%	8.70
26	20	0.33	0.15	0.2	2447	11.7%	1800	2540	15.0%	1805	2564	15.1%	1800	2564	15.7%	1806
27	20	0.33	0.3	0.4	2676	19.3%	1800	2676	19.3%	1800	2676	0.0%	243	2676	0.0%	236
28	20	0.33	0.3	0.2	2989	27.7%	1800	2866	24.6%	1804	2866	5.8%	1800	2866	5.8%	1800
29	20	0.66	0.15	0.4	2173	0.0%	119	2173	0.0%	158	2173	0.0%	15.10	2173	0.0%	25.18
30	20	0.66	0.15	0.2	2427	11.0%	1805	2520	14.3%	1800	2361	7.1%	1801	2582	16.3%	1800
31	20	0.66	0.3	0.4	2416	10.3%	1801	2416	10.6%	1800	2416	0.0%	465	2416	0.0%	424

32	20	0.66	0.3	0.2	2786	22.5%	1800	2675	19.3%	1800	2675	0.0%	378	2762	14.0%	1800
33	25	0.33	0.15	0.4	2981	22.1%	1801	2981	20.8%	1801	2919	7.5%	1801	2919	7.5%	1801
34	25	0.33	0.15	0.2	3016	21.9%	1810	3442	37.2%	1800	3053	11.5%	1800	3016	10.5%	1801
35	25	0.33	0.3	0.4	3369	32.8%	1811	3864	30.1%	1800	3543	23.8%	1800	3381	20.1%	1800
36	25	0.33	0.3	0.2	3943	45.2%	1800	4072	33.7%	1800	3505	23.0%	1800	3646	25.9%	1800
37	25	0.66	0.15	0.4	2974	27.4%	1801	3188	32.2%	1800	2974	9.2%	1801	2974	9.2%	1800
38	25	0.66	0.15	0.2	2956	26.9%	1800	3266	33.9%	1800	3089	23.4%	1800	3016	21.7%	1800
39	25	0.66	0.3	0.4	3771	28.4%	1800	4046	46.6%	1800	3616	25.3%	1800	3533	23.6%	1800
40	25	0.66	0.3	0.2	3736	40.3%	1800	na	na	1800	3677	26.6%	1800	3933	31.3%	1800
41	30	0.33	0.15	0.4	4095	40.7%	1800	4412	38.8%	1800	3704	20.8%	1800	3723	27.5%	1800
42	30	0.33	0.15	0.2	3878	37.6%	1801	na	na	1800	4477	43.5%	1800	4042	35.4%	1800
43	30	0.33	0.3	0.4	5402	50.0%	1800	na	na	1800	4552	28.8%	1800	4762	38.7%	1800
44	30	0.33	0.3	0.2	5486	51.4%	1800	5237	48.4%	1800	4504	44.2%	1800	4757	44.8%	1800
45	30	0.66	0.15	0.4	3716	32.4%	1815	4075	43.2%	1809	3625	27.6%	1805	3745	27.9%	1800
46	30	0.66	0.15	0.2	3627	25.6%	1801	4536	49.0%	1800	4039	33.2%	1800	3998	32.5%	1800
47	30	0.66	0.3	0.4	4255	45.3%	1814	4975	45.7%	1800	4569	41.8%	1800	4575	41.0%	1800
48	30	0.66	0.3	0.2	4878	51.2%	1815	na	na	1800	5144	51.8%	1800	4717	47.7%	1800
49	50	0.33	0.15	0.4	21730	80.9%	1800	na	na	1800	16544	73.9%	1800	na	na	1800

50	50	0.33	0.15	0.2	9290	58.5%	1800	na	na	1800	9404	59.0%	1800	na	na	1800
51	50	0.33	0.3	0.4	17341	77.8%	1801	na	na	1800	15679	75.2%	1800	na	na	1800
52	50	0.33	0.3	0.2	20177	80.2%	1800	na	na	1800	na	na	1800	na	na	1800
53	50	0.66	0.15	0.4	8428	54.2%	1800	na	na	1800	8677	55.5%	1800	11634	66.8%	1800
54	50	0.66	0.15	0.2	12287	68.6%	1800	na	na	1800	10446	63.1%	1800	na	na	1800
55	50	0.66	0.3	0.4	11791	67.3%	1800	na	na	1800	11131	65.3%	1800	na	na	1800
56	50	0.66	0.3	0.2	na	na	1800	na	na	1800	na	na	1800	na	na	1800

Exhibit 3: Detailed results of basic formulation with $X_{111} = 1$ and $SBC1^a - 3^a$ with $X_{111} = 1$

					Basic formulation + $X_{111} = 1$			$SBC1^a + X_{111} = 1$			$SBC2^a + X_{111} = 1$			$SBC3^a + X_{111} = 1$		
Instance	N	WR	CR	DR	B	G	T	B	G	T	B	G	T	B	G	T
1	5	0.33	0.15	0.4	731	0.0%	1.33	731	0.0%	0.25	731	0.0%	0.11	731	0.0%	0.14
2	5	0.33	0.15	0.2	665	0.0%	0.36	665	0.0%	0.20	665	0.0%	0.13	665	0.0%	0.14
3	5	0.33	0.3	0.4	731	0.0%	0.30	731	0.0%	0.14	731	0.0%	0.05	731	0.0%	0.13
4	5	0.33	0.3	0.2	1079	0.0%	0.17	1079	0.0%	0.25	1079	0.0%	0.03	1079	0.0%	0.27
5	5	0.66	0.15	0.4	731	0.0%	0.11	731	0.0%	0.14	731	0.0%	0.08	731	0.0%	0.11
6	5	0.66	0.15	0.2	611	0.0%	0.63	611	0.0%	0.14	611	0.0%	0.16	611	0.0%	0.14
7	5	0.66	0.3	0.4	731	0.0%	0.30	731	0.0%	0.31	731	0.0%	0.08	731	0.0%	0.19
8	5	0.66	0.3	0.2	817	0.0%	0.36	817	0.0%	0.30	817	0.0%	0.09	817	0.0%	0.20
9	10	0.33	0.15	0.4	1355	10.8%	1800	1355	0.0%	3.23	1355	0.0%	4.58	1355	0.0%	2.47
10	10	0.33	0.15	0.2	1311	0.0%	228	1311	0.0%	8.24	1311	0.0%	2.33	1311	0.0%	2.52
11	10	0.33	0.3	0.4	1202	0.0%	76.91	1202	0.0%	2.72	1202	0.0%	2.58	1202	0.0%	1.28
12	10	0.33	0.3	0.2	1376	14.4%	1800	1376	0.0%	4.07	1376	0.0%	5.57	1376	0.0%	5.92

13	10	0.66	0.15	0.4	1223	0.0%	290	1223	0.0%	3.36	1223	0.0%	3.78	1223	0.0%	1.92
14	10	0.66	0.15	0.2	1243	0.0%	757	1243	0.0%	4.11	1243	0.0%	3.02	1243	0.0%	3.88
15	10	0.66	0.3	0.4	1334	0.0%	252	1334	0.0%	2.10	1334	0.0%	1.91	1334	0.0%	1.38
16	10	0.66	0.3	0.2	1363	3.2%	1800	1363	0.0%	10.95	1363	0.0%	5.16	1363	0.0%	1.86
17	15	0.33	0.15	0.4	1904	14.9%	1801	1904	0.0%	162	1904	0.0%	209	1904	0.0%	302
18	15	0.33	0.15	0.2	2051	21.0%	1800	2051	0.0%	671	2051	0.0%	461	2051	0.0%	678
19	15	0.33	0.3	0.4	2018	19.7%	1800	2018	0.0%	184	2018	0.0%	240	2018	0.0%	32.50
20	15	0.33	0.3	0.2	2028	20.1%	1800	2028	0.0%	251	2028	0.0%	1293	2028	0.0%	105
21	15	0.66	0.15	0.4	1676	3.4%	1801	1676	0.0%	95.90	1676	0.0%	211	1676	0.0%	74.28
22	15	0.66	0.15	0.2	1803	10.1%	1803	1803	0.0%	360	1803	0.0%	175	1803	0.0%	80.63
23	15	0.66	0.3	0.4	1819	10.9%	1808	1819	0.0%	106	1819	0.0%	190	1819	0.0%	65.89
24	15	0.66	0.3	0.2	2023	19.9%	1804	2023	0.0%	192	2023	0.0%	239	2023	0.0%	353
25	20	0.33	0.15	0.4	2053	21.1%	1800	2053	9.7%	1800	2464	36.1%	1800	2053	0.0%	1685
26	20	0.33	0.15	0.2	2616	38.1%	1800	2616	30.2%	1800	2687	32.8%	1800	2687	39.4%	1800
27	20	0.33	0.3	0.4	2971	45.5%	1800	3023	28.6%	1800	3646	55.6%	1800	2676	18.9%	1800
28	20	0.33	0.3	0.2	3105	47.8%	1800	3056	29.3%	1800	3335	53.4%	1800	3355	35.6%	1800
29	20	0.66	0.15	0.4	2173	25.4%	1800	2173	0.0%	751	2173	0.0%	1073	2173	0.0%	630
30	20	0.66	0.15	0.2	2520	35.7%	1800	2973	45.1%	1800	2937	44.5%	1800	2734	35.0%	1800

31	20	0.66	0.3	0.4	2588	37.4%	1800	2710	20.3%	1800	2828	23.6%	1800	2747	19.3%	1800
32	20	0.66	0.3	0.2	3092	47.6%	1800	3848	44.2%	1800	3428	52.7%	1800	3393	42.3%	1800
33	25	0.33	0.15	0.4	3255	33.6%	1800	3541	39.0%	1800	3550	44.3%	1800	3458	37.5%	1800
34	25	0.33	0.15	0.2	3470	37.7%	1800	4436	51.3%	1800	3751	42.4%	1800	4395	55.1%	1800
35	25	0.33	0.3	0.4	3848	43.9%	1800	3699	41.6%	1800	4597	55.8%	1800	5188	62.8%	1800
36	25	0.33	0.3	0.2	4081	50.6%	1800	5076	61.9%	1800	4109	52.9%	1800	4103	52.7%	1800
37	25	0.66	0.15	0.4	3341	35.4%	1800	3709	41.8%	1800	3709	41.8%	1800	3895	49.2%	1800
38	25	0.66	0.15	0.2	3273	40.8%	1800	4118	48.7%	1800	10119	81.0%	1800	3567	39.4%	1800
39	25	0.66	0.3	0.4	3875	44.3%	1800	5894	63.4%	1800	4563	52.7%	1800	4883	54.3%	1800
40	25	0.66	0.3	0.2	3596	42.3%	1800	5492	64.8%	1800	4556	57.8%	1800	3871	44.2%	1800
41	30	0.33	0.15	0.4	4183	44.7%	1800	6398	63.8%	1800	5354	56.8%	1800	5428	57.4%	1800
42	30	0.33	0.15	0.2	4651	50.2%	1800	5087	46.9%	1800	9763	76.6%	1800	5403	54.1%	1800
43	30	0.33	0.3	0.4	5248	48.6%	1800	7382	63.4%	1800	9953	75.5%	1800	15948	84.4%	1800
44	30	0.33	0.3	0.2	4659	50.3%	1800	6346	63.5%	1800	9169	74.8%	1800	5625	57.5%	1800
45	30	0.66	0.15	0.4	3971	41.7%	1800	6555	63.8%	1800	3878	40.6%	1800	4650	41.9%	1800
46	30	0.66	0.15	0.2	4160	44.4%	1800	5889	60.4%	1800	5651	59.2%	1800	5593	58.6%	1800
47	30	0.66	0.3	0.4	4243	44.0%	1800	6234	62.9%	1800	6231	62.9%	1800	7922	70.8%	1800
48	30	0.66	0.3	0.2	4841	52.2%	1800	6552	62.6%	1800	5732	59.6%	1800	5623	69.9%	1800

49	50	0.33	0.15	0.4	8910	51.5%	1800	21142	80.4%	1800	na	na	1800	na	na	1800
50	50	0.33	0.15	0.2	8577	55.0%	1800	18298	78.9%	1800	na	na	1800	na	na	1800
51	50	0.33	0.3	0.4	17778	78.3%	1800	na	na	1800	na	na	1800	na	na	1800
52	50	0.33	0.3	0.2	14777	70.8%	1800	na	na	1800	na	na	1800	na	na	1800
53	50	0.66	0.15	0.4	11846	67.4%	1800	na	na	1800	na	na	1800	na	na	1800
54	50	0.66	0.15	0.2	12161	68.3%	1800	21157	81.8%	1802	111656	96.6%	1800	na	na	1800
55	50	0.66	0.3	0.4	9154	57.9%	1800	112043	96.6%	1800	26365	85.4%	1800	na	na	1800
56	50	0.66	0.3	0.2	17675	78.2%	1800	16282	76.3%	1800	23484	83.7%	1800	110696	97.9%	1800

Exhibit 4: Detailed results of $SBC0^c$, $SBC1^c$, and $SBC3^c$

					$SBC0^c$			$SBC1^c$			$SBC3^c$		
Instance	N	WR	CR	DR	B	G	T	B	G	T	B	G	T
1	5	0.33	0.15	0.4	731	0.0%	0.19	731	0.0%	0.16	731	0.0%	0.18
2	5	0.33	0.15	0.2	665	0.0%	0.17	665	0.0%	0.14	665	0.0%	0.15
3	5	0.33	0.3	0.4	731	0.0%	0.17	731	0.0%	0.15	731	0.0%	0.16
4	5	0.33	0.3	0.2	1079	0.0%	0.13	1079	0.0%	0.11	1079	0.0%	0.13
5	5	0.66	0.15	0.4	731	0.0%	0.11	731	0.0%	0.13	731	0.0%	0.14
6	5	0.66	0.15	0.2	611	0.0%	0.14	611	0.0%	0.14	611	0.0%	0.14
7	5	0.66	0.3	0.4	731	0.0%	0.14	731	0.0%	0.14	731	0.0%	0.15
8	5	0.66	0.3	0.2	817	0.0%	0.13	817	0.0%	0.17	817	0.0%	0.20
9	10	0.33	0.15	0.4	1355	0.0%	0.93	1355	0.0%	0.97	1355	0.0%	0.97
10	10	0.33	0.15	0.2	1311	0.0%	1.59	1311	0.0%	1.67	1311	0.0%	2.92
11	10	0.33	0.3	0.4	1202	0.0%	1.17	1202	0.0%	0.92	1202	0.0%	1.11
12	10	0.33	0.3	0.2	1376	0.0%	2.82	1376	0.0%	2.49	1376	0.0%	2.51
13	10	0.66	0.15	0.4	1223	0.0%	1.03	1223	0.0%	1.22	1223	0.0%	0.91

14	10	0.66	0.15	0.2	1243	0.0%	1.13	1243	0.0%	1.48	1243	0.0%	1.86
15	10	0.66	0.3	0.4	1334	0.0%	1.98	1334	0.0%	1.79	1334	0.0%	1.37
16	10	0.66	0.3	0.2	1363	0.0%	2.46	1363	0.0%	1.74	1363	0.0%	1.32
17	15	0.33	0.15	0.4	1904	0.0%	64.73	1904	0.0%	30.96	1904	0.0%	69.43
18	15	0.33	0.15	0.2	2051	0.0%	96.25	2051	0.0%	80.28	2051	0.0%	54.27
19	15	0.33	0.3	0.4	2018	0.0%	18.32	2018	0.0%	55.13	2018	0.0%	46.42
20	15	0.33	0.3	0.2	2028	0.0%	52.10	2028	0.0%	30.36	2028	0.0%	63.16
21	15	0.66	0.15	0.4	1676	0.0%	10.69	1676	0.0%	13.04	1676	0.0%	9.16
22	15	0.66	0.15	0.2	1803	0.0%	110	1803	0.0%	130	1803	0.0%	93.45
23	15	0.66	0.3	0.4	1819	0.0%	19.26	1819	0.0%	10.44	1819	0.0%	13.97
24	15	0.66	0.3	0.2	2023	0.0%	98.03	2023	0.0%	97.24	2023	0.0%	104
25	20	0.33	0.15	0.4	2053	0.0%	47.41	2053	0.0%	18.10	2053	0.0%	23.17
26	20	0.33	0.15	0.2	2564	15.7%	1819	2564	15.7%	1807	2564	15.7%	1804
27	20	0.33	0.3	0.4	2676	0.0%	380	2676	0.0%	471	2676	0.0%	860
28	20	0.33	0.3	0.2	2940	26.5%	1800	2940	26.5%	1800	2866	5.8%	1800
29	20	0.66	0.15	0.4	2173	0.0%	40.78	2173	0.0%	55.95	2173	0.0%	48.45
30	20	0.66	0.15	0.2	2361	8.5%	1808	2520	14.3%	1802	2405	8.8%	1800
31	20	0.66	0.3	0.4	2416	0.0%	366	2416	0.0%	538	2416	0.0%	281

32	20	0.66	0.3	0.2	2675	11.6%	1800	2675	0.0%	1424	3130	31.0%	1806
33	25	0.33	0.15	0.4	2919	8.3%	1801	2919	7.5%	1801	2919	7.5%	1801
34	25	0.33	0.15	0.2	3016	10.5%	1800	3016	10.5%	1801	3016	10.5%	1811
35	25	0.33	0.3	0.4	3524	23.4%	1800	3743	27.9%	1800	3370	19.9%	1802
36	25	0.33	0.3	0.2	3939	45.2%	1800	4449	51.4%	1800	3588	24.7%	1800
37	25	0.66	0.15	0.4	2974	21.0%	1813	3044	21.3%	1808	2974	27.1%	1805
38	25	0.66	0.15	0.2	2906	15.0%	1811	2906	25.4%	1800	2967	16.4%	1800
39	25	0.66	0.3	0.4	3656	26.2%	1800	3900	30.8%	1800	3604	25.1%	1800
40	25	0.66	0.3	0.2	4158	53.6%	1800	3779	28.6%	1814	3944	31.5%	1800
41	30	0.33	0.15	0.4	4091	34.0%	1800	3906	30.9%	1803	3873	30.3%	1800
42	30	0.33	0.15	0.2	4279	45.8%	1800	3985	32.2%	1800	4080	33.8%	1800
43	30	0.33	0.3	0.4	5605	55.8%	1800	4762	32.0%	1800	6344	57.4%	1800
44	30	0.33	0.3	0.2	5399	57.1%	1800	4799	43.7%	1801	4803	43.8%	1801
45	30	0.66	0.15	0.4	3745	27.9%	1816	3659	26.2%	1800	4063	33.6%	1818
46	30	0.66	0.15	0.2	3618	25.4%	1800	3803	29.0%	1800	3930	38.8%	1810
47	30	0.66	0.3	0.4	5649	59.0%	1800	4266	36.7%	1800	4620	47.2%	1800
48	30	0.66	0.3	0.2	4800	51.8%	1800	4674	45.2%	1800	5535	55.0%	1800
49	50	0.33	0.15	0.4	9347	53.8%	1800	8855	51.2%	1800	9679	55.4%	1800

50	50	0.33	0.15	0.2	na	na	1800	8350	53.8%	1800	na	na	1800
51	50	0.33	0.3	0.4	na	na	1800	19824	80.5%	1800	na	na	1800
52	50	0.33	0.3	0.2	na	na	1800	25446	83.0%	1800	na	na	1800
53	50	0.66	0.15	0.4	13208	70.8%	1800	na	na	1800	8372	53.9%	1800
54	50	0.66	0.15	0.2	na	na	1800	14792	73.9%	1800	9946	61.2%	1800
55	50	0.66	0.3	0.4	na	na	1800	na	na	1800	na	na	1800
56	50	0.66	0.3	0.2	na	na	1800	na	na	1800	na	na	1800

Exhibit 5: Detailed results of $SBC0^d$, $SBC1^d$, and $SBC3^d$

					$SBC0^d$			$SBC1^d$			$SBC3^d$		
Instance	N	WR	CR	DR	B	G	T	B	G	T	B	G	T
1	5	0.33	0.15	0.4	731	0.0%	0.25	731	0.0%	0.34	731	0.0%	0.20
2	5	0.33	0.15	0.2	665	0.0%	0.52	665	0.0%	0.17	665	0.0%	0.41
3	5	0.33	0.3	0.4	731	0.0%	0.41	731	0.0%	0.17	731	0.0%	0.16
4	5	0.33	0.3	0.2	1079	0.0%	0.14	1079	0.0%	0.16	1079	0.0%	0.13
5	5	0.66	0.15	0.4	731	0.0%	0.28	731	0.0%	0.19	731	0.0%	0.24
6	5	0.66	0.15	0.2	611	0.0%	0.42	611	0.0%	0.31	611	0.0%	0.13
7	5	0.66	0.3	0.4	731	0.0%	0.44	731	0.0%	0.22	731	0.0%	0.16
8	5	0.66	0.3	0.2	817	0.0%	0.41	817	0.0%	0.27	817	0.0%	0.31
9	10	0.33	0.15	0.4	1355	0.0%	14.19	1355	0.0%	4.95	1355	0.0%	8.38
10	10	0.33	0.15	0.2	1311	0.0%	7.84	1311	0.0%	5.35	1311	0.0%	3.72
11	10	0.33	0.3	0.4	1202	0.0%	10.77	1202	0.0%	4.25	1202	0.0%	3.28
12	10	0.33	0.3	0.2	1376	0.0%	9.16	1376	0.0%	9.09	1376	0.0%	5.06
13	10	0.66	0.15	0.4	1223	0.0%	19.12	1223	0.0%	5.14	1223	0.0%	3.24

14	10	0.66	0.15	0.2	1243	0.0%	27.89	1243	0.0%	10.42	1243	0.0%	8.77
15	10	0.66	0.3	0.4	1334	0.0%	10.44	1334	0.0%	5.64	1334	0.0%	1.89
16	10	0.66	0.3	0.2	1363	0.0%	9.22	1363	0.0%	1.70	1363	0.0%	2.02
17	15	0.33	0.15	0.4	1904	6.9%	1800	1904	0.0%	486	1904	0.0%	284
18	15	0.33	0.15	0.2	2059	15.1%	1800	2051	0.0%	1204	2051	0.0%	1510
19	15	0.33	0.3	0.4	2018	0.0%	279	2018	0.0%	558	2018	0.0%	80.34
20	15	0.33	0.3	0.2	2028	0.0%	803	2028	0.0%	409	2028	0.0%	791
21	15	0.66	0.15	0.4	1676	0.0%	370	1676	0.0%	123	1676	0.0%	275
22	15	0.66	0.15	0.2	1803	0.0%	815	1803	0.0%	311	1803	0.0%	272
23	15	0.66	0.3	0.4	1819	0.0%	253	1819	0.0%	348	1819	0.0%	92.34
24	15	0.66	0.3	0.2	2023	0.0%	520	2023	0.0%	582	2023	0.0%	120
25	20	0.33	0.15	0.4	2464	34.2%	1800	2053	0.0%	1102	2053	0.0%	401
26	20	0.33	0.15	0.2	3069	47.2%	1800	2805	42.2%	1800	2616	30.9%	1801
27	20	0.33	0.3	0.4	2888	25.2%	1800	3116	30.7%	1800	3635	57.6%	1800
28	20	0.33	0.3	0.2	3368	51.9%	1800	4631	65.0%	1800	2989	27.7%	1800
29	20	0.66	0.15	0.4	3029	49.1%	1800	2173	23.7%	1800	2173	0.6%	1800
30	20	0.66	0.15	0.2	2973	44.6%	1800	2725	40.5%	1800	2734	37.6%	1800
31	20	0.66	0.3	0.4	2937	44.8%	1800	2828	23.6%	1800	2910	44.3%	1800

32	20	0.66	0.3	0.2	3218	49.7%	1800	3373	45.5%	1800	3046	46.8%	1800
33	25	0.33	0.15	0.4	3318	34.9%	1800	3638	40.6%	1800	3430	37.0%	1800
34	25	0.33	0.15	0.2	3976	45.7%	1800	3807	49.3%	1800	3451	37.4%	1800
35	25	0.33	0.3	0.4	4917	56.1%	1800	4434	56.5%	1800	4050	46.7%	1800
36	25	0.33	0.3	0.2	5447	60.3%	1800	7111	72.9%	1800	4342	50.3%	1800
37	25	0.66	0.15	0.4	3372	35.9%	1800	3802	48.0%	1800	3739	42.2%	1800
38	25	0.66	0.15	0.2	3383	36.2%	1800	3567	39.4%	1800	4186	53.9%	1800
39	25	0.66	0.3	0.4	4321	50.0%	1800	4229	48.9%	1800	5154	62.6%	1800
40	25	0.66	0.3	0.2	4274	54.9%	1800	4984	61.3%	1800	4254	49.2%	1800
41	30	0.33	0.15	0.4	10078	77.0%	1800	15428	85.0%	1800	6164	62.5%	1800
42	30	0.33	0.15	0.2	4532	40.4%	1800	5427	57.4%	1800	6470	64.2%	1800
43	30	0.33	0.3	0.4	9354	71.1%	1800	6241	56.7%	1800	8249	69.8%	1800
44	30	0.33	0.3	0.2	7170	67.7%	1800	16312	85.8%	1800	10852	78.7%	1800
45	30	0.66	0.15	0.4	4528	47.2%	1800	6913	66.5%	1800	6304	63.3%	1800
46	30	0.66	0.15	0.2	8453	72.6%	1800	13839	83.3%	1800	5267	56.1%	1800
47	30	0.66	0.3	0.4	5414	57.3%	1800	15244	84.8%	1800	6923	66.6%	1800
48	30	0.66	0.3	0.2	13309	82.6%	1800	15176	84.7%	1800	7516	69.2%	1800
49	50	0.33	0.15	0.4	na	na	1800	23417	82.3%	1802	na	na	1800

50	50	0.33	0.15	0.2	15343	74.9%	1802	109560	96.5%	1800	na	na	1800
51	50	0.33	0.3	0.4	na	na	1800	108225	96.4%	1800	na	na	1800
52	50	0.33	0.3	0.2	na	na	1800	107324	96.3%	1800	na	na	1800
53	50	0.66	0.15	0.4	15076	74.4%	1800	26784	85.6%	1800	na	na	1800
54	50	0.66	0.15	0.2	na	na	1800	15732	75.5%	1800	na	na	1800
55	50	0.66	0.3	0.4	na	na	1800	112043	96.6%	1800	na	na	1800
56	50	0.66	0.3	0.2	na	na	1800	27164	85.8%	1800	na	na	1800

Exhibit 6: The impact of constraint (2) on the total cost

					SBC3[a]		SBC3[a] + constraint (2) changed	
Instance	N	WR	CR	DR	B	T	B	T
1	5	0.33	0.15	0.4	731	0.21	731	0.22
2	5	0.33	0.15	0.2	665	0.30	665	0.16
3	5	0.33	0.3	0.4	731	0.06	731	0.14
4	5	0.33	0.3	0.2	1079	0.03	1079	0.22
5	5	0.66	0.15	0.4	731	0.23	731	0.14
6	5	0.66	0.15	0.2	611	0.09	611	0.19
7	5	0.66	0.3	0.4	731	0.17	731	0.27
8	5	0.66	0.3	0.2	817	0.13	817	0.19
9	10	0.33	0.15	0.4	1355	2.17	1355	2.42
10	10	0.33	0.15	0.2	1311	3.70	1311	3.03
11	10	0.33	0.3	0.4	1202	2.14	1202	3.53
12	10	0.33	0.3	0.2	1376	5.83	1376	6.14
13	10	0.66	0.15	0.4	1223	2.80	1223	3.22
14	10	0.66	0.15	0.2	1243	1.95	1243	3.02
15	10	0.66	0.3	0.4	1334	2.50	1334	3.03
16	10	0.66	0.3	0.2	1363	2.11	1363	2.27
17	15	0.33	0.15	0.4	1904	261	1904	350
18	15	0.33	0.15	0.2	2051	727	2051	949
19	15	0.33	0.3	0.4	2018	95.29	2018	384
20	15	0.33	0.3	0.2	2028	213	2028	65.83
21	15	0.66	0.15	0.4	1676	36.27	1676	169
22	15	0.66	0.15	0.2	1803	143	1803	122
23	15	0.66	0.3	0.4	1819	146	1819	53.10
24	15	0.66	0.3	0.2	2023	133	2023	237

Exhibit 7: The impact of constraint (5) on the total cost

					Max 2 colors		Max 3 colors		Max 4 colors		Max 5 colors		Max 6 colors		Max 7 colors	
Instance	N	WR	CR	DR	B	T	B	T	B	T	B	T	B	T	B	T
1	5	0.33	0.15	0.4	731	0.21	731	0.20	731	0.19	731	0.17	731	0.19	731	0.20
2	5	0.33	0.15	0.2	665	0.30	665	0.19	665	0.17	665	0.16	665	0.16	665	0.16
3	5	0.33	0.3	0.4	731	0.06	731	0.05	731	0.05	731	0.06	731	0.05	731	0.05
4	5	0.33	0.3	0.2	1079	0.03	1079	0.05	1079	0.03	1079	0.05	1079	0.03	1079	0.06
5	5	0.66	0.15	0.4	731	0.23	731	0.23	731	0.25	731	0.25	731	0.25	731	0.22
6	5	0.66	0.15	0.2	611	0.09	611	0.13	611	0.11	611	0.16	611	0.13	611	0.11
7	5	0.66	0.3	0.4	731	0.17	731	0.17	731	0.17	731	0.19	731	0.17	731	0.16
8	5	0.66	0.3	0.2	817	0.13	817	0.06	817	0.05	817	0.06	817	0.06	817	0.05
9	10	0.33	0.15	0.4	1355	2.17	1355	2.09	1355	1.95	1355	2.27	1355	2.31	1355	2.16
10	10	0.33	0.15	0.2	1311	3.70	1311	3.64	1311	3.42	1311	3.91	1311	3.52	1311	3.61
11	10	0.33	0.3	0.4	1202	2.14	1092	2.31	1092	2.44	1092	2.52	1092	2.27	1092	2.20
12	10	0.33	0.3	0.2	1376	5.83	1376	5.77	1376	5.86	1376	6.42	1376	6.02	1376	5.97
13	10	0.66	0.15	0.4	1223	2.80	1223	3.00	1223	2.69	1223	3.05	1223	2.78	1223	2.89
14	10	0.66	0.15	0.2	1243	1.95	1243	1.74	1243	2.02	1243	2.05	1243	1.99	1243	1.89

15	10	0.66	0.3	0.4	1334	2.50	1223	3.00	1223	2.86	1223	3.20	1223	3.02	1223	2.87
16	10	0.66	0.3	0.2	1363	2.11	1336	4.34	1336	4.13	1336	4.61	1336	4.25	1336	4.25
17	15	0.33	0.15	0.4	1904	261	1828	359	1828	362	1828	367	1828	366	1828	366
18	15	0.33	0.15	0.2	2051	727	1815	431	1815	437	1815	436	1815	439	1815	435
19	15	0.33	0.3	0.4	2018	95.29	1904	1141	1798	1006	1798	672	1798	671	1798	668
20	15	0.33	0.3	0.2	2028	213	1933	544	1790	643	1790	383	1790	381	1790	390
21	15	0.66	0.15	0.4	1676	36.27	1676	64.08	1676	63.15	1676	64.03	1676	64.96	1676	64.88
22	15	0.66	0.15	0.2	1803	143	1680	62.54	1680	65.14	1680	61.63	1680	61.93	1680	61.26
23	15	0.66	0.3	0.4	1819	146	1702	621	1702	65.68	1702	62.87	1702	63.85	1702	63.16
24	15	0.66	0.3	0.2	2023	134	1794	444	1794	546	1794	368	1794	373	1794	371

Exhibit 8: The impact of constraint (7) on the total cost

					SBC3[a]		SBC3[a] without constraint (7)	
Instance	N	WR	CR	DR	B	T	B	T
1	5	0.33	0.15	0.4	731	0.21	731	0.06
2	5	0.33	0.15	0.2	665	0.30	577	0.14
3	5	0.33	0.3	0.4	731	0.06	731	0.06
4	5	0.33	0.3	0.2	1079	0.03	1079	0.06
5	5	0.66	0.15	0.4	731	0.23	731	0.08
6	5	0.66	0.15	0.2	611	0.09	611	0.33
7	5	0.66	0.3	0.4	731	0.17	731	0.08
8	5	0.66	0.3	0.2	817	0.13	817	0.08
9	10	0.33	0.15	0.4	1355	2.17	1355	4.16
10	10	0.33	0.15	0.2	1311	3.70	1311	3.27
11	10	0.33	0.3	0.4	1202	2.14	1202	1.34
12	10	0.33	0.3	0.2	1376	5.83	1376	6.67
13	10	0.66	0.15	0.4	1223	2.80	1223	2.63
14	10	0.66	0.15	0.2	1243	1.95	1243	2.58
15	10	0.66	0.3	0.4	1334	2.50	1334	1.78
16	10	0.66	0.3	0.2	1363	2.11	1363	3.27
17	15	0.33	0.15	0.4	1904	261	1904	165
18	15	0.33	0.15	0.2	2051	727	2043	413
19	15	0.33	0.3	0.4	2018	95.29	2018	87.95
20	15	0.33	0.3	0.2	2028	213	2028	202
21	15	0.66	0.15	0.4	1676	36.27	1676	58.10
22	15	0.66	0.15	0.2	1803	143	1803	125
23	15	0.66	0.3	0.4	1819	146	1819	95.77
24	15	0.66	0.3	0.2	2023	134	2023	84.06

Exhibit 9: The impact of constraint (8) on the total cost

					SBC3[a]		SBC3[a] without constraint (8)	
Instance	N	WR	CR	DR	B	T	B	T
1	5	0.33	0.15	0.4	731	0.21	625	0.17
2	5	0.33	0.15	0.2	665	0.30	665	0.16
3	5	0.33	0.3	0.4	731	0.06	625	0.05
4	5	0.33	0.3	0.2	1079	0.03	926	0.22
5	5	0.66	0.15	0.4	731	0.23	625	0.22
6	5	0.66	0.15	0.2	611	0.09	611	0.19
7	5	0.66	0.3	0.4	731	0.17	625	0.05
8	5	0.66	0.3	0.2	817	0.13	663	0.20
9	10	0.33	0.15	0.4	1355	2.17	1251	3.72
10	10	0.33	0.15	0.2	1311	3.70	1311	4.17
11	10	0.33	0.3	0.4	1202	2.14	1202	2.79
12	10	0.33	0.3	0.2	1376	5.83	1376	6.32
13	10	0.66	0.15	0.4	1223	2.80	1214	3.55
14	10	0.66	0.15	0.2	1243	1.95	1243	3.83
15	10	0.66	0.3	0.4	1334	2.50	1334	2.55
16	10	0.66	0.3	0.2	1363	2.11	1363	6.04
17	15	0.33	0.15	0.4	1904	261	1904	310
18	15	0.33	0.15	0.2	2051	727	2051	580
19	15	0.33	0.3	0.4	2018	95.29	2018	222
20	15	0.33	0.3	0.2	2028	213	2028	938
21	15	0.66	0.15	0.4	1676	36.27	1676	173
22	15	0.66	0.15	0.2	1803	143	1803	445
23	15	0.66	0.3	0.4	1819	146	1819	271
24	15	0.66	0.3	0.2	2023	134	2023	112

Exhibit 10: The impact of constraints (9) and (10) on the total cost

					SBC3[a]		SBC3[a] without constraint (9) and (10)	
Instance	N	WR	CR	DR	B	G	B	G
1	5	0.33	0.15	0.4	731	0.21	731	0.16
2	5	0.33	0.15	0.2	665	0.30	665	0.25
3	5	0.33	0.3	0.4	731	0.06	731	0.21
4	5	0.33	0.3	0.2	1079	0.03	1079	0.05
5	5	0.66	0.15	0.4	731	0.23	731	0.13
6	5	0.66	0.15	0.2	611	0.09	611	0.20
7	5	0.66	0.3	0.4	731	0.17	731	0.22
8	5	0.66	0.3	0.2	817	0.13	817	0.30
9	10	0.33	0.15	0.4	1355	2.17	1355	6.23
10	10	0.33	0.15	0.2	1311	3.70	1311	4.72
11	10	0.33	0.3	0.4	1202	2.14	1202	8.49
12	10	0.33	0.3	0.2	1376	5.83	1207	10.92
13	10	0.66	0.15	0.4	1223	2.80	1223	13.47
14	10	0.66	0.15	0.2	1243	1.95	1243	5.02
15	10	0.66	0.3	0.4	1334	2.50	1334	8.28
16	10	0.66	0.3	0.2	1363	2.11	1363	32.50
17	15	0.33	0.15	0.4	1904	261	1871	1800
18	15	0.33	0.15	0.2	2051	727	1938	1800
19	15	0.33	0.3	0.4	2018	95.29	2274	1800
20	15	0.33	0.3	0.2	2028	213	2118	1800
21	15	0.66	0.15	0.4	1676	36.27	1676	202
22	15	0.66	0.15	0.2	1803	143	1686	425
23	15	0.66	0.3	0.4	1819	146	1819	498
24	15	0.66	0.3	0.2	2023	134	2023	219

References

Albareda-Sambola, M., Fernández, E., & Laporte, G. (2011). A computational comparison of several models for the exact solution of the capacity and distance constrained plant location problem. Computers & Operations Research, 38(8), 1109-1116.

Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W., & Vance, P. H. (1998). Branch-and-price: Column generation for solving huge integer programs. Operations research, 46(3), 316-329.

Baumann, P., Forrer, S., & Trautmann, N. $(2014)^{a}$ Planning of a make-to-order production process in the printing industry. Flexible Services and Manufacturing Journal, 1-27.

Baumann, P., & Trautmann, N. $(2014)^{b}$. Efficient symmetry-breaking formulations for grouping customer orders in a printing shop. In Industrial Engineering and Engineering Management (IEEM), 2014 IEEE International Conference on (pp. 506-510). IEEE.

Ben Amor, H., & Valério de Carvalho, J. (2005). Cutting stock problems. Column generation, 131-161.

Coelho, L. C., & Laporte, G. (2014). Improved solutions for inventory-routing problems through valid inequalities and input ordering. International Journal of Production Economics, 155, 391-397.

Colbourn, C. J., & Dinitz, J. H. (Eds.). (2006). Handbook of combinatorial designs. CRC press.

Degraeve, Z., & Vandebroek, M. (1998). A mixed integer programming model for solving a layout problem in the fashion industry. Management Science, 44(3), 301-310.

Degraeve, Z., Gochet, W., & Jans, R. (2002). Alternative formulations for a layout problem in the fashion industry. European Journal of Operational Research, 143(1), 80-93.

Denton, B. T., Miller, A. J., Balasubramanian, H. J., & Huschka, T. R. (2010). Optimal allocation of surgery blocks to operating rooms under uncertainty. Operations research, 58(4-part-1), 802-816.

Ekici, A., Ergun, Ö., Keskinocak, P., & Lagoudakis, M. G. (2010). Optimal job splitting on a multi-slot machine with applications in the printing industry. Naval Research Logistics (NRL), 57(3), 237-251.

Elaoud, S., Teghem, J., & Bouaziz, B. (2007). Genetic algorithms to solve the cover printing problem. Computers & operations research, 34(11), 3346-3361.

Jans, R., & Degraeve, Z. (2008). A note on a symmetrical set covering problem: The lottery problem. European Journal of Operational Research, 186(1), 104-110.

Jans, R. (2009). Solving lot-sizing problems on parallel identical machines using symmetry-breaking constraints. INFORMS Journal on Computing, 21(1), 123-136.

Jans, R., & Desrosiers, J. (2013). Efficient symmetry breaking formulations for the job grouping problem. Computers & Operations Research, 40(4), 1132-1142.

Margot, F. (2002). Pruning by isomorphism in branch-and-cut. Mathematical Programming, 94(1), 71-90.

Margot, F. (2003). Exploiting orbits in symmetric ILP. Mathematical Programming, 98(1), 3-21.

Martens, J. (2004). Two genetic algorithms to solve a layout problem in the fashion industry. European Journal of Operational Research, 154(1), 304-322.

Margot, F. (2007). Symmetric ILP: Coloring and small integers. Discrete Optimization, 4(1), 40-62.

Margot, F. (2010). Symmetry in integer linear programming. In 50 Years of Integer Programming 1958-2008 (pp. 647-686). Springer Berlin Heidelberg.

Nascimento, D. B., de Figueiredo, J. N., Mayerle, S. F., Nascimento, P. R., & Casali, R. M. (2010). A state-space solution search method for apparel industry spreading and cutting. International Journal of Production Economics, 128(1), 379-392.

Pochet, Y., & Wolsey, L. A. (2006). Production planning by mixed integer programming. Springer Science & Business Media.

Romero, D., & Alonso-Pecina, F. (2012). Ad hoc heuristic for the cover printing problem. Discrete Optimization, 9(1), 17-28.

Rose, D. M., & Shier, D. R. (2007). Cut scheduling in the apparel industry. Computers & Operations Research, 34(11), 3209-3228.

Sherali, H. D., & Smith, J. C. (2001). Improving discrete model representations via symmetry considerations. Management Science, 47(10), 1396-1407.

Sherali, H. D., Bae, K. H., & Haouari, M. (2013). An integrated approach for airline flight selection and timing, fleet assignment, and aircraft routing. Transportation Science, 47(4), 455-476.

Teghem, J., Pirlot, M., & Antoniadis, C. (1995). Embedding of linear programming in a simulated annealing algorithm for solving a mixed integer production planning problem. Journal of Computational and Applied Mathematics, 64(1-2), 91-102.

Tuyttens, D., & Vandaele, A. (2010). Using a greedy random adaptative search procedure to solve the cover printing problem. Computers & Operations Research, 37(4), 640-648.

Tuyttens, D., & Vandaele, A. (2014). Towards an efficient resolution of printing problems. Discrete Optimization, 14, 126-146.

Yang, C. L., Huang, R. H., & Huang, H. L. (2011). Elucidating a layout problem in the fashion industry by using an ant optimisation approach. Production Planning & Control, 22(3), 248-256.