processes

Article

Optimizing Painting Sequence Scheduling Based on Adaptive Partheno-Genetic Algorithm

Jun Yang [1,2], Tong Sun [1], Xiuxiang Huang [1], Ke Peng [1], Zhongxiang Chen [1,*], Guoguang Qian [3] and Zekai Qian [3]

[1] College of Engineering and Design, Hunan Normal University, Changsha 410081, China; yangjun@hunnu.edu.cn (J.Y.); 201970182808@hunnu.edu.cn (T.S.); hxxpxj01@163.com (X.H.); pengke@hunnu.edu.cn (K.P.)
[2] State Key Laboratory of High Performance Complicated Manufacturing, Central South University, Changsha 410081, China
[3] Hunan Aotong Intelligent Research Institute Co., Ltd., Changsha 410081, China; 18637221111@163.com (G.Q.); qianzekai2005@163.com (Z.Q.)
* Correspondence: chenzx@hunnu.edu.cn

Abstract: In this paper, we formulate and solve a novel real-life large-scale automotive parts paint shop scheduling problem, which contains color arrangement restrictions, part arrangement restrictions, bracket restrictions, and multi-objectives. Based on these restrictions, we construct exact constraints and two objective functions to form a large-scale multi-objective mixed-integer linear programming problem. To reduce this scheduling problem's complexity, we converted the multi-objective model into a multi-level objective programming problem by combining the rule-based scheduling algorithm and the adaptive Partheno-Genetic algorithm. The rule-based scheduling algorithm is adopted to optimize color changes horizontally and bracket replacements vertically. The adaptive Partheno-Genetic algorithm is designed to optimize production based on the rule-based scheduling algorithm. Finally, we apply the model to the actual optimization problem that contained 829,684 variables and 137,319 constraints, and solved this problem by Python. The proposed method solves the optimal solution, consuming 575 s.

Keywords: NP-hard; multi-objective mixed integer linear programming; rule-based scheduling algorithm; adaptive Partheno-Genetic algorithm

Citation: Yang, J.; Sun, T.; Huang, X.; Peng, K.; Chen, Z.; Qian, G.; Qian, Z. Optimizing Painting Sequence Scheduling Based on Adaptive Partheno-Genetic Algorithm. *Processes* **2021**, *9*, 1714. https://doi.org/10.3390/pr9101714

Academic Editor: Shixin Liu

Received: 13 August 2021
Accepted: 20 September 2021
Published: 24 September 2021

1. Introduction

The automotive parts manufacturing industry has entered a golden period of development [1]. In the automotive supply industry, the paint shops provide a large number of painted items for car manufacturing [2]. At present, some important factors, such as efficiency, cost, low carbon and environmental protection, should be considered in the design of paint shops for the automotive supply industry. Many modern factories have introduced high-level automation manufacturing systems, including conveyor belt systems, paint guns, loading robots, and unloading robots. In this sophisticated production process, an excellent paint schedule is difficult to establish to fulfill many constraints and optimization requirements. Thus, there is a strong need to develop efficient automated scheduling techniques for painting sequence scheduling in the automotive supply industry.

In recent years, an increasing number of studies have discussed related scheduling problems. Each case of research has its unique characteristics that must be considered for developing appropriate models and solving approaches. In the automotive supply industry, the general paint scheduling problem has been proved to be a non-deterministic polynomial (NP) complete problem [3,4]. To reduce the difficulty of scheduling and obtain feasible solutions, a single objective, such as the number of color changes, is often considered. A color change occurs whenever two consecutive parts need to be painted with different colors. For each color change, the color jets of the spray robots have to be cleaned, resulting in inevitable costs and water pollution. Based on these reasons, some researchers have studied the scheduling scheme for optimizing the number of color changes. In reference [3], the main goal of this paint shop problem is to find the optimal coloring for a given sequence of jobs that minimizes the required color changes. A dynamic programming method is applied to solve this problem. Linear programming and local-search-based approaches, combined with a metaheuristic for tackling the size of the word that is small for both the necklace problem and the paint shop problem, have been studied [5]. For minimizing the number of color changes, reference [6] provides a method that uses selectivity banks to establish a model based on color batches. The optimal schedule is achieved by the branch and bound approach. On the topic of color-batching, further studies, such as the rule-based fill approach and the heuristic-based release approach, are presented in references [7,8]. Moreover, they introduce two virtual resequencing techniques, both of which can be individually integrated into physical resequencing while integrated resequencing is used.

A capacity limitation and a type restriction of overhead hangers were considered in reference [9]. A mixed-integer programming model for the scheduling problem was developed. Moreover, a 2-Opt improvement algorithm and a tabu search metaheuristic algorithm were designed. However, these solution methods are not suitable for scheduling problems that consider due dates, color changes, and carrier changes. In references [4,10], the authors proposed a novel paint shop scheduling problem that aims to achieve the optimal schedule by minimizing color changes, minimizing carrier changes, carrier constraints, and due date constraints. They provided a novel metaheuristic solution approach using simulated annealing to solve this problem. Furthermore, constraint programming models were established in reference [4], motivated by reference [10]. Additionally, the researchers analyzed the problem's complexity and proved that the decision variant is NP-complete. However, to the best of our knowledge, some colors have order constraints and some automotive parts cannot be placed adjacent to each other. In addition, it is necessary to design a novel algorithm to improve the efficiency of the solution.

In this paper, based on [4], we developed a multi-objective mixed-integer linear programming model for the real-life painting sequence scheduling problem, which contained paint arrangement restrictions, parts arrangement restrictions, and multi-objectives that have not been reported in the existing literature. Our main contribution can be summarized as follows. First, we analyzed the processing characteristics and established a multi-objective mixed-integer programming model based on the above restrictions. Second, inspired by references [7,8], a rule-based scheduling algorithm is designed to minimize the number of color changes, and an adaptive Partheno-Genetic algorithm is designed to maximize production and minimize carrier changes. That means our scheduling problem was reduced by combining these two algorithms. Finally, we combined the proposed method to solve an actual instance and obtain the optimal solutions. Computational results illustrated that the proposed model, combined with the designed algorithms, can effectively solve the large-scale paint shop scheduling problem.

This paper is organized as follows. We describe the researched problem in Section 2, including color arrangement restrictions, part arrangement restrictions and objectives. Section 3 establishes the constraints and objective functions. Section 4 proposes the rule-based scheduling algorithm and the adaptive Partheno-Genetic algorithm, and applies the designed algorithms to an actual instance to demonstrate the validity of the proposed method. Finally, Section 5 provides a conclusion.

2. Problem Formulation

2.1. Problem Description

In this article, the general production process of the painting operation is depicted in Figure 1. We considered the painting process of car parts using automated devices, including a cyclic conveyor belt, a spray robot, and 303 skids. On each skid, we can place six brackets, and one part can be arranged on each bracket. The painting time for each skid is 60 s, and the painting time for the 303 skids was approximately 5.5 h. Restrictions in the painting process include the number of brackets, the painting sequence of the parts, the arrangement of colors, the arrangement of parts, and the production demands. Thus, the painting plan should be scheduled to meet the production demands in one lead time of 44 h as much as possible.

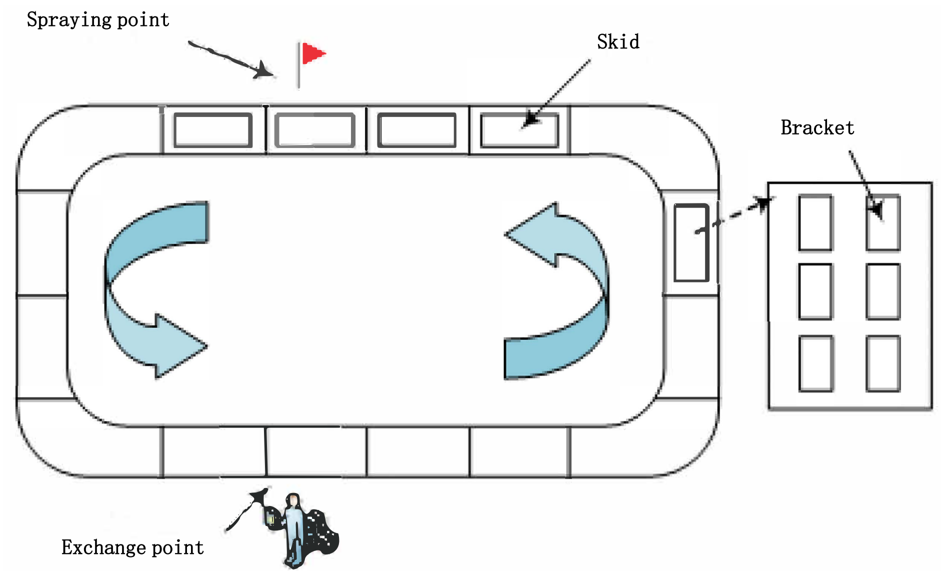


Figure 1. Schematic diagram of spraying process.

2.2. Color Arrangement Restrictions

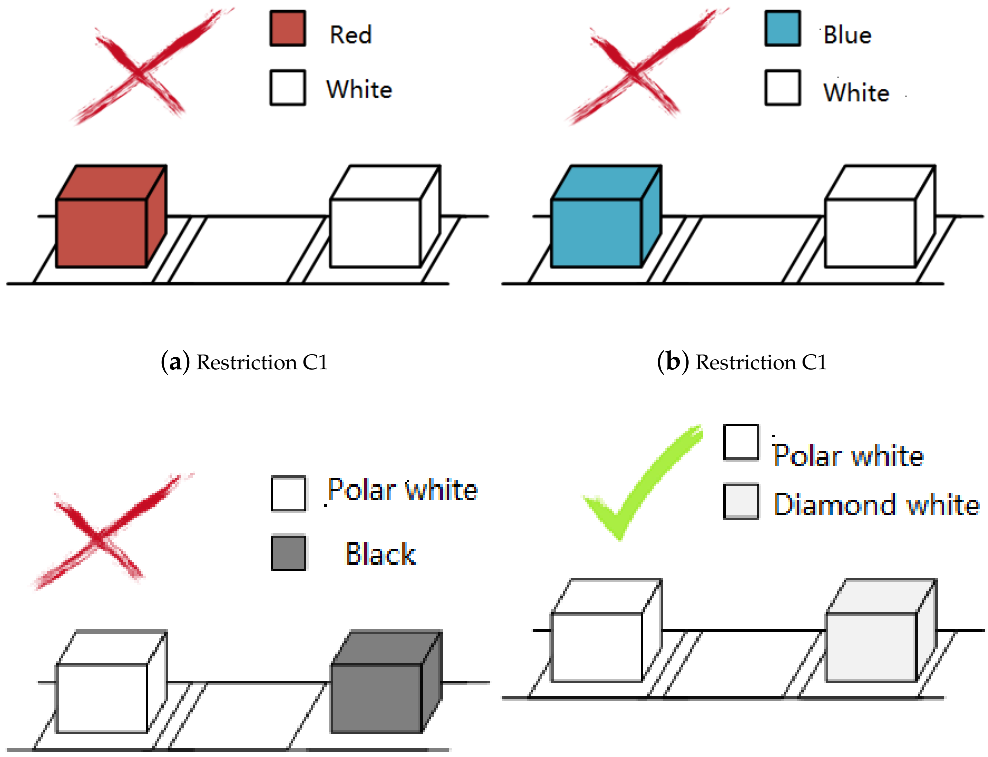
The parts placed on the bracket are subjected to the following painting sequence: primer (black/white)-topcoat-varnish (high gloss/matte). The topcoat determines the color of a part, and each topcoat has a corresponding primer and varnish color. If parts placed on the two skids successively require different topcoats, the process requires a color change, which means that the color of the corresponding spray gun must be changed. The color change process requires inserting primer parts on a skid between these two skids as a transition. The transition can be any type of part, such as an after-sale spare part without a topcoat. All categories of topcoat are given in Table 1.

Table 1. All categories of topcoats.

Color	Topcoat	No.	Color	Topcoat	No.
White	Polar White	1	Blue	Shining Blue	6
	Diamond White	2	Silver	Milan Silver	7
Red	Ruby Red	3		Iridium Silver	8
Blue	Denim Blue	4	Black	Universe Black	9
	Sapphire Blue	5		Obsidian Black	10

Restrictions on the sequence of topcoat changes are detailed as:

C1. The color white cannot be arranged after the colors red and blue as shown in Figure 2a,b;
C2. Black cannot be arranged after the topcoat polar white as shown in Figure 2c;
C3. In front of the topcoat diamond white, the topcoat polar white is required as shown in Figure 2d.



(**a**) Restriction C1 (**b**) Restriction C1 (**c**) Restriction C2 (**d**) Restriction C3

Figure 2. Restrictions on the sequences of topcoats.

2.3. Part Arrangement Restrictions

To ensure the quality of the final product, certain restrictions of parts arrangements should be considered during the painting process. All types of parts are given in Table A1. The restrictions on the placement of parts are detailed as:

P1. The parts of threshold B, the parts of threshold C, and the parts of threshold A, threshold D, Rear bumper A, and threshold trim A are classified by three kinds of incompatible processing parts, which means that they cannot be placed on two skids successively. If they are, they will cause product collision damage;

P2. The parts of threshold B and threshold C cannot be placed on two skids successively with any radar bracket.

2.4. Purpose and Symbol Statements

The purpose of this paper is to schedule an optimized one lead time 44 h painting plan, which is required to fulfill the production demands as much as possible. The number of the color changes and the bracket replacement are to be minimized. This is a multi-objective optimization problem. In this research, we established a mixed-integer linear programming model, and achieved the optimal solutions by designing a novel algorithm.

This article proclaims the following ten symbols that are used frequently, and the remaining symbols are indicated when they are used as shown in Table 2.

Table 2. The ten symbols used frequently.

Symbol	Symbol Description
i	Parts number, $i \in I$; $I = \{1, 2, \cdots, 31\}$
j	Topcoat number, $j \in J$; $J = \{1, 2, \cdots, 10\}$
k	Skid number, $k \in K$; $K = \{1, 2, \cdots, 303\}$
n	Cycle number, $n \in N$; $N = \{1, 2, \cdots, 8\}$
$x_{ijk}^{(n)}$	Decision variable, the parts i are painted with topcoat j on the k-th skid in the nth cycle.
y_{ij}	Indicator variable, the parts i, painted with topcoat j, may be completed or not.
D_i	The maximum number of brackets for parts i
$X_{ij}^{(n)}$	The total number of parts i painted with topcoat j in the n-th cycle
$\tilde{X}_{ij}$	The number of parts i painted with topcoat j
R_{ij}	The production demands for parts i painted with topcoat j

3. Model

3.1. Decision Variables and Indicator Variables

Each skid can only be scheduled for one painting task at most in the same cycle. Thus, the decision variables are 0–1 and are defined as follows:

$$x_{ijk}^{(n)} = \begin{cases} 1 & \text{if parts } i \text{ are painted with topcoat } j \text{ on the } k\text{-th skid in the } n\text{-th cycle.} \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

The constraints to ensure that the variable does not conflict can be expressed as:

$$\sum_{i \in I} \sum_{j \in J} x_{ijk}^{(n)} \leq 1 \quad k \in K, n \in N, \quad (2)$$

where K and N are sets defined in Table 2. In order to characterize the completion of the types of parts, we define the indicator variables as:

$$y_{ij} = \begin{cases} 1 & \text{if the number of parts } i \text{ sprayed with topcoat } j \text{ meet the production demands } R_{ij}, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

3.2. Global Constraints

3.2.1. Restrictions of Brackets

The number of corresponding brackets restricts the number of painted parts in each circle, and the same type of brackets may also correspond to different topcoats for the same type of parts. Each skid can arrange six brackets, which means that the m skids used and the number of painted parts are subject to $6(m-1)+1 \leq X_{ij}^{(n)} \leq 6m$. Therefore, to express these bracket restrictions, the symbol $X_{ij}^{(n)}$ represents the total number of parts i painted with topcoat j during the $n-$th circle with the following constraints:

$$6\left(\sum_{k \in K} x_{ijk}^{(n)} - 1\right) + 1 \leq X_{ij}^{(n)} \leq 6 \sum_{k \in K} x_{ijk}^{(n)} \quad i \in I, j \in J, n \in N, \quad (4)$$

where I, J and N are sets defined in Table 2. In addition, the total number of painted parts is constrained by the restrictions of the bracket, which has the following constraint:

$$\sum_{j \in J} X_{ij}^{(n)} \leq D_i \quad i \in I, n \in N. \quad (5)$$

We assumed that transitional parts do not count the number of used brackets. Figure 3 shows that if a specific type of part is arranged on a skid, it should be filled as much as possible. Therefore, the $X_{ij}^{(n)}$ corresponding to the above constraints (4) and (5) can be used to represent the total processing number of parts i sprayed with topcoat j during the n-th cycle.

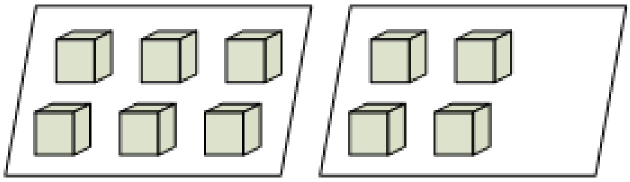


Figure 3. A reasonable arrangement.

3.2.2. Demands Restrictions

The total number of parts i painted with topcoat j should fulfill the production demands R_{ij} leading to the following constraints:

$$\sum_{n \in N} X_{ij}^{(n)} \geq R_{ij} \quad i \in I, j \in J. \tag{6}$$

Due to resource limitations, constraints (6) may not be true. Parts that exceed the demand will not produce economic benefits, and we restrict the total number of painted parts so that they cannot exceed twice the production demand. Thus, the constraints (6) are rewritten:

$$R_{ij} \cdot (y_{ij} + 1) \geq \sum_{n \in N} X_{ij}^{(n)} \geq R_{ij} \cdot y_{ij} \quad i \in I, j \in J. \tag{7}$$

3.2.3. Color Change Restrictions

If parts placed on two skids successively must be painted with different topcoats, the color of the corresponding spray gun needs to be changed. Color change requires a transition skid, and the corresponding parts are not counted in the total output of the transition skid. The corresponding situations are depicted in Figure 4.

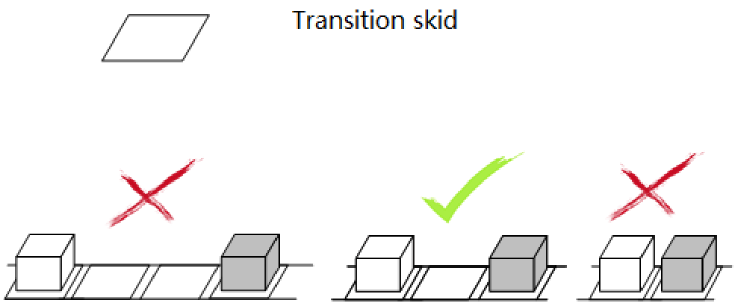


Figure 4. Rational utilization of a transition skid.

The constraints that satisfy the above relationship are as follows:

$$\underbrace{\sum_{i_1 \in I} x_{i_1,j_1,k}^{(n)}}_{①} + \underbrace{\sum_{i_2 \in I} \sum_{\substack{j_2 \in J \\ j_2 \neq j_1}} x_{i_2,j_2,k+1}^{(n)}}_{②} \leq 1, \quad j_1 \in J, k \in K - \{303\}, n \in N. \tag{8}$$

$$\sum_{i_1 \in I} \sum_{j_1 \in J} x_{i_1,j_1,k}^{(n)} + \sum_{i_2 \in I} \sum_{j_2 \in J} x_{i_2,j_2,k+1}^{(n)} \geq 1 \quad k \in K - \{303\}, n \in N, \tag{9}$$

where the set $K - \{303\}$ denotes that the set K removes the point $\{303\}$.

Inequality (8) demonstrates that, when the k-th skid is arranged with topcoat j_1, expression ① is equal to 1. In order to meet the above constraint (8), the other topcoat $j_2 \neq j_1$ on the $(k+1)$-th skid cannot be arranged, and the expression ② is equal to

0. Inequality (9) demonstrates that one empty skid can often be placed between two consecutive painting tasks.

3.2.4. Color Arrangement Restrictions

- According to restriction C1 and Figure 5, the paint tasks $j_1 = \{1,2\}$ and $j_2 = \{3,4,5,6\}$ are incompatible with the skids k and $k+2$. Thus, we establish the following constraints.

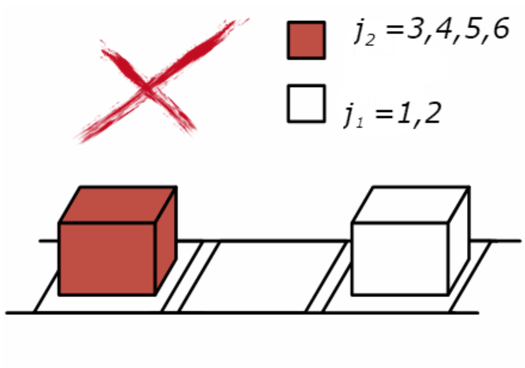


Figure 5. Diagram of Restriction C1.

$$\underbrace{\sum_{i_1 \in I} x^{(n)}_{i_1,j_1,k+2}}_{③} + \underbrace{\sum_{i_2 \in I} x^{(n)}_{i_2,j_2,k}}_{④} \le 1 \quad j_1 \in \{1,2\}, j_2 \in \{3,4,5,6\}, k \in K - \{302,303\}, n \in N, \tag{10}$$

where the set $K - \{302,303\}$ denotes that the set K removes the points $\{302,303\}$. When the parts on the $(k+2)$-th skid are painted with the color white, the expression ③ is equal to 1. To satisfy the constraint (10), the parts on the k-th skid cannot be painted with the color red or blue, that is, the expression ④ is equal to 0. Other scenarios are not subject to this constraint.

- According to restriction C2 and Figure 6, the paint tasks $j_1 = \{9,10\}$ and $j_2 = 1$ are incompatible with the skids k and $k+2$. Thus, we establish following constraints.

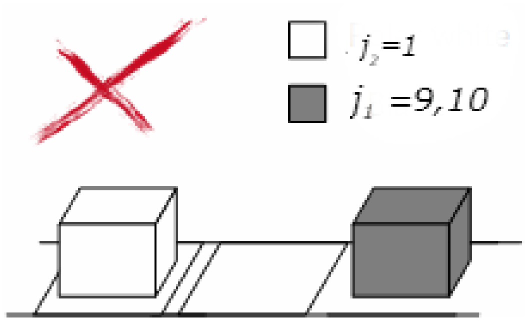


Figure 6. Diagram of Restriction C2.

$$\underbrace{\sum_{i_1 \in I} x^{(n)}_{i_1,j_1,k+2}}_{⑤} + \underbrace{\sum_{i_2 \in I} x^{(n)}_{i_2,j_2,k}}_{⑥} \le 1 \quad j_1 \in \{9,10\}; j_2 \in \{1\}; k \in K - \{302,303\}. \tag{11}$$

When the parts placed on the k-th skid are painted with polar white, the expression ⑤ is equal to 1. To satisfy the constraint (11), the parts on the $(k+1)$-th skid cannot be painted with the color black, that is, the value of expression ⑥ is 0. Other scenarios are not subject to this constraint.

- According to restriction C3 and Figure 7, the paint task $j_2 = 2$ must be arranged after the task $j_1 = 1$. Thus, we establish the following constraints.

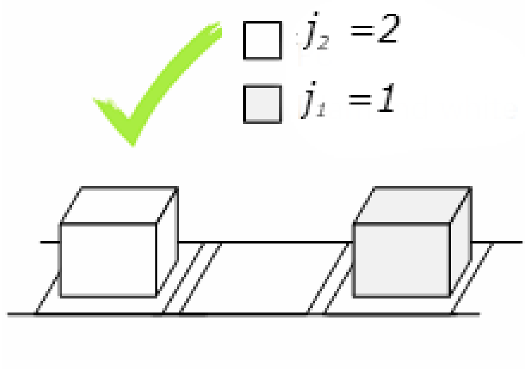


Figure 7. Diagram of Restriction C3.

$$\underbrace{\sum_{i_1 \in I} x^{(n)}_{i_1,2,k+2}}_{⑦} \leq \underbrace{\sum_{i_2 \in I} \left(x^{(n)}_{i_2,2,k+1} + x^{(n)}_{i_2,1,k} \right)}_{⑧} \quad k \in K - \{302, 303\}, n \in N. \tag{12}$$

When the parts placed on the $(k+2)$-th skid are painted with diamond white, the value of expression ⑦ is equal to 1. Diamond white may come from polar white on the k-th skid by color change, or the previous parts are painted diamond white on the $(k+1)$-th skid. The constraints (8) and (9) exactly guarantee that $x^{(n)}_{i_2,2,k+1}$ and $x^{(n)}_{i_2,1,k}$ in expression ⑧ are not equal to 1 simultaneously. Therefore, the constraints in the form of (12) can express all possible layouts of diamond white, as seen in Figure 8.

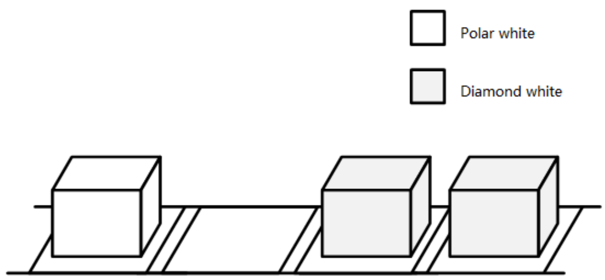


Figure 8. Arrangement of polar white and diamond white.

3.2.5. Parts Arrangement Restrictions

Some parts cannot be arranged successively to avoid scratching and damaging. However, they can be arranged with a transition skid as seen in Figure 9. To facilitate the following constraints expressions, the following symbols in Table 3 are introduced to indicate these specific parts.

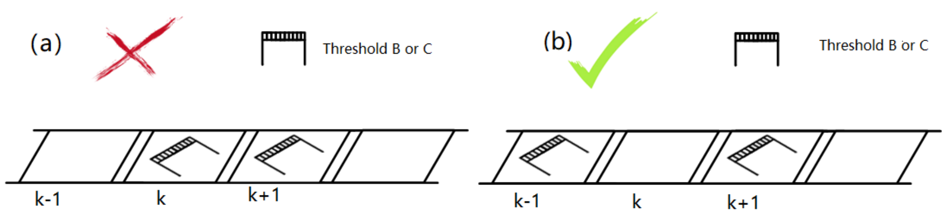


Figure 9. A reasonable arrangement of specific parts.

Table 3. Classification of the parts according to restrictions.

Symbol	Symbol Description
$I_1 = \{22\}$	Threshold B
$I_2 = \{23\}$	Threshold C
$I_3 = \{11, 21, 24, 26\}$	Rear bumper A, threshold A, threshold D and threshold trim A
$I_4 = \{27, 28, 29, 30, 31\}$	All radar brackets

- According to the restriction P1, we established the following constraints.

$$2\left(1 - \sum_{i_1 \in I_m} \sum_{j_i \in J} x^{(n)}_{i_1, j_i, k}\right) \geq \sum_{i_2 \in \Omega} \sum_{j_2 \in J} \left(x^{(n)}_{i_2, j_2, k+1} + x^{(n)}_{i_2, j_2, k-1}\right) \quad m \in \{1, 2, 3\}, k \in K - \{1, 303\}, n \in N, \tag{13}$$

where $\Omega = I_1 \cup I_2 \cup I_3 - I_m$. As shown in Figure 9a, when parts belonging to I_m are placed on the k-th skid, the left of (13) is equal to 0, and the right of (13) must be 0. The parts in Ω cannot be arranged on the $(k+1)$-th and $(k-1)$-th skids. As shown in Figure 9b, when parts belonging to I_m are not placed on the k-th skid, the left of (13) is equal to 2, and the right of (13) can reach 0, 1, and 2, that is, the types of parts placed on two consecutive skids are not restricted. Therefore, the scale 2 in the left of (13) guarantees the feasibility of the solution.

- According to the restriction P2, we established the following constraints.

$$2\left(1 - \sum_{i_1 \in I_1 \cup I_2} \sum_{j_1 \in J} x^{(n)}_{i_1, j_1, k}\right) \geq \sum_{i_2 \in I_4} \sum_{j_2 \in J} \left(x^{(n)}_{i_2, j_2, k+1} + x^{(n)}_{i_2, j_2, k-1}\right) \quad k \in K - \{1, 303\}, n \in N. \tag{14}$$

As shown in Figure 9a, when parts belonging to $I_1 \cup I_2$ are placed on the k-th skid, the left of (14) is equal to 0 and the right of (14) must be equal 0. At this time, the parts belonging to I_4 cannot be placed on the $(k+1)$-th and $(k-1)$-th skids. As shown in Figure 9b, when parts belonging to $I_1 \cup I_2$ are not placed on the k-th skid, the left of (14) is equal to 2, and the right of (14) can reach 0, 1, and 2, that is, the types of parts placed on two consecutive skids are not restricted. Therefore, the scale 2 in the left of (14) guarantees the feasibility of the solution.

3.3. Objective

3.3.1. Objective of Production Demands

Here, the number of painted parts in the scope of production demand is called the number of effective painted parts, and its expression is defined as:

$$\tilde{X}_{ij} = \min\left\{\sum_{n \in N} X^{(n)}_{ij}, R_{ij}\right\} \quad i \in I, j \in J. \tag{15}$$

We constructed an objective that maximizes the number of all effective painted parts as follows:

$$\max \sum_{i \in I} \sum_{j \in J} \tilde{X}_{ij}. \tag{16}$$

Remark 1. *According to the analysis, when the production scale is small and the production demands cannot be met, the larger the number of effectively painted parts, the fewer the color changes. Therefore, the optimal objective of meeting production demands also includes minimal color changes.*

3.3.2. Minimize the Number of Bracket Replacements

We used the following formula:

$$\left|\sum_{j_1\in J} x_{i,j_1,k}^{(n)} - \sum_{j_2\in J} x_{i,j_2,k}^{(n+1)}\right| = \begin{cases} 0 & \text{if bracket } i \text{ is placed on } k\text{-th skid in} \\ & n\text{-th cycle and } (n+1)\text{-th cycle} \\ 1 & otherwise \end{cases}. \quad (17)$$

We can indicate whether the parts placed on the kth skid are same in the n-th and $n+1$-th cycle, then

$$\min \sum_{i\in I}\sum_{k\in K}\sum_{n\in N-\{8\}}\left|\sum_{j_1\in J} x_{i,j_1,k}^{(n)} - \sum_{j_2\in J} x_{i,j_2,k}^{(n+1)}\right| \quad (18)$$

can be used to denote the objective to minimize the number of bracket replacements. To simplify the expression (18), we introduced indicator variables $S_{ik}^{(n)} \in \{0,1\}$ and set

$$\underbrace{\sum_{j_1\in J} x_{i,j_1,k}^{(n)}}_{⑨} \le \underbrace{\sum_{j_2\in J} x_{i,j_2,k}^{(n+1)} + S_{ik}^{(n)}}_{⑩} \quad i \in I, k \in K, n \in N-\{8\}, \quad (19)$$

where the set $N-\{8\}$ denotes that the set N removes the point $\{8\}$. When the expression ⑨ is equal to 1, the right of (19) takes at least 1. If the expression ⑩ is equal to 1, $S_{ik}^{(n)}$ can take 0, that is, there is the same type of bracket on the k-th skid in the n-th and $(n+1)$-th cycles. If the expression ⑩ is equal to 0, $S_{ik}^{(n)}$ must be equal to 1 to ensure (19) holds. Therefore, the objective in (18) can be converted into:

$$\min \sum_{i\in I}\sum_{k\in K}\sum_{n\in N-\{8\}} S_{ik}^{(n)}. \quad (20)$$

By (20), the minimal number of bracket replacements can be realized.

3.4. Multi-Objective Mixed-Integer Linear Programming Model

With the above decision variables and restriction constraints, the complete mixed-integer linear programming model can be established.

$$\begin{array}{l} \text{Obj1: max} \sum_{i\in I}\sum_{j\in J} \tilde{X}_{ij} \\ \text{Obj2: min} \sum_{i\in I}\sum_{k\in K}\sum_{n\in N-\{8\}} S_{ik}^{(n)} \end{array} \text{ subject to (2), (4), (5), (7)–(15), (19).} \quad (21)$$

Equation (16) defines the first objective, that is, maximizing the number of all effective painted parts. Equation (20) defines the second objective, that is, minimizing the total number of bracket replacements.

This scheduling problem is NP-hard, and we can prove this by restriction. If a special case of the considered problem generated by restriction is the same as the known NP-hard problem, then the considered problem is also NP-hard because it contains the hard problem [11]. If we fulfill all demands and do not consider the number of bracket restrictions, this special scheduling problem is a single objective programming approach. With regard to the spray as a salesman, with one paint task as a city and the color change and transition skid as the travel cost, then minimizing the number of color changes is equivalent to minimizing the cost of the Travel Salesman Problem (TSP), which is a well known NP-hard problem [12]. Thus, our scheduling problem is NP-hard. Due to this problem's large scale, it is necessary to design an effective mathematical model and choose an algorithm suitable for addressing the NP-hardness of the considered problem.

4. Algorithm and Solution

4.1. Rule-Based Scheduling Algorithm (RSA)

There are mutually exclusive rules, ordered rules, and disordered rules for arranging colors and parts for the painting process in this research. According to the basic idea of a rule-based classifier in machine learning, we designed a rule-based scheduling algorithm (RSA) to obtain the initial solution to reduce the iteration number for the heuristic algorithm [13]. Therefore, the RSA is designed as follows.

Step 1. Make statistics on production demands according to the type of color. According to the principle of fewer color changes, color types with the greatest demand are processed first. For example, consider the real instance in this paper, and the priority is as follows, based on the parts requirements. Color 10(4068)>Color 1(3854)>Color 2(1746)>Color 6(1110)>Color 3(935)>Color 8(704)>Color 7(551)>Color 5(429)>Color 9(43)>Color 4(15);

Step 2. Construct a task priority sequence and the sequence is defined by a vector (i, j, R_{ij}) (parts, colors, demand). Firstly, the painting tasks are scheduled based on the priority of the parts type. When the parts types have the same priorities, they could be scheduled according to the colors;

Step 3. Traverse every skid in each cycle and conduct polling according to task priority;

Step 4. Assign task to the $k-$th skid on the $n-$th cycle$(1 \leq n \leq 8, 1 \leq k \leq 303)$, and set the task number $t = 1$;

Step 5. Select the $t-$th task from the task sequence. If $R_{ij} = 0$ or $D_i^n = 0$ in the $n-$th circle, update the skid status as an unused skid, set the task number as $t = t + 1$, and return to Step 5. Otherwise, execute Step 7;

Step 6. Check the constraints. If it is not satisfied, update the skid's status as an unused skid, set the task number as $t = t + 1$, and return to step 5. Otherwise, continue to execute step 7;

Step 7. Set the decision variable $x_{ijk}^n = 1$, parts number $Y_k^n = \min\{6, R_{ij}, D_i^n\}$ and update $D_i^n = D_i^n - Y_k^n$ and $R_{ij} = R_{ij} - Y_k^n$;

Step 8. If $n < 8$, set $n = n + 1$, return to Step 4. Otherwise, set $n = 1$, set $k = k + 1$ and return to step 4.

We consider the production demands in Table A2 and the restrictions of the brackets in Table A3 in Appendix A. The total number of part types is 83, and 13,445 parts must be painted based on the production demands. Apply the RSA algorithm for this scheduling problem and obtain the results as shown in Table 4.

Table 4. The results of Rule-Based Scheduling Algorithm.

Work Type	Completed Tasks
Total painted part types	73/83
Total painted parts	11,699/13,445
Total color changes	56
Total brackets replacements per cycle	143

The average color changes and average brackets replacements per cycle are seven and 18, respectively. However, the total painted part types are 73, and the total painted parts are 11,699, which means the output is not maximized. Note that the unfinished tasks are all diamond white because diamond white must be arranged after the polar white, and the current priority mode cannot meet this constraint. To achieve the optimal schedule, we take the solution obtained in this subsection as an initial solution and design an Adaptive Partheno-Genetic algorithm to solve this scheduling problem.

4.2. Adaptive Partheno-Genetic Algorithm (APGA)

In this part, we design an adaptive Partheno-Genetic algorithm, a heuristic algorithm simulating single-parent reproduction. In the solution generation stage, the interval non-repetitive sampling method is adopted to generate individuals without any dimensional repetition. The unique methods are introduced in the mutation and crossover stages to ensure that all solutions are feasible in each genetic algorithm stage. Since all genetic operations are performed on one individual, the genetic operation process is simplified. Moreover, there is no premature convergence problem, and it has a good effect on solving the optimization problem in the research. The principles of the adaptive Partheno-Genetic algorithm are given as follows.

1. Fitness function
   The rule-based scheduling algorithm has minimized the number of color changes and the number of bracket replacements. Thus, the total part production (16) is selected as the fitness function;
2. Regeneration operator
   To prevent the optimal solution produced in the evolution process from being destroyed by the crossover operator or the mutation operator, set the elite individuals' proportion as 0.2, and copy the individuals with higher fitness directly to the next generation as part of the offspring population;
3. Mutation operator
   The mutation operator is the primary operator for maintaining the genetic algorithm population's diversity and prevent it from prematurely falling into a locally optimal solution. Let p be the mutation probability of an individual, then for any solution $x = (x_1, x_2, \cdots, x_m)$, the mutation operator includes the rearrangement operator and the rotation operator.
   (i) Rearrangement operator
   The rearrangement operator is defined as the random rearrangement of the individual x in all dimensions and expressed as
   $$x' = (x'_1, x'_2, \cdots, x'_m),$$
   where, $x'_1, x'_2, \cdots, x'_m$ is a random arrangement of $x_1, x_2, \ldots, x_m$.
   (ii) Rotation operator
   Rotation operator maps the individual x to a new individual x' expressed as
   $$x' = (x_k, x_{k+1}, \cdots, x_m, x_1, x_2, \cdots, x_{k-1}),$$
   where k is a random integer in $[0, m]$.
4. Probability exchange operator
   We introduce a probability exchange mechanism to avoid an infinite cycle. The exchange mechanism exchanges the positions of any two elements in an individual x and obtains the new individual x' as
   $$x' = (x_1, x_2, \cdots, x_{r-1}, x_k, x_{r+1}, \cdots, x_{k-1}, x_r, x_{k+1}, \cdots, x_m),$$
   where r and k are random integers in $[0, m]$. The probability of an exchange occurring is q.
5. Adaptive adjustment
   If all elite individuals have the same fitness, reduce the proportion of elites and increase the proportion of mutations to increase the mutation population. Set an adjustment scale factor $\gamma_1 < 1$ and the proportion of elites is $0.2\gamma_1$. Moreover, the ratio of mutations is $1 - 0.2\gamma_1$. If all individuals have the same fitness, a more radical mutation strategy is adopted. Reduce the proportion of elite individuals and increase

the ratio of the mutations. Set an adjustment scale factor $\gamma_2 < \gamma_1$, the proportion of elites and mutations are $0.2\gamma_2$ and $1 - 0.2\gamma_2$, respectively.

According to the above principles, the APGA algorithm is designed as follows.

Step 1. Set parameters of the population size, the maximum number of iterations, the probability of mutation, the probability of exchange, the expected optimal solution, and the adaptive adjustment coefficients;

Step 2. Generate the initial population and obtain the initial solution vector;

Step 3. Evaluate each individual through fitness function and sort them in order of fitness;

Step 4. If the number of iterations is equal to the maximum number of iterations or the current optimal fitness is better than the expected optimal solution, end this program and output the optimal result. Otherwise, go to Step 5;

Step 5. Utilize elite operator, mutation operator, and exchange operator to generate individual offspring.

Step 6. Adaptively adjust the proportion of the elite population and the proportion of the mutant population and return to Step 3.

For the real-life instance in this paper, model (21) is a multi-objective mixed-integer linear programme containing 829,684 variables and 137,319 constraints. This article will combine the above two algorithms (RSA + APGA) to optimize it. The initial solution is obtained by applying RSA and utilizing APGA to solve the optimal scheduling solution. Here, the mutation probability is 0.1, the exchange probability is 0.2, and the expected optimal solution is the default value. The adaptive adjustment parameters are 0.8 and 0.5, respectively. The stratified sequencing method (SSM) equipped in the Gurobi solver [14] is a multi-level objective optimization method, and we can apply it by Python. To compare with the results solved by SSM and demonstrate the effectiveness of the proposed methods, the RSA and APGA algorithms in this article are implemented by Python. The brief results are presented in Table 5.

Table 5. Results of maximizing production and minimizing the number of bracket replacements.

Algorithm	RSA + APGA	SSM	GA
Total painted part types	83	81	79
Total painted parts	13,445	13,413	13,405
Total color changes	63	155	29
Total brackets replacements	104	1988	1440
Time	575 s	6000 s	>5 h

The results show that the RSA + APGA algorithm can fully meet the output demand. Furthermore, it has realized the optimized color changes and bracket replacements. The optimization result of RSA + APGA takes 575 s calculation time. The proposed method is compared with the SSM and the genetic algorithm (GA). As shown in Table 5, neither of these two methods can fulfill the production demands, and the numbers of bracket replacements have not been optimized. Moreover, the SSM and GA algorithms have consumed more than 6000 s and 5 h, respectively.

5. Conclusions

We addressed a scheduling problem in a factory designed for the painting of automotive parts, in which more than ten thousand automotive parts can be painted in different colors. The problem has various important and practical characteristics, such as resource restriction, color sequence restriction, and part arrangement restrictions. These characteristics make the problem unique and have not been addressed previously. In this paper, a multi-objective mixed integer model was developed for the scheduling problem.

This model cannot be solved directly due to its large scale and complexity. Therefore, we designed a rule-based scheduling algorithm (RSA) to optimize color changes horizontally and bracket replacements vertically. Furthermore, we introduced the adaptive adjustment for the Partheno-Genetic algorithm to obtain the optimal scheduling solution based on the RSA. Finally, we applied the proposed method to an actual instance containing 83 types and 13,445 parts that needed to be painted with different colors. In this mathematical programming approach, there are 829,684 variables and 137,319 constraints. The model solution results showed that all production demands could be fulfilled. The total number of color changes was 63, and the brackets changed 104 times in eight cycles. Furthermore, the RSA + APGA algorithm consumed 575 s, exhibiting more computing efficiency than other methods.

In conclusion, the painting sequence arrangement model proposed in this paper has strong practicability and excellent effects, and can reasonably solve the problems raised here. This model can also be extended to other application areas and more general production resources.

Author Contributions: Conceptualization, J.Y. and Z.C.; methodology, K.P.; software, T.S.; validation, J.Y., T.S. and Z.C.; formal analysis, J.Y.; investigation, X.H.; resources, T.S.; data curation, T.S.; writing—original draft preparation, J.Y. and Z.C.; writing—review and editing, J.Y., T.S., X.H., K.P., Z.C., G.Q. and Z.Q.; visualization, Z.C.; supervision, J.Y.; project administration, J.Y.; funding acquisition, G.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Science and Technology Major Project (Grant. 2018AAA0102102) and the National Natural Science Foundation of China (Grant No. 61803152).

Institutional Review Board Statement: This study did not deal with animal or human subjects. Existing data were used for analysis. Therefore, the IRB statement is not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data, such as the all types of parts, the list of production demands, and the number of bracket for different types of parts is shown in Appendix A.

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

Appendix A

Table A1 is the all types of parts, Table A2 is the list of production demands, and Table A3 is the number of bracket for differet types of part.

Table A1. All types of parts.

Parts Name	Type	No.	Parts Name	Type	No.	Parts Name	Type	No.
Upper grille	A	1	Rear bumper	B	12	Threshold	C	23
	B	2		C	13		D	24
Middle spoiler	A	3		D	14		E	25
Front bumper	A	4		E	15	Threshold trim	A	26
	B	5		F	16	Radar bracket	A	27
	C	6		G	17		B	28
	D	7	Shell	A	18		C	29
	E	8	Rim decoration	A	19		D	30
	F	9		B	20		E	31
	G	10	Threshold	A	21			
Rear bumper	A	11		B	22			

Table A2. Demands list.

Part Name	Topcoat	Demands	Part Name	Topcoat	Demands
Upper grille A	Iridium Silver	135	Rear bumper E	Polar White	6
Upper grille B	Iridium Silver	150	Rear bumper F	Shining Blue	115
Middle spoiler A	Shining Blue	39		Ruby Red	101
	Obsidian Black	72		Obsidian Black	278
	Polar White	149		Milan Silver	41
	Milan Silver	76		Diamond White	323
Front bumper A	Shining Blue	249	Rear bumper G	Milan Silver	79
	Sapphire Blue	80		Diamond White	427
	Obsidian Black	212	Shell A	Shining Blue	13
	Polar White	885		Ruby Red	7
	Milan Silver	84		Sapphire Blue	18
	Iridium Silver	40		Polar White	87
Front bumper B	Sapphire Blue	27		Demim Blue	6
	Polar White	28		Milan Silver	3
	Iridium Silver	24	Rim decoration A	Shining Blue	51
Front bumper C	Shining Blue	12		Sapphire Blue	6
	Sapphire Blue	3		Polar White	168
	Obsidian Black	424	Rim decoration B	Demim Blue	4
	Diamond White	78		Diamond White	26
Front bumper D	Sapphire Blue	15	Threshold A	Shining Blue	204
	Obsidian Black	992		Ruby Red	351
Front bumper E	Shining Blue	115		Sapphire Blue	86
	Ruby Red	99		Obsidian Black	47
	Obsidian Black	373		Polar White	505
	Milan Silver	75		Milan Silver	83
	Diamond White	475		Iridium Silver	10
Front bumper F	Shining Blue	128	Threshold B	Sapphire Blue	94
	Polar White	961		Obsidian Black	579
Front bumper G	Universe Black	43		Diamond White	177
	Milan Silver	20	Threshold C	Obsidian Black	276
	Diamond White	121	Threshold D	Polar White	468
Rear bumper A	Shining Blue	174	Threshold E	Milan Silver	12
	Ruby Red	372	Threshold trim A	Iridium Silver	299
	Sapphire Blue	3	Radar bracket A	Ruby Red	5
	Polar White	565	Radar bracket B	Sapphire Blue	12
	Iridium Silver	35		Polar White	32

Table A2. *Cont.*

Part Name	Topcoat	Demands	Part Name	Topcoat	Demands
Rear bumper B	Milan Silver	75	Radar bracket C	Diamond White	26
Rear bumper C	Sapphire Blue	85	Radar bracket D	Obsidian Black	4
	Obsidian Black	808		Milan Silver	3
	Diamond White	87	Radar bracket E	Demim Blue	5
Rear bumper D	Obsidian Black	3		Diamond White	6
	Iridium Silver	11			

Table A3. The number of bracket for different types of part.

Part Name	Bracket Number	Part Name	Bracket Number
Upper grille A	34	Rear bumper G	131
Upper grille B	39	Shell A	52
Middle spoiler A	95	Rim decoration A	63
Front bumper A	400	Rim decoration B	15
Front bumper B	28	Threshold A	341
Front bumper C	141	Threshold B	219
Front bumper D	255	Threshold C	69
Front bumper E	297	Threshold D	118
Front bumper F	280	Threshold E	3
Front bumper G	51	Threshold trim A	76
Rear bumper A	304	Radar bracket A	5
Rear bumper B	21	Radar bracket B	16
Rear bumper C	251	Radar bracket C	9
Rear bumper D	11	Radar bracket D	10
Rear bumper E	3	Radar bracket E	11
Rear bumper F	225		

References

1. Abdulrahman, M.D.-A.; Subramanian, N.; Liu, C.; Shu, C. Viability of remanufacturing practice: A strategic decision making framework for chinese auto-parts companies. *J. Clean. Prod.* **2015**, *105*, 311–323. [CrossRef]
2. Wang, J.; Li, J.; Huang, N. Optimal scheduling to achieve energy reduction in automotive paint shops. In Proceedings of the ASME 2009 International Manufacturing Science and Engineering Conference (MSEC2009), West Lafayette, IN, USA, 4 October 2009; pp. 161–167.
3. Epping, T.; Hochstattler, W.; Oertel, P. Complexity results on a paint shop problem. *Discret. Appl. Math.* **2004**, *136*, 217–226. [CrossRef]
4. Winter, F.; Musliu, N. Constraint-based Scheduling for Paint Shops in the Automotive Supply Industry. *ACM Trans. Intell. Syst. Technol.* **2021**, *12*, 1–25. [CrossRef]
5. Meunier, F.; Neveu, B. Computing solutions of the paintshop–necklace problem. *Comput. Oper. Res.* **2012**, *39*, 2666–2678. [CrossRef]
6. Spieckermann, S.; Gutenschwager, K.; Voß, S. A sequential ordering problem in automotive paint shops. *Int. J. Prod. Res.* **2004**, *42*, 1865–1878. [CrossRef]
7. Sun, H.; Fan, S.; Shao, X.; Zhou, J. A colour-batching problem using selectivity banks in automobile paint shops. *Int. J. Prod. Res.* **2015**, *53*, 1124–1142. [CrossRef]
8. Sun, H.; Han, J. A study on implementing color-batching with selectivity banks in automotive paint shops. *J. Manuf. Syst.* **2017**, *44*, 42–52. [CrossRef]

9. Singgih, I.K.; Yu, O.; Kim, B.I.; Koo, J.; Lee, S. Production scheduling problem in a factory of automobile component primer painting. *J. Intell. Manuf.* **2020**, *31*, 1483–1496. [CrossRef]
10. Winter, F.; Musliu, N.; Demirovic, E.; Mrkvicka, C. Solution approaches for an automotive paint shop scheduling problem. *Proc. Conf. Autom. Plan. Sched.* **2019**, *29*, 573–581.
11. Garey, M.R.; Johnson, D.S. *Computers and Intractability. A Guide to the Theory of NP-Completeness*; Freeman: New York, NY, USA, 1979.
12. Korte, B.; Vygen, J. *Combinatorial Optimization: Theory and Algorithms*; Springer: Berlin, Germany, 2008.
13. Feng, L.; Chunhua, T.; Hao, Z.; Wade, K. Rule-based optimization approach for airline load planning system. *Procedia Comput. Sci.* **2010**, *1*, 1455–1463.
14. LLC Gurobi Optimization. Gurobi Optimizer Reference Manual. 2021. Available online: https://www.gurobi.com/documentation/9.1/refman/refman.html (accessed on 19 September 2021).