Constraint Based Modeling for Scheduling Paint Shops in the Automotive Supply Industry

Felix Winter, Nysret Musliu

Christian Doppler Laboratory for Artificial Intelligence and
Optimization for Planning and Scheduling
TU Wien
Favoritenstraße 9-11, 1040 Vienna, Austria

{winter,musliu}@dbai.tuwien.ac.at

May 21, 2019

CD-TR 2019/1

Constraint Based Modeling for Scheduling Paint Shops in the Automotive Supply Industry[*]

Felix Winter and Nysret Musliu

Christian Doppler Laboratory for Artificial Intelligence and Optimization for Planning and Scheduling
Institute for Logic and Computation, DBAI, TU Wien
{winter,musliu}@dbai.tuwien.ac.at

Abstract. Every day, the factories of the automotive supply industry have to process the painting of large amounts of items that are requested by car manufacturing companies. Because of the many complex constraints and optimization objectives, finding a good schedule becomes a challenging task in practice and currently several full time employees are required to manually create feasible production plans.
In this paper we propose a novel constraint programming model for a real life paint shop scheduling problem. Using this model, we compare results produced by state of the art solvers using a collection of real life based instances. Additionally, we show that the decision variant of the problem is NP-complete.

Keywords: Constraint Programming · Paint Shop Scheduling · Exact Methods · NP-complete

1 Introduction

Every day, the paint shops of the automotive supply industry will paint a large number of items that are requested by car manufacturing companies. To ensure a cost efficient production, modern factories will utilize a high level of automation that includes multiple painting robots and conveyor belt systems. Because of the sophisticated production process it becomes a hard task to find good painting schedules, and human planners are usually not able to find optimized production sequences. Therefore, there is a strong need to develop automated techniques for paint shop scheduling.

In the literature related problems have been studied and several publications consider the minimization of color changes for paint shop scheduling (e.g. [9], [8], [7], [2]). However, the problem we investigate in this paper includes additional important practical features like the optimized allocation of materials onto carrying devices and the consideration of many sequence and resource constraints.

[*] The financial support by the Austrian Federal Ministry for Digital and Economic Affairs and the National Foundation for Research, Technology and Development is gratefully acknowledged.

We have previously introduced this real life paint shop scheduling problem that appears in the automotive industry [11]. Its aim is to find a production sequence that fulfills a large set of given demands and to minimize the number of color changes as well as the number of carrying devices that are used to carry materials through the paint shop. To solve the problem, we previously proposed a greedy algorithm as well as a local search based approach and we provided a set of practical benchmark instances in [11]. However, up to now no exact solution approaches have been described and no optimal solutions could be provided.

In this paper we investigate constraint programming (CP) modeling techniques to solve the paint shop scheduling problem. We describe two different modeling techniques that can be used to efficiently formulate the problem. One of them using a direct modeling approach and the other one using deterministic finite automatons (DFAs). Furthermore, we evaluate and compare our proposed modeling techniques by performing a series of benchmark experiments using state of the art solvers on known practical paint shop scheduling benchmark instances. Although currently the exact methods we describe cannot be used to solve very large practical instances, the proposed approaches can provide optimal solutions for 7 benchmark instances that have been previously unknown. Additionally, we analyze the complexity of the paint shop scheduling problem and show that it is NP-complete.

In the next section we give a short description of the paint shop scheduling problem as it appears in practice. Afterwards we will provide a formal description of the problem's input and describe our modeling techniques. Later we will present the NP-completeness proof and discuss experimental results before we give concluding remarks at the end of the paper.

2 The Paint Shop Scheduling Problem

The aim of the paint shop scheduling problem we investigate in this paper is to determine an optimized production sequence that schedules the painting for a number of raw material items within given due dates. To understand the details behind the problem's optimization objective, one needs to know that all items scheduled for painting have to be placed on custom carrier devices that will move through the paint shop's painting cabins. Due to the fact that there are many different carrier types available, each being able to transport certain configurations of demanded materials, it will be necessary to use a variety of different carrier device types during production. Although combinations of different raw material items may be transported by a single carrier, it is never possible to schedule products that should be painted with different colors on a single carrying device.

The paint shops of the automotive supply industry are designed to support an almost fully automated production process. Therefore, any scheduled carrying devices will be automatically moved through the paint shop on a circular conveyor belt system. Carriers can be inserted and removed from and onto the conveyor belt at two carrier gates. One of the gates is used to insert carrying devices, while the other one can be used to remove carriers from the circular conveyor belt system. Once a carrier has been inserted, it will be moved through the cyclic paint shop system where it repeatedly will pass by the painting cabins, the carrier gates, and a material gate, until the schedule will select the carrier for ejection at the output gate. At the material gate unpainted raw materials may be placed on any empty carrying device by paint shop employees. A loaded carrier will then move to the painting cabins, where the scheduled color will be applied on all carried items. Whenever a loaded carrier arrives at the material gate after having completed a full round, another employee will take off the colored material pieces and may place new uncolored raw materials onto the carrier that will then be painted in the succeeding round.

Figure 1 shows a schematic of the paint shop's layout and visualizes the movement of carriers through the paint shop.

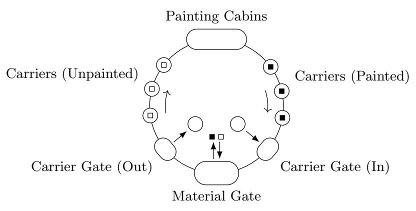

Fig. 1: Schematic showing a paint shop layout that is commonly used in the automotive supply industry.

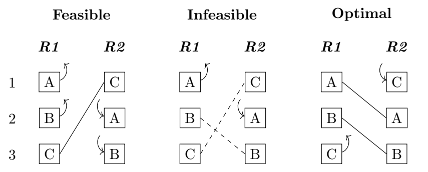

Fig. 2: Three possible options to reuse carriers between two consecutive rounds. The feasible option on the left side of the figure will reuse only a single carrier of type C and requires a total of two carrier insertions and two carrier removals. The infeasible option shown in the middle of the figure suggests to keep carriers of type B and C between two consecutive rounds, however this is technically not possible. The option shown on the right side of the figure requires the fewest number of carrier insertions and removals for this example.

Because of the circular layout of the paint shop, the painting schedule is organized in rounds. Within each painting round, several carrier units will be painted one after the other in a sequence that is predetermined by the schedule. We represent a candidate solution to the paint shop scheduling problem as a table, where each column represents the scheduling sequence for a single round. Each table cell will then assign the carrier type, material configuration, and color that should be scheduled in the associated round sequence. Considering all carrier configurations and colors that can be scheduled for production, usually a tremendous number of different schedules can be created, however many constraints impose restrictions regarding due dates and allowed carrier sequences have to be fulfilled on feasible schedules. A multi-objective minimization function further includes two optimization criteria. The first optimization goal is to minimize color changes in the scheduling sequence, while the second optimization goal is concerned with an efficient utilization of carrying devices. In the following we will further explain the second minimization goal.

Since a paint shop schedule will usually not use the same carrier type sequence in each round, it is often required to remove and insert the carriers from the conveyor belt system between rounds. However, if carriers of the same type are scheduled in two consecutive rounds it may be possible to reuse some of them as long as the sequence of kept carriers is compatible with the scheduled carrier sequence in the succeeding round. Since the insertion and removal of carrier units from the circular track might lead to delays and can in general not be done in parallel, it is desired to keep the number of such operations as low as possible. Note that for any given two consecutive rounds, the minimal amount of required carrier insertions and removals can be calculated by determining the longest common subsequence (LCS) [10] of the two carrier type round sequences. Figure 2 visualizes three alternative ways how carriers may be reused between consecutive rounds.

3 Modeling the Problem with Constraint Programming

In this section we describe the problem's input parameters and afterwards propose a CP model for the paint shop scheduling problem.

3.1 Input parameters

The following parameters describe instances of the problem:

Number of carrier configurations: $k \in \mathbb{N}$

Set of carrier configurations: $K = \{1, \dots, k\}$

Number of carrier types: $t \in \mathbb{N}$

Set of carrier types: $T = \{1, \dots, t\}$

Number of colors: $c \in \mathbb{N}$

Set of colors: $C = \{1, \dots, c\}$

Number of materials: $m \in \mathbb{N}$

Set of materials: $M = \{1, \dots, m\}$

Number of rounds to schedule: $r \in \mathbb{N}$

Set of all rounds to schedule: $R = \{1, \dots, r\}$

Maximum number of carrier positions per round: $s \in N$

Set of carrier positions per round: $S = \{1, \dots, s\}$

Minimum number of carriers that have to be scheduled in each round: $q \in \mathbb{N}_{>0}$

Number of available carriers of type t in round r:

$$a_{r,t} \in \{1, \dots, s\}, \forall r \in R, t \in T$$

Number of demands: $d \in \mathbb{N}$

Set of demands: $D = \{1, \dots, d\}$

Number of requested items per demand: $a_d \in \mathbb{N}_{>0}, \forall d \in D$

Material type of demand: $m_d \in M, \forall d \in D$

Due round of demand: $r_d \in \mathbb{N}_{>0}, \forall d \in D$

Color of demand: $c_d \in C, \forall d \in D$

Number of pieces of material type m that can be placed on configuration k:

$$u_{k,m} \in \mathbb{N}, \forall k \in K, m \in M$$

Carrier type of each carrier configuration (v_0 will be set to 0):

$$v_x \in \{0, \dots, t\}, \forall x \in \{0, \dots, k\}$$

Number of skids scheduled in the round previous to the scheduling horizon (history round): $p \in \mathbb{N}$

Carrier type of the scheduled carrier at position i of the history round:

$$pt_i \in T, \forall i \in \{1, \dots, p\}$$

Used color at position i of the history round: $pc_i \in C \forall i \in \{1, \dots, p\}$

Number of forbidden carrier type sequences of length two: $f \in \mathbb{N}$

Forbidden carrier type sequences: $F = \{1, \dots, f\}$

First carrier type of forbidden sequence f: $t^1_f \in T, \forall f \in F$

Second carrier type of forbidden sequence f: $t^2_f \in T, \forall f \in F$

Minimum block size of consecutive carriers with type t:

$$b^{\min}_t \in \mathbb{N}_{>0}, \forall t \in T$$

Whenever a carrier of type t is scheduled, the same carrier type has to be used for the next consecutive carriers until the given minimum block length is reached. (For example let $b^{\min}_{t_1} = 3$ and the previously scheduled carrier type sequence be $\langle t_3, t_3, t_2, t_1 \rangle$, then to satisfy the minimum block length at least the next two carriers in the sequence have to be t_1).

Maximum block size of consecutive carriers with type t:

$$b_t^{\max} \in \mathbb{N}_{>0}, \forall t \in T$$

Number of forbidden color sequences: $o \in \mathbb{N}$

Set of forbidden color sequences: $O = \{1, \ldots, o\}$

First color of forbidden color sequence o: $c_o^1 \in C, \forall o \in O$

Second color of forbidden color sequence o: $c_o^2 \in C, \forall o \in O$

The number of carriers that have to be painted in a different color before a switch from color c^1 to color c^2 becomes legal for sequence o:

$$j_o \in \mathbb{N}_{>0}, \forall o \in O$$

For example let $o_{v,w} = 3$ for colors v and w. Then the color sequences $\langle v, w\rangle$ and $\langle v, y, w\rangle$ would be illegal while the color sequence $\langle v, y, y, y, w\rangle$ would be legal (assuming that $y \neq v$ and $y \neq w$).

Color transition costs for all pairs of colors: $f_{c_1,c_2} \in \mathbb{N}, \forall c_1, c_2 \in C$

3.2 Decision Variables

Scheduled carrier configuration in round i and position j:

$$x_{i,j} \in \{0, \ldots, k\}, \forall i \in R, j \in S$$

If the value 0 is assigned, the position is empty and no carrier will be scheduled at the position.

Scheduled color configuration in round i and position j:

$$y_{i,j} \in \{0, \ldots, c\}, \forall i \in \{0, \ldots, r\}, j \in S$$

If the value 0 is assigned, the position is empty and will not be painted.

3.3 Helper Variables

Number of scheduled carriers per round: $p_i \in \{0, \ldots, s\}, \forall i \in \{0, \ldots, r\}$

Number of totally scheduled carriers: $ps \in \{0, \ldots, s \cdot r + p\}$

Sequence variables that will convert a given round index i and position index j into a one dimensional position index:

$$seq_{i,j} \in \{0, \ldots, s \cdot r + p\}, \forall i \in \{0, \ldots, r\}, j \in S$$

For example let exactly 100 carriers be scheduled in round 1 and the length of the history round p be 5, then $seq_{2,3}$ will be set to the value 108. $seq_{i,j}$ will be set to 0 if and only if no carrier is scheduled at position j in round i.

3.4 Hard Constraints

1. Bind the correct number of scheduled skids to the associated helper variables:

$$\begin{gathered} p_0 = p \\ p_i = \sum_{\{j \in S | x_{r,j} \neq 0\}} 1 \quad \forall i \in R \\ ps = \sum_{i \in \{0,\ldots,r\}} p_i \end{gathered} \tag{1}$$

2. Set correct values to sequence variables:

$$\begin{gathered} seq_{0,j} = j \quad \forall j \in \{1,\ldots,p\} \\ seq_{0,j} = 0 \quad \forall j \in \{p+1,\ldots,s\} \\ seq_{i,1} = p + \sum_{z \in \{2,\ldots,i\}} p_{z-1} \quad \forall i \in R \\ seq_{i,j} = seq_{i,j-1} + 1 \quad \forall i \in R, j \in \{2,\ldots,s\} \text{ where } x_{i,j} \neq 0 \\ seq_{i,j} = 0 \quad \forall i \in R, j \in S \text{ where } x_{i,j} = 0 \end{gathered} \tag{2}$$

3. Unplanned carrier positions should always be scheduled last in a round:

$$(x_{i,j} = 0) \Rightarrow (x_{i,j+1} = 0) \quad \forall i \in R, j \in \{1,\ldots,s-1\} \tag{3}$$

4. Any scheduled carrier position must also assign a color and any unscheduled position must not assign a color:

$$(x_{i,j} \neq 0) \Leftrightarrow (c_{i,j} \neq 0) \quad \forall i \in R, j \in S \tag{4}$$

5. All demands must be satisfied in time (overproduction is allowed):

$$\begin{gathered} \sum_{\{d \in D |\ m_d = m \wedge r_d <= r \wedge c_d = c\}} a_d \leq \sum_{\{x_{i,j} | i \in \{1,\ldots,r\} \wedge j \in \{1,\ldots,s\} \wedge y_{i,j} = c\}} u_{(x_{i,j}),m} \\ \forall r \in R, m \in M, c \in C \end{gathered} \tag{5}$$

6. Carrier availabilities must be respected in each round:

$$\sum_{\{j | j \in S \wedge v_{(x_{r,j})} = t\}} 1 \leq a_{r,t} \quad \forall r \in R, t \in T \tag{6}$$

7. The minimum round capacity must be fulfilled in each round:

$$p_r >= q, \forall r \in R \tag{7}$$

8. Forbidden carrier type sequences must not appear in the schedule:

$$\begin{gathered} (v_{(x_{i,j})} \neq t_f^1) \vee (v_{(x_{i,j+1})} \neq t_f^2) \quad \forall f \in F, i \in R, j \in \{1,\ldots,s-1\} \text{ where } j < p_i \\ (v_{(x_{i,(p_i)})} \neq t_f^1) \vee (v_{(x_{i+1,1})} \neq t_f^2) \quad \forall f \in F, i \in \{1,\ldots,r-1\} \\ (pt \neq t_f^1) \vee (v_{(x_{1,1})} \neq t_f^2) \quad \forall f \in F \end{gathered} \tag{8}$$

9. Minimum carrier type block size restrictions must be fulfilled[1]:

[1] For simplicity we omit an additional corner case that has to be regarded: The last carrier type and color that appears in the history round also needs to be checked regarding the sequence constraints. This can simply be modeled by adding additional constraints for the history round.

$$\begin{gathered}
\bigwedge_{z\in\{j+2,\ldots,s\}} (seq_{i,z}=0 \vee seq_{i,z} \geq seq_{i,j+1}+b_t^{\min} \vee v_{(x_{i,z})}=t) \wedge \\
\bigwedge_{y\in\{i+1,\ldots,r\},z\in S} (seq_{y,z}=0 \vee seq_{y,z} \geq seq_{i,j+1}+b_t^{\min} \vee v_{(x_{y,z})}=t) \wedge \\
\left(\bigvee_{z\in\{j+1,\ldots,s\}} (seq_{i,z}=seq_{i,j}+b_t^{\min} \wedge v_{(x_{i,z})}=t) \vee \right. \\
\left. \bigvee_{y\in\{i+1,\ldots,r\},z\in S} (seq_{y,z}=seq_{i,j}+b_t^{\min} \wedge v_{(x_{y,z})}=t) \right) \\
\forall t\in T, i\in R, j\in\{1,\ldots,s-1\} \text{ where } j<p_i \wedge v_{(x_{i,j})}\neq t \wedge v_{(x_{i,j+1})}=t
\end{gathered} \tag{9}$$

$$\begin{gathered}
\bigwedge_{z\in\{2,\ldots,s\}} (seq_{i+1,z}=0 \vee seq_{i+1,z} \geq seq_{i+1,1}+b_t^{\min} \vee v_{(x_{i+1,z})}=t) \wedge \\
\bigwedge_{y\in\{i+2,\ldots,r\},z\in S} (seq_{y,z}=0 \vee seq_{y,z} \geq seq_{i+1,1}+b_t^{\min} \vee v_{(x_{y,z})}=t) \\
\left(\bigvee_{z\in\{1,\ldots,s\}} (seq_{i+1,z}=seq_{i+1,1}+b_t^{\min}-1 \wedge v_{(x_{i+1,z})}=t) \vee \right. \\
\left. \bigvee_{y\in\{i+2,\ldots,r\},z\in S} (seq_{y,z}=seq_{i+1,1}+b_t^{\min}-1 \wedge v_{(x_{y,z})}=t) \right) \\
\forall t\in T, i\in\{1,\ldots,r-1\} \text{ where } v_{(x_{i,p_i})}\neq t \wedge v_{(x_{i+1,1})}=t
\end{gathered} \tag{10}$$

10. Maximum carrier type block size restrictions must be fulfilled[1]:

$$\begin{gathered}
\bigvee_{z\in\{j+1,\ldots,s\}} (seq_{i,z}>seq_{i,j} \wedge seq_{i,z} \leq seq_{i,j}+b_t^{\max} \wedge v_{(x_{i,z})}\neq t) \vee \\
\bigvee_{y\in\{i+1,\ldots,r\},z\in S} (seq_{y,z}>seq_{i,j} \wedge seq_{y,z} \leq seq_{i,j}+b_t^{\max} \wedge v_{(x_{y,z})}\neq t) \\
\forall t\in T, i\in R, j\in S, \text{ where } j\leq p_i \wedge v_{(x_{i,j})}=t \wedge seq_{i,j} =\leq ps-b_t^{\max}
\end{gathered} \tag{11}$$

11. No forbidden color sequences should occur in the schedule[1]:

$$\begin{gathered}
\bigwedge_{z\in\{j+1,\ldots,s\}} (seq_{i,z}=0 \vee seq_{i,z}>seq_{i,j}+j_o \vee y_{i,z}\neq c_o^2) \wedge \\
\bigwedge_{x\in\{i+1,\ldots,r\},z\in S} (seq_{x,z}=0 \vee seq_{x,z}>seq_{i,j}+j_o \vee y_{x,z}\neq c_o^2) \\
\forall o\in O, i\in R, j\in S \text{ where } j\leq p_i \wedge y_{i,j}=c_o^1
\end{gathered} \tag{12}$$

3.5 Helper Variables for the Objective Function

The amount of color change costs occurring in round r of the schedule:

$$cc_r \in \mathbb{N}, \forall r\in R$$

The number of required carrier type changes between round r and $r+1$:

$$sc_r \in \{0,\ldots,s\cdot 2\}, \forall r\in\{0,\ldots,n-1\}$$

The number of carriers that will be reused between round r and round $r+1$:

$$sk_r \in \{0,\dots,s\}, \forall r \in \{0,\dots,n-1\}$$

Information on the position of the kept carrier sequence in the next/previous round:

$$kept^1_{i,j} \in \{0,\dots,s\}, \forall i \in \{0,\dots,r-1\}, j \in S$$
$$kept^2_{i,j} \in \{0,\dots,s\}, \forall i \in R, j \in S$$

3.6 Hard Constraints for Objective Function

Calculate color changes per round: [2]

$$cc_i = \sum_{j \in \{1,\dots,s-1\}} f_{(y_{i,j}),(y_{i,j+1})} + f_{(y_{i-1,p_i-1}),(y_{i,1})} \quad \forall i \in R \tag{13}$$

All kept carrier type sequences between consecutive rounds have to be legal: [3]
In Section 2 we have described that one can visualize which carrier types will be reused between two consecutive rounds by drawing edges that connect the associated positions. We initially experimented with a modeling approach that introduces variables for all possible edges and tries to maximize the number of selected edges without causing any edge crossings. However, using this model it was not possible to find the optimal number of edges for two given rounds of practical size within a time limit of one hour. Therefore, we instead propose a modeling approach that introduces variables that store the positions of all reused carriers.

$$\begin{aligned} kept^1_{i,j} > kept^1_{i,j-1} \quad &\forall i \in \{0,\dots,r-1\}, i \in \{2,\dots,s\} \text{ where } kept^1_{i,j} \neq 0 \\ kept^2_{i,j} > kept^2_{i,j-1} \quad &\forall i \in R, i \in \{2,\dots,s\} \text{ where } kept^2_{i,j} \neq 0 \end{aligned} \tag{14}$$

$$\begin{gathered} kept^2_{1,j} \leq p \quad \forall j \in S \text{ where } kept^2_{1,j} \neq 0 \\ kept^1_{i,j} \leq p_{i+1} \quad \forall i \in \{0,\dots,r-1\}, j \in S \text{ where } kept^1_{1,j} \neq 0 \\ kept^2_{i,j} \leq p_{i-1} \quad \forall i \in \{2,\dots,r\}, j \in S \text{ where } kept^2_{1,j} \neq 0 \\ kept^1_{0,j} = 0 \wedge kept^2_{1,j} = 0 \quad \forall j \in \{p+1,\dots,s\} \\ kept^1_{i,j} = 0 \wedge kept^2_{i+1,j} = 0 \quad \forall i \in \{1,\dots,r-1\}, j \in S \text{ where } j > p_i \\ kept^1_{i,j} > 0 \wedge kept^2_{i+1,j} > 0 \quad \forall i \in \{0,\dots,r-1\}, j \in S \end{gathered} \tag{15}$$

$$\begin{gathered} pt_{(kept^2_{1,j})} = v_{(x_{1,(kept^1_{0,j})})} \quad \forall j \in S \text{ where } kept^2_{1,j} \neq 0 \\ v_{(x_{i,(kept^2_{i+1,j})})} = v_{(x_{i+1,(kept^1_{i,j})})} \quad \forall i \in \{1,\dots,r-1\}, j \in S \text{ where } kept^1_{i,j} \neq 0 \end{gathered} \tag{16}$$

[2] We assume that color costs from and to 0 will always be 0

[3] For simplicity we omit a special condition that handles the corner case of an empty history round. In this case one can simply add a constraint that forces all carriers of round 1 to be inserted if $p = 0$.

Calculate the number of reused carriers after each round:

$$sk_i = \sum_{\{j | j \in S \wedge kept^1_{i,j} \neq 0\}} 1 \quad \forall i \in \{0, \dots, r-1\} \tag{17}$$

Count the total number of required carrier changes between two rounds:

$$sc_i = p_i - sk_i + p_{i+1} - sk_i \quad \forall i \in \{0, \dots, r-1\} \tag{18}$$

3.7 Objective Function

The objective function of the paint shop scheduling problem aims to minimize the number of carrier changes (sc) and color change costs (cc) per round. The sums are squared, since it is preferable to distribute the required changes over the scheduling horizon and to avoid peaks of many changes within single rounds.

$$\min \sum_{i \in R} cc_i^2 + \sum_{i \in \{0,\dots,r-1\}} sc_i^2 \tag{19}$$

4 Modeling the Problem with DFAs

In this section, we propose a different way to model the sequence constraints by using DFAs. All automatons will process either the total sequence of scheduled carrier types or the total sequence of scheduled colors. We can provide the total carrier type or color sequence in our model by simply concatenating the values of all two indexed decision variables ($x_{i,j}$ or $y_{i,j}, \forall i \in R, j \in S$) into a one dimensional list. The automatons can then be used to check whether or not the total color or carrier type sequence can be accepted.

Forbidden carrier type sequences: For each forbidden carrier type sequence $f \in F$ we model an automaton that processes the total sequence of scheduled carrier types. Figure 3 shows how automatons can be constructed to check that no forbidden carrier type sequence occur in the schedule.

Minimum carrier type block sizes: For each carrier type $t \in T$ we model an automaton that processes the total sequence of scheduled carrier types. Figure 4 shows how automatons can be constructed to check the minimum carrier type block size constraint.

Maximum carrier type block sizes: For each carrier type $t \in T$ we model an automaton that processes the total sequence of scheduled carrier types. Figure 5 shows how automatons can be constructed to check the maximum carrier type block size constraint.

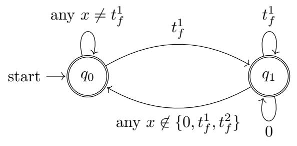

Fig. 3: Automaton that will be generated for each carrier type sequence $f \in F$ to check the forbidden carrier type sequence constraint. q_0 will accept any carrier type. State q_1 will be entered whenever the first carrier type of the forbidden sequence (t_f^1) is encountered and will not accept the second type (t_f^2) before any other type is encountered. Both states will be legal final states.

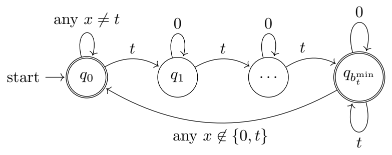

Fig. 4: Automaton that will be constructed for each carrier type $t \in T$ to check the minimum carrier block type size constraint. q_0 will accept any carrier type that is different to t. States q_1– $q_{b_t^{\min}}$ will be used to count consecutive assignments of carrier type t. States q_0 and $q_{b_t^{\min}}$ are the only legal final states.

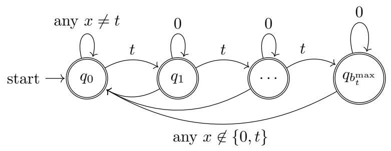

Fig. 5: Automaton that will be generated for each carrier type $t \in T$ to check the maximum carrier block type size constraint. q_0 will accept any carrier type that is different to t. States q_1– $q_{b_t^{\max}}$ will be used to count consecutive assignments of carrier type t. All states will be legal final states.

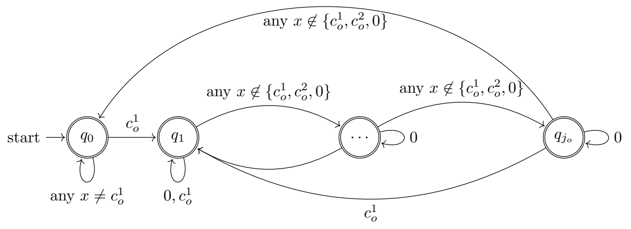
any $x \notin \{c_o^1, c_o^2, 0\}$
any $x \notin \{c_o^1, c_o^2, 0\}$
any $x \notin \{c_o^1, c_o^2, 0\}$
start
q_0
c_o^1
q_1
$\cdots$
0
q_{j_o}
0
any $x \neq c_o^1$
$0, c_o^1$
c_o^1

Fig. 6: Automaton that will be generated for each forbidden color sequence $o \in O$ to check the forbidden color sequence constraint. q_0 will accept any color assignment that is different to c_o^1. States q_1– q_{j_o} will be used to assert that an assignment of color c_o^2 may only occur if color c_o^1 has not been encountered within the previous j positions. All states will be legal final states.

Forbidden color sequences: For each forbidden color sequence $o \in O$ we model an automaton that processes the total sequence of scheduled colors. Figure 6 shows how automatons can be constructed to check the forbidden color sequences constraint.

5 Complexity Results

In this section we show that the decision variant of the paint shop scheduling problem (which asks whether or not a feasible schedule with an objective value $\leq t$ can be found) is NP-complete. We prove the following:

Theorem 1. *The decision variant of the paint shop scheduling problem is NP-complete.*

Proof. Firstly, we provide a polynomial time reduction from the set cover problem [4] to the paint shop scheduling problem.

Let we have given an arbitrary instance of set cover problem consisting of the universe of elements $U = \{1, \dots, w\}$, an integer k, and a set S that denotes the collection of z sets. The union of all sets in S is equal the universe $U = \{1, \dots, w\}$. The question is whether there is a set covering of size k or less.

We construct an instance of the paint shop scheduling problem can then be constructed as follows: We set the scheduling horizon to a single round ($r = 1$) and set $C = \{1\}$ as only one color should be considered for scheduling. The set of materials M will be set to match all items of the universe $M = \{1, \dots, w\}$. The maximum number of allowed carrier devices per round will be set to $s = k$, while the minimum number of required carrier devices per round will be set to $q = 1$. We only consider a single carrier type and therefore set $T = \{1\}$. The set of all demands D that need to be scheduled will be set to $D = \{(1, i, 1, 1) | i \in \{1, \dots, w\}\}$ (we want to schedule each material exactly once until round 1). We further create z carrier configurations, each one corresponding to a single set in S: $K = \{1, \dots, z\}$. All configurations will belong to the same carrier type and we therefore set $v_k = 1, \forall k \in K$. The materials contained within each configuration should equal all elements contained in the associated set and we therefore set $\forall x \in K, m \in M$:

$$u_{x,m} = \begin{cases} 1, & \text{if } m \text{ is contained in the associated set of configuration } x \\ 0, & \text{otherwise} \end{cases}$$

Furthermore, we set the number of carriers in the history round to $p = 0$. Finally, we set all color costs to 0 and disable all sequence dependent hard constraints as well as the carrier availability constraint by setting $f_c(1, 1) = 0, b_1^{\min} = 1, b_1^{\max} = s, o = 0, f = 0$ and $a_{1,1} = s$.

We now prove the following:

Claim. There exists a set cover of size k or less if and only if there exists a feasible paint shop schedule with total costs lower or equal to k^2.

Since we set all color transition costs to 0 and the history round contains 0 carriers, the objective function of the paint shop scheduling problem will be equal to the squared number of scheduled carriers in round 1 (any carrier has to be inserted). As we have set $s = k$, it is easy to see that any feasible paint shop schedule which fulfills the maximum round capacity hard constraint will have an objective value $\leq k^2$. Furthermore, since we have disabled all hard constraints except the demand constraint in the paint shop scheduling instance, any schedule that satisfies all demands will be a feasible schedule.

Let S be a set cover using k' sets (where $k' \leq k$), we then know that all elements of the universe U are contained in at least one of selected the sets. The demand constraint was constructed in such a way that each element of the universe has to be scheduled at least once in round 1 and we also know that for each set $s \in S$ there exists a carrier configuration that carries each element in s exactly once. Therefore, there exists a set of k' carrier configurations that can be scheduled in any sequence to fulfill all demands.

Now we prove the opposite direction. Let P be a feasible paint shop schedule. We know that any material is demanded to be scheduled exactly once in round 1. Therefore, it must be possible to remove any repeatedly used configurations from P in such a way that each carrier configuration which is scheduled in P is used exactly once without violating the demand constraint. As in our reduction we set $s = k$, we know that we have k' (where $k' \leq k$) carriers in the schedule. Given such a schedule that uses k' carriers, we can construct a feasible set covering of size k' by using the sets which correspond to the configurations used in P.

Finally we show that the decision variant of the paint shop scheduling problem is in NP. Suppose that we have given a candidate solution P. We show below that we can verify in polynomial time if P is a feasible solution to the problem.

The number of carriers in P cannot be larger than $s \cdot r$ (see input parameters in Section 3) and we can simply check the carrier availability and round capacity constraints by counting the number of carriers in each round. Similarly, the sequence constraints (minimum/maximum block length of consecutive carrier types, forbidden carrier type and color sequences) can be checked by iterating over the scheduled sequence. Furthermore, we can check the demand constraint by verifying that Equation 5 holds. We have to perform not more than $r \cdot m \cdot c$ comparisons to check this equation. For each comparison we have to consider at most d demands and $s \cdot r$ positions to calculate the sums.

To calculate the total color change costs of P we iterate over the scheduled sequence, similar as we did for the sequence dependent hard constraints. Finally, we have to determine the maximum number of carriers that can be reused between any two consecutive rounds in the schedule to calculate the total carrier change costs. As already mentioned in an earlier section, this can be achieved by solving the corresponding longest common subsequence problem for each pair of consecutive rounds (polynomial time algorithms to solve the LCS problem have been described in [10]). $\square$

6 Empirical Evaluation

We evaluated our models using the set of benchmark instances that we have previously provided in [11]. This collection of benchmark instances includes 24 instances that are based on real life planning scenarios. Instances 1–12 include small to medium sized instances, whereas instances 13–24 describe very large problems. Both, the set of small to medium instances as well as the set of very large instances describe six different planning horizons of 7, 20, 50, 70, 100 and 200 rounds (two instances for each horizon). We could not solve any of the very large instances using the exact approaches described in this paper, since all used solvers went out of memory before any feasible solution could be achieved. Therefore, in the following we describe our experiments with instances 1–12.

We implemented the models proposed in this paper using the MiniZinc [6] modeling language together with current versions of Chuffed [1], Gurobi [5] and Cplex [3]. Initially we performed benchmark experiments under a time limit of one hour on an Intel Core i7-8550U 1.80 Ghz CPU with 16 GB RAM to compare the results produced by the direct modeling approach and the DFA model. Our initial experiments showed that the DFA based model produced the best results for all instances. Therefore, we decided to further investigate different search strategies with the DFA model and evaluated several combinations of variable and value selection strategies by setting the associated MiniZinc annotations (we considered *first fail*, *smallest*, and *input order* for variable selection and *indomain min* as well as *indomain split* for value selection). For Chuffed we additionally activated the free search parameter which allows the solver to alternate between the given search strategy and its default one on each restart. Our final experiments have been conducted using an Intel Xeon E5345 2.33 GHz CPU with 48 GB RAM under a time limit of one hour.

Results achieved using Chuffed in our initial experiments are shown in Table 1. The best results achieved in our final experiments are shown in Table 2. The final results show that Chuffed was able to produce the best results for 10 of the instances and prove optimal solutions for 7 instances. Gurobi and Cplex could not provide any additional best results but could also prove optimal solutions for instances 4 instances within the time limit. As we mentioned, the benchmark instances were previously solved in [11] with a metaheuristic approach. This metaheuristic approach produced better results than our initial exact method for most of the instances. However, the current exact approach that uses DFAs provides 7 previously unknown optimal solutions. Therefore, these two approaches complement each other.

	Chuffed	Chuffed DFA
I1	**775***	**775***
I2	**842***	**842***
I3	56550	**961***
I4	NA	**918***
I5	**17880**	**17880**
I6	**842***	**842***
I7	NA	NA
I8	NA	NA
I9	NA	NA
I10	NA	NA
I11	NA	NA
I12	NA	NA

Table 1: Results of our initial experiments with Chuffed using default parameters. The first column shows the best solution costs produced with the direct model, while the second column shows the best results achieved with the DFA model.

	Chuffed	Gurobi	Cplex	LS
I1	**775***	**775***	**775***	806
I2	**842***	**842***	**842***	868
I3	**961***	**961***	**961***	990
I4	**918***	NA	1160	975
I5	**530***	17880	17880	593
I6	**842***	**842***	**842***	887
I7	**1046**	NA	NA	1084
I8	**1237***	NA	NA	1834
I9	**1006**	NA	NA	1735
I10	**973**	NA	NA	1180
I11	NA	NA	NA	**5476**
I12	NA	NA	NA	**5723**

Table 2: The final best results achieved for instances 1–12 using Chuffed, Gurobi and Cplex compared with the best known upper bounds from [11](LS). The best result within each line is formatted in bold face. Results marked with a * denote proven optimal solutions.

7 Conclusion

In this paper we have proposed CP modeling techniques to solve a real life paint shop scheduling problem. Additionally, we analyzed the problem's complexity and have proven that the decision variant is NP-complete.

Our experimental results show that the exact methods investigated in this paper can be used successfully to solve small to medium sized instances which still have a huge search space. The advantage of these methods is that they could provide previously unknown optimal solutions for some instances. Our experiments show that the best results could be achieved by using the DFA based model. Based on this model Chuffed produced better results than Gurobi and Cplex for the majority of the benchmark instances.

In the future we plan to consider the hybridization of the proposed modeling techniques together with existing metaheuristic approaches within the framework of large neighborhood search.

References

1. Chu, G., Stuckey, P.J., Schutt, A., Ehlers, T., Gange, G., Francis, K.: Chuffed, a lazy clause generation solver. https://github.com/chuffed/chuffed (2018)
2. Epping, T., Hochstättler, W., Oertel, P.: Complexity results on a paint shop problem. Discrete Applied Mathematics **136**(2), 217 – 226 (2004)

3. IBM, CPLEX: 12.8.0 cplex user's manual (2017)
4. Karp, R.M.: Reducibility among combinatorial problems. In: Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA. pp. 85–103 (1972)
5. LLC, G.O.: Gurobi optimizer reference manual. http://www.gurobi.com (2018)
6. Nethercote, N., Stuckey, P.J., Becket, R., Brand, S., Duck, G.J., Tack, G.: Minizinc: Towards a standard CP modelling language. In: Principles and Practice of Constraint Programming - CP 2007, 13th International Conference, CP 2007, Providence, RI, USA, September 23-27, 2007, Proceedings. pp. 529–543 (2007)
7. Prandtstetter, M., Raidl, G.R.: An integer linear programming approach and a hybrid variable neighborhood search for the car sequencing problem. European Journal of Operational Research **191**(3), 1004 – 1022 (2008)
8. Solnon, C., Cung, V.D., Nguyen, A., Artigues, C.: The car sequencing problem: Overview of state-of-the-art methods and industrial case-study of the roadef'2005 challenge problem. European Journal of Operational Research **191**(3), 912 – 927 (2008)
9. Spieckermann, S., Gutenschwager, K., Voß, S.: A sequential ordering problem in automotive paint shops. International Journal of Production Research **42**(9), 1865–1878 (2004)
10. Wagner, R.A., Fischer, M.J.: The string-to-string correction problem. J. ACM **21**(1), 168–173 (1974)
11. Winter, F., Demirović, E., Musliu, N., Mrkvicka, C.: Modeling and solving an automotive paint shop scheduling problem. In: Practice and Theory of Automated Timetabling, 12th International Conference, PATAT 2018, Vienna, Austria. pp. 477–480 (2018 (Extended abstract, An extended version of the paper is under submission))