Analysis and Design of Sequencing Rules for Car Sequencing

Uli Golle[a], Nils Boysen[b], Franz Rothlauf[a]

[a]Johannes Gutenberg-Universität Mainz, Lehrstuhl für Wirtschaftsinformatik und BWL, Jakob-Welder-Weg 9, D-55128 Mainz, Germany, {golle,rothlauf}@uni-mainz.de

[b]Friedrich-Schiller-Universität Jena, Lehrstuhl für Operations Management, Carl-Zeiß-Straße 3, D-07743 Jena, Germany, nils.boysen@uni-jena.de

Abstract

This paper presents novel approaches for generating sequencing rules for the car sequencing (CS) problem in cases of two and multiple processing times per station. The CS problem decides on the succession of different car models launched down a mixed-model assembly line. It aims to avoid work overloads at the stations of the line by applying so-called sequencing rules, which restrict the maximum occurrence of labor-intensive options in a subsequence of a certain length. Thus to successfully avoid work overloads, suitable sequencing rules are essential. The paper shows that the only existing rule generation approach leads to sequencing rules which misclassify feasible sequences. We present a novel procedure which overcomes this drawback by generating multiple sequencing rules. Then, it is shown how to apply both procedures in case of multiple processing times per station. For both cases analytical and empirical results are derived to compare classification quality.

Keywords: Mixed-model assembly lines; Car sequencing; Sequencing rules

1 Introduction

Mixed-model assembly lines allow car manufacturers to produce a large variety of different models of a common base product on a single production line. The sequence of models is important since it affects economic parameters. A major cost driver are work overloads of assembly workers, which can occur if several labor-intensive models are scheduled consecutively on the line. Work overloads need to be compensated by cost-intensive strategies, e.g., application of utility workers or line stoppage. Therefore, car manufacturer are interested in finding model sequences with minimum work overload. Common approaches are car sequencing (CS) and mixed-model sequencing (MMS).

CS (Parrello et al., 1986; Solnon et al., 2008) introduces sequencing rules $H_o : N_o$ for each labor-intensive option o, which restrict the occurrence of this option to at most H_o in any subsequence of N_o successive models. The goal is to find a sequence, which does not violate any of the given sequencing rules or – if such a sequence is not existent – minimizes rule violations. The alternative MMS approach (Wester and Kilbridge, 1964) evaluates a sequence by explicitly considering operation times, worker movements, station borders, and other operational characteristics of a line. This way, work overloads can exactly be quantified and, thus, be minimized. A sequence is called feasible if no work overload occurs during the execution of the sequence and unfeasible otherwise.

Both, MMS and CS, try to minimize work overload. MMS directly determines the work overload resulting into high effort for data collection, data preciseness, and computation time. In contrast, CS, which is based on a surrogate objective for work overload by applying sequencing rules, is simple to apply and data requirements are low. However, if the applied sequencing rules are not suitable, CS can be less accurate than MMS and wrongly classify sequences to be either feasible or unfeasible. Therefore, the generation of adequate rules for CS is important. The CS research mainly focuses on the development of efficient solution procedures (Solnon et al., 2008; Boysen et al., 2009) for finding optimal sequences. Only little work deals with the definition of sequencing rules. Drexl and Kimms (2001) provide a rather intuitive example:

"Assume that 60% of the cars manufactured on the line need the option 'sun roof'. Moreover, assume that five cars (copies) pass the station where the sun roofs are installed during the time for the installation of a single copy. Then, three operators (installation teams) are necessary for the installation of sun roofs. Hence, the capacity constraint of the final assembly line for the option 'sun roof' is three out of five in a sequence, or 3:5 for short." (Drexl and Kimms, 2001)

Bolat and Yano (1992) presented the only analytical approach on how to derive sequencing rules. It is limited to cases where two different processing times occur at each station. In this paper, we study how well sequencing rules obtained by the Bolat and Yano (BY) approach correctly classify sequences to be either feasible or unfeasible. Since our analysis shows that a large percentage of feasible sequences are classified as unfeasible, we develop a multiple sequencing rule approach (MSR) that generates CS rules that correctly classify sequences as feasible and unfeasible respectively. Finally, we consider multiple processing times per station, discuss how the BY and MSR approach can be used for such scenarios, and study how well sequences are correctly classified.

Section 2 introduces fundamental assumptions and defines measures for the classification quality of sequencing rules. Then, we study classification quality of the BY approach (Sect. 3) as well as the novel MSR approach (Sect. 4). Section 5 presents different possibilities of how to develop sequencing rules for stations with multiple processing times and studies how well the BY and MSR approach correctly classifies sequences to be either feasible or unfeasible. The paper ends with concluding remarks.

2 Assumptions and Classification Quality

We introduce fundamental assumptions of the mixed-model assembly line and define measures on the classification quality of sequencing rules.

- The model-mix, i.e., the demand for models throughout the planning horizon, is known with certainty. Thus, rush orders or breakdowns do not occur, so that only static sequencing problems are considered.

- Workpieces are moved with constant velocity through the station which are successively arranged along the line. W.l.o.g. lines flow from left to right. We assume no buffers between stations.

- Fixed rate launching is applied, so that consecutive units are placed on the line at the same intervals equal to cycle time c.

- We assume closed stations, so working across the stations' boundaries is not possible.

- Assembly workers return with infinite velocity to the next workpiece. This is an adequate simplification whenever the conveyor speed is much slower than the walking speed of workers. Otherwise, cycle time can ex ante be reduced by a constant return time. Furthermore, processing starts instantaneously once worker and workpiece meet inside a station.

- We assume a deterministic problem since processing times p_{mk} per model $m \in M$ and station $k \in K$ are known with certainty. Moreover, all possible models can be processed inside a station when starting work at the left-hand border: $p_{mk} \leq l_k \;\forall\; m \in M;\; k \in K$, with l_k being the length of station k. Otherwise, sequence-independent work overload exists and a station inevitably is overloaded with any occurrence of the respective model.

  Cycle times range in between the minimum and maximum processing times at each station: $\min_{m \in M}\{p_{mk}\} \leq c \leq \max_{m \in M}\{p_{mk}\} \,\forall\, k \in K$.

- Work overload occurs whenever an assembly operator is not able to finish his/her present workpiece (with normal processing velocity) before reaching the right-hand station border. Then, in the real-world some kind of compensation, e.g., line stoppage or applying cross-trained utility workers, is required.

These assumptions define the status of an assembly system for a given model sequence over the complete planning horizon. Therefore, for any given sequence MMS can accurately quantify the resulting work overload and correctly classify a sequence to be either feasible or unfeasible.

Throughout the paper, we restrict ourselves to feasibility problems and label feasible sequences as *MMS-feasible*. If a sequence does not violate any sequencing rule of a CS approach, it is denoted as *CS-feasible*. Both approaches, CS and MMS, are equivalent if CS-feasibility induces MMS-feasibility (CS-feasible$\rightarrow$MMS-feasible) and vice versa (MMS-feasible$\rightarrow$CS-feasible). For CS, such a one-to-one mapping is desirable since it correctly classifies all sequences to be either feasible or unfeasible. However, there are two possible types of misclassifications:

- CS-feasible$\nrightarrow$MMS-feasible. The sequencing rules of CS classify sequences to be feasible although they are unfeasible. Sequencing rules are not strict enough and do not identify all sequences that cause work overload. This case causes major problems in the real-world since additional costs for dealing with unforeseen work overloads occur.

- CS-feasible$\nleftarrow$MMS-feasible. CS classifies sequences to be unfeasible although they are feasible. There are sequences that cause no work overload but violate at least one sequencing rule. In this case, no unforeseen work overloads occur but CS excludes feasible sequences from consideration. Obviously, this impedes the search for any solution procedure and potentially excludes optimal solutions if the feasibility version of CS is coupled with an additional objective function (Drexl and Kimms, 2001). As worst case scenario, CS wrongly classifies all feasible solutions, so no feasible solution can be found.

The lower the percentage of misclassifications the better the classification quality of the applied sequencing rules and, thus, the better the rule generation approach that lead to this rules.

3 Classification Quality of the Bolat and Yano approach

To our best knowledge, the only analytical approach on how to derive sequencing rules was proposed by Bolat and Yano (1992). They assumed one option per station with two processing times. All models containing the option (denoted as *option models*) require processing time p^+ and all basic models without the option require processing time p^-, with $p^- < c < p^+ \leq l$, where c is the cycle time and l the station length. Bolat and Yano proposed to generate sequencing rules $H : N$ that restrict the number of option models to at most H within a subsequence of N consecutive models, where

$$H = \left\lfloor \frac{l - c}{p^+ - c} \right\rfloor \quad \text{and} \tag{1}$$

$$N = H + \left\lceil \frac{H \cdot (p^+ - c)}{c - p^-} \right\rceil . \tag{2}$$

H are the maximum possible number of consecutive option models without outreaching the right-hand station border. N adds to H the number of basic models required to reset the workers starting point to the left-hand station border after H successive option models.

For an example with $l = 15$, $c = 10$, $p^+ = 12$ and $p^- = 7$, the BY approach returns the sequencing rule $2 : 4$. Therefore, at most $H = 2$ option models can be produced consecutively, before $N - H = 2$ basic models are required to reset the subsequence to the left border again. The following properties of the BY approach hold:

Proposition: For the BY approach, CS-feasible$\rightarrow$MMS-feasible.
Proof: A sequence is CS-feasible if the $H : N$-rule is not violated (i) inside any subsequence of N cars and (ii) when concatenating subsequences to form a longer sequence. In the worst case, all H option models succeed in a row. In a sequence of N cars, the worker starts processing a subsequence of H consecutive option models at the left-hand border and ends at $H(p^+ - c)$. According to (1), this end lies before station's length l. (2) calculates the sequence length N such that a sequence of $N - H$ basic models resets position from the right most point $H(p^+ - c)$ back to the left-hand border. Therefore, (i)

holds. If only feasible subsequences are concatenated, there is no interplay between the subsequences and (ii) holds. $\square$

Proposition: For the BY approach, CS-feasible$\not\leftarrow$MMS-feasible.

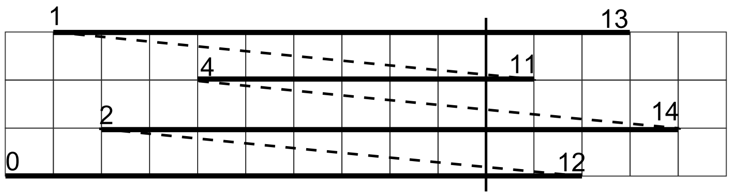


Figure 1: Counterexample for CS-feasible $\leftarrow$ MMS-feasible

Proof: We prove by giving a counterexample to the contradiction of our proposition: CS-feasible$\leftarrow$MMS-feasible. For the above example with $l = 15$, $c = 10$, $p^+ = 12$ and $p^- = 7$, the sequence $\pi =< +, +, -, +]$ is MMS-feasible (compare movement diagram of Figure 1). "+" and "-" represents an option model and basic model, respectively. In the movement diagram, workers accompanying their workpiece are solid horizontal lines; return movements are dashed diagonal lines. The BY approach yields a $2:4$-rule, which is violated by sequence π. Thus, π is MMS-feasible but not CS-feasible and the proposition holds. $\square$

A MMS-feasible sequence evaluated by rules generated with the BY approach is considered as CS-feasible, only if it allows the worker to reset to the left-hand station border after processing N consecutive models. MMS-feasible sequences where this is not the case (see example in Figure 1) violate at least one sequencing rule and are, thus, misclassified. Although CS-feasible$\not\leftarrow$MMS-feasible holds in general, there are two special cases for which CS-feasible$\leftarrow$MMS-feasible holds:

1. $H = 1$ and $l - p^+ < r$, with

$$r = \begin{cases} (p^+ - c) \mod (c - p^-) & \text{if } (p^+ - c) \mod (c - p^-) > 0, \\ c - p^- & \text{if } (p^+ - c) \mod (c - p^-) = 0. \end{cases}$$

   After each option model, at least $N - H$ basic models must be processed before the next option model can follow. Therefore, every feasible sequence allows the worker to reset after at most N successive models.

2. $N - H = 1$.
   The worker resets immediately to the left-hand border by processing one basic model. Since only H option models can be processed consecutively without inducing work overload, every feasible sequence allows the worker to reset after at most $H + 1 = N$ successive models.

For the one and two station case, we study the number of sequences in percent for which CS-feasible$\not\leftarrow$MMS-feasible holds. For each sequence length T, we create $1,000$ random problem instances. The parameters settings c, l, p^+, and p^- of each instance are

cycle time c	$\in [10, 20]$
station length l	$\in (c, 40]$
processing time of option models p^+	$\in (c, l]$
processing time of basic models p^-	$\in [1, c)$

Table 1: Parameter settings

chosen randomly according to Table 1. For one station (two stations) and $T \leq 22$ (≤ 12), we consider all possible sequences by enumeration; for larger T, at least $1,000,000$ random sequences are generated for each instance. If necessary, the sampling size is increased until it contains at least 10 MMS-feasible sequences.

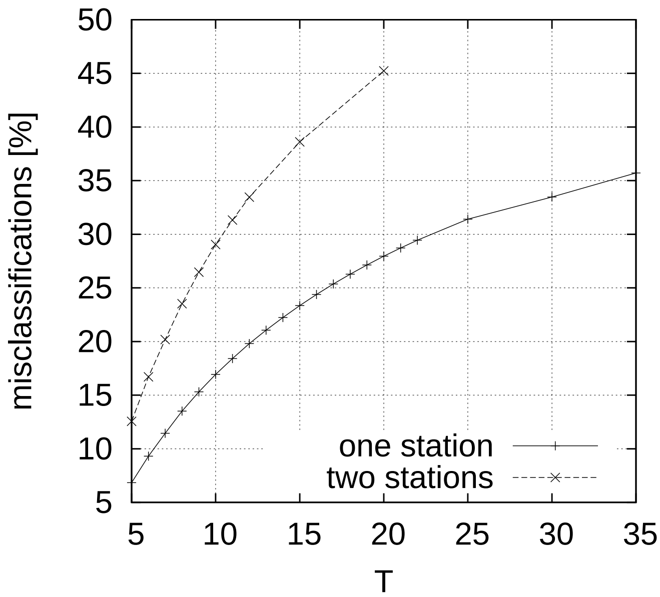


Figure 2: Number of misclassified sequences where CS-feasible$\nLeftarrow$MMS-feasible (in %)

For each sequence, we determine the CS-feasibility using the sequencing rules obtained by the BY approach as well as the MMS-feasibility. Figure 2 shows the number of sequences (in percent) that are MMS-feasible but CS-unfeasible. For example, when considering only one station and a sequence length of $T = 30$, then about 33% of MMS-feasible solutions are classified by the BY approach as unfeasible. The percent of sequences where CS-feasible$\nLeftarrow$MMS-feasible increases with T and with the number of stations.

In summary, all sequences that are CS-feasible are also MMS-feasible. However, a large portion of MMS-feasible solutions are incorrectly classified by the sequencing rules obtained from the BY approach to be CS-unfeasible. Therefore, using this sequencing rules in optimization approaches that search for feasible solutions can be problematic since many feasible solutions are excluded from the search space.

4 MSR – A Novel Rule Generation Approach

We propose a new rule generation approach with multiple sequencing rules (MSR) for stations with one option and two processing times. The new approach generates a number of sequencing rules which correctly classifies sequences to be either feasible or unfeasible.

The MSR approach calculates k_{min}, which is the maximum number of successive option models, as

$$k_{min} = \left\lfloor \frac{l-c}{p^+ - c} \right\rfloor . \tag{3}$$

k_{min} equals H from the BY approach (1). Consider a sequence of length T with k option models and $T-k$ basic models. After processing the whole sequence a worker can be at most $l-c$ time units away from the left-hand station border without inducing work overload. Thus, $l-c \geq k(p^+ - c) - (T-k)(c-p^-)$ holds. Rewriting this inequation leads to the maximum number of option models k_{max} that may occur in a sequence of length T:

$$k_{max} = \left\lfloor \frac{T(c-p^-) + (l-c)}{p^+ - p^-} \right\rfloor . \tag{4}$$

MSR generates $k_{max} - k_{min} + 1$ different sequencing rules $H^{k-k_{min}+1} : N^{k-k_{min}+1}$ with

$$H^{k-k_{min}+1} = k \text{ and } \; N^{k-k_{min}+1} = k + m \tag{5}$$

$\forall k \in [k_{min}, k_{max}]$ and

$$m = \left\lceil \frac{k(p^+ - c) - (l - p^+)}{c - p^-} \right\rceil . \tag{6}$$

Equation (6) is a modification of (2) of the BY approach and calculates the minimum number of basic models m that are required after k option models before another option model can be processed. For an example with $l = 17$, $c = 10$, $p^+ = 13$, and $p^- = 5$, the BY approach returns a $2 : 4$ rule. For a sequence of $T = 4$, MSR generates two rules $H^1 : N^1 = 2 : 3$ and $H^2 : N^2 = 3 : 4$. A sequence that satisfies all rules generated by MSR is CS-feasible. For the situation that three option models (+) and one basic model (-) need to be assembled, Figure 3 shows the four different sequences ($\pi_1, \ldots, \pi_4$) which are correctly classified by the MSR approach to be either feasible or unfeasible. The BY approach would wrongly classify all four sequences as CS-unfeasible.

Proposition: For the MSR approach, CS-feasible $\rightarrow$ MMS-feasible.
Proof: We prove by contraposition and show CS-unfeasible$\leftarrow$MMS-unfeasible. A MMS-unfeasible sequence contains at least one option model at position j with starting time $s_j > l - p^+$. Processing the option model at position j leads to work overload. W.l.o.g. j is the first position in the sequence, where work overload occurs. Position $i = max(x | x < j \wedge s_x = 0)$ is the last time where the worker resets before j. i always exists, since the sequence starts with $s_1 = 0$. We neglect all models at positions less than i since they have no influence on s_j. Furthermore, we only consider the subsequence $\pi_{[i,j)}$ excluding the option model at position j. Since no reset or work overload occurs in $\pi_{[i,j)}$, the starting time s_j is the sum of the displacements of all models in $\pi_{[i,j)}$. With $\overline{k}$ option models

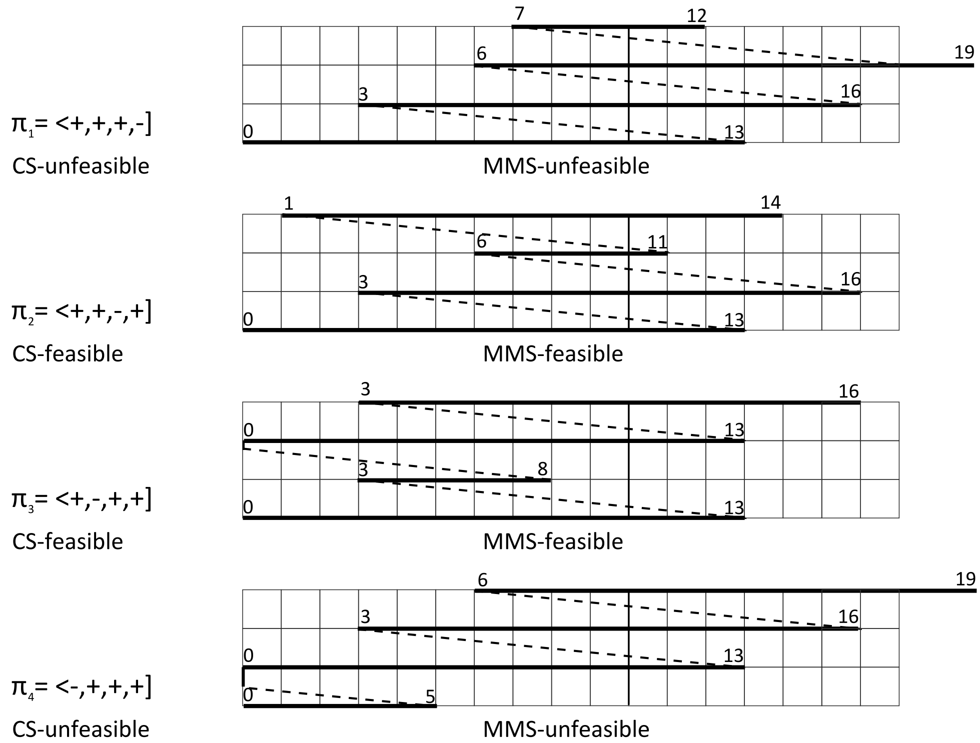


Figure 3: Example for the MSR approach with sequences that assemble three option models (+) and one basic model (-)

and $\overline{m}$ basic models, $s_j = \overline{k}(p^+ - c) + \overline{m}(p^- - c) = \overline{k}(p^+ - c) - \overline{m}(c - p^-) > l - p^+$. Since $k_{min} \leq \overline{k} \leq k_{max}$, a sequencing rule exists for $\overline{k}$ with $N^{\overline{k}-k_{min}+1} = \overline{k} + m$. According to (6), this sequencing rule demands $l - p^+ \geq \overline{k}(p^+ - c) - m(c - p^-)$. Therefore, $\overline{k}(p^+ - c) - \overline{m}(c - p^-) > \overline{k}(p^+ - c) - m(c - p^-)$ which leads to $m > \overline{m}$. The subsequence $\pi_{[i,j)}$ has a length lower than $N^{\overline{k}-k_{min}+1}$ but already contains $\overline{k}$ option models. Having an option model at position j does not only induce work overload but also leads to a violation of the respective sequencing rule, since the subsequence $\pi_{[i,j]} = \pi_{[i,j)} + \{j\}$ has length $\leq N^{\overline{k}-k_{min}+1}$ but contains $\overline{k} + 1$ option models. This completes the proof that CS-unfeasible←MMS-unfeasible and therefore CS-feasible→MMS-feasible. □

Proposition: For the MSR approach, CS-feasible ← MMS-feasible.
Proof: We use a contraposition proof and show CS-unfeasible → MMS-unfeasible. A CS-unfeasible sequence violates at least one sequencing rule. Thus, a subsequence π of length $t = k + m$ with $k \in [k_{min}, k_{max}]$ contains at least $k + 1$ option models and $m - 1$ basic models. W.l.o.g. the violation occurs at the last position in π. Therefore, in the first $t - 1$ positions of π, there are k option models and $m - 1$ basic models. The first position i in π has starting time s_i. Assuming that no work overload occurs in the first $t - 1$ positions, the starting time s_j of the last position j in π must be $\geq s_i + k(p^+ - c) - (m - 1)(c - p^-)$. For $s_i \geq 0$ and $k(p^+ - c) - (m - 1)(c - p^-) > l - p^+$ (compare (6)), we get $s_i + k(p^+ - c) - (m - 1)(c - p^-) > l - p^+$ and hence $s_j > l - p^+$. Therefore, the option model at position j induces work overload. This completes the proof that CS-unfeasible→MMS-unfeasible and therefore CS-feasible←MMS-feasible. □

MSR has the disadvantage that the number $k_{max}-k_{min}+1$ of sequencing rules is relatively large and some of the rules are redundant in the sense that they are covered by other rules. We introduce the concept of strictness.

Definition: A sequencing rule $H : N$ is *stricter* than another sequencing rule $P : Q$, if all possible permutations of option and basic models for a given model mix which are feasible under $H : N$ are also feasible under $P : Q$ but at least one permutation exists which is unfeasible under $H : N$, but feasible under $P : Q$.

A test for strictness compares the maximum number of allowed option models within a subsequence of length N and Q, respectively. In a sequence of length T, the maximal number of options models allowed by sequencing rule $H : N$ is $H\left\lfloor\frac{T}{N}\right\rfloor+\min(T \mod N; H)$ (Fliedner and Boysen, 2008). Thus, $H : N$ is stricter than $P : Q$, if

$$H\left\lfloor\frac{Q}{N}\right\rfloor+\min(Q \mod N; H) \leq P, \text{ and} \tag{7}$$

$$H \neq P \vee N \neq Q \tag{8}$$

(7) ensures that the number of option occurrences under rule $H : N$ never exceeds P for any feasible subsequence of length Q. Thus, any permutation which does not violate the $H : N$ rule is also feasible for $P : Q$. Inequalities (8) ensure that both rules are not identical. For an example with $l = 20$, $c = 10$, $p^+ = 20$, $p^- = 0$, and $T = 10$, MSR generates five rules $1 : 2$, $2 : 4$, $3 : 6$, $4 : 8$, and $5 : 10$. According to strictness, the latter four rules are redundant and the rule set can be reduced to a single $1 : 2$ rule.

Note that redundant rules can not be identified by interpreting rules as fractions and arguing that rule $H : N$ is stricter than rule $P : Q$ if $H/N < P/Q$. Such an argumentation may lead to wrong results. For example, in a feasible sequence of length $T = 4$, we assume two option models and two basic models leading to six possible permutations. A $1 : 2$ classifies three of them as feasible. The conjecture that a $2 : 5$ rule is more strict ($2/5 < 1/2$) is wrong since $2 : 5$ classifies all six permutations as feasible.

For different T, we study the number of redundant sequencing rules for problems with one station. We create 1,000 random problem instances with parameters from Table 1. Figure 4 shows the average number of sequencing rules generated by MSR over the sequence length T with and without eliminating redundant sequencing rules. The MSR approach using strictness for the elimination of redundant sequencing rules is denoted as MSRstrict. When eliminating redundant rules, the number of sequencing rules can approximately cut in half. For $T = 50$, MSR generates on average 22.5 sequencing rules per station; MSRstrict, which eliminates redundant rules, only produces on average 12.6 rules.

In summary, the new MSR approach for one option per station with two processing times generates multiple sequencing rules. It correctly classifies all sequences either as feasible or unfeasible. The definition of strictness is able to halve the number of necessary rules for MSR by eliminating redundant rules.

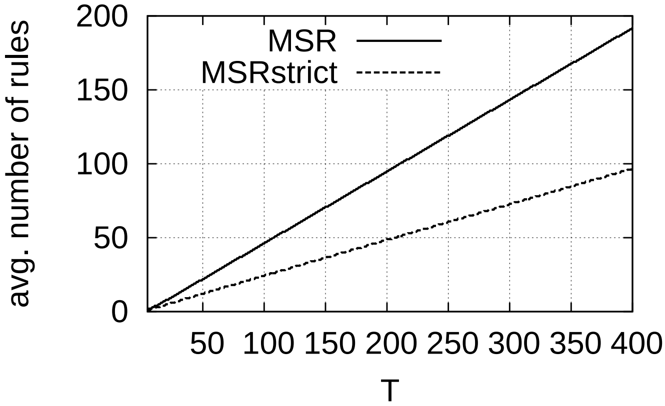


Figure 4: Average number of sequencing rules over sequence length T for MSR and MSRstrict, which removes redundant rules.

5 Multiple processing times

Although the literature focused on rule generation approaches for stations with only two possible options (Bolat and Yano, 1992), in real life, options with multiple processing times per station are common. For example, in automobile assembly, sunroofs can come in three options with no sunroof, manual, or electric; transmission may be manual or automatic in 4 to 6 speeds; onboard electronic devices, such as stereo systems can come in various configurations. Furthermore, multiple options (parts or modules) might need to be installed per station.

We assume more than two models $m \in M$ with diverging processing times p_m. M is split into two subsets $M^+ = \{m \in M | p_m > c\}$ and $M^- = \{m \in M | p_m < c\}$ containing all models with processing time greater and smaller than cycle time c, respectively. All models with $p_m = c$ are excluded from consideration, as their production does not modify the ending position within the station. Thus, they can never produce additional work overload and can be scheduled at facultative sequence positions.

We reduce the multiple options case to the two option case by introducing one *virtual option* per station. All models $m \in M^+$ require the virtual option with processing time p^{v+}; all $m \in M^-$ with processing time p^{v-} do not require the virtual option. For the virtual processing times p^{v+} and p^{v-}, there are different possibilities: using the maximal (MAX), average (AVG), or minimal (MIN) processing times of each set:

$$\text{MAX:} \quad p^{v+} = \max_{m \in M^+} \{p_m\}, \qquad p^{v-} = \max_{m \in M^-} \{p_m\} \tag{9}$$

$$\text{AVG:} \quad p^{v+} = \frac{\sum_{m \in M^+} \{p_m\}}{|M^+|}, \qquad p^{v-} = \frac{\sum_{m \in M^-} \{p_m\}}{|M^-|} \tag{10}$$

$$\text{MIN:} \quad p^{v+} = \min_{m \in M^+} \{p_m\}, \qquad p^{v-} = \min_{m \in M^-} \{p_m\} \tag{11}$$

Having two options (either the virtual option or not) and corresponding processing times, sequencing rules can be derived using either the BY or MSR approach. We give an example with four models to be processed at a station with length $l = 10$ and cycle time

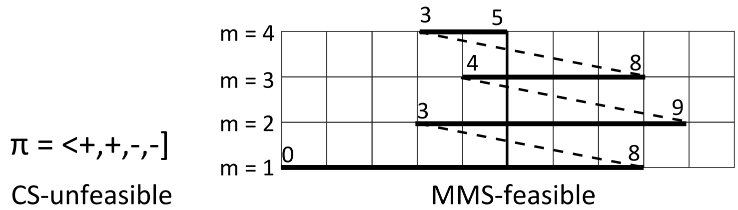


Figure 5: Counterexample for CS-feasible$\not\leftarrow$MMS-feasible

$c = 5$. $M^+ = \{1, 2\}$ contains two models with processing times $p_1 = 8$ and $p_2 = 6$. Models 3 and 4 with processing times of $p_3 = 4$ and $p_4 = 2$ belong to set M^-. Models 1 and 2 require the virtual option; models 3 and 4 are declared as basic models. Using the maximal processing times (MAX), we get $p^{v+} = 8$ and $p^{v-} = 4$. For a sequence length $T = 4$, the BY approach results into a $1 : 4$ rule and the MSR approach to rules $1 : 2$ and $2 : 6$.

Proposition: For the virtual option approach with MAX aggregation, CS-feasible$\rightarrow$MMS-feasible.

Proof: CS-feasible$\rightarrow$MMS-feasible holds for both the BY and MSR approach with one option and processing times $p^+ = p^{v+} = \max_{m \in M^+}\{p_m\}$ and $p^- = p^{v-} = \max_{m \in M^-}\{p_m\}$. Having additional options $m \in M^+$ with processing $p_m < p^{v+}$ and $m \in M^-$ with processing $p_m < p^{v-}$ can only move ending times to the left and never lead to additional work overload. □

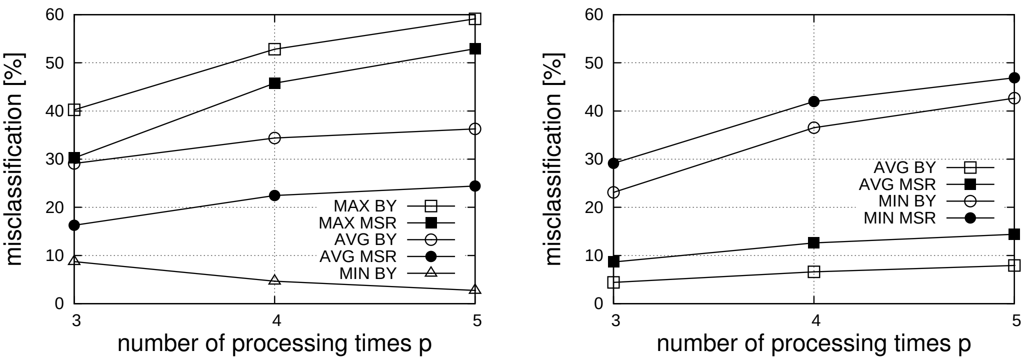
In contrast, for the AVG and MIN aggregation, CS-feasible$\rightarrow$MMS-feasible does not hold, which could simply be proven by a counterexample.

Proposition: For the virtual option approach with MAX aggregation, CS-feasible$\not\leftarrow$MMS-feasible holds.

Proof: We prove by a counterexample to the contradiction of our proposition: CS-feasible$\leftarrow$MMS-feasible. Consider the example with $l = 10$, $c = 5$, $p_1 = 8$, $p_2 = 6$, $p_3 = 4$, $p_4 = 2$, $M^+ = \{1, 2\}$, and $M^- = \{3, 4\}$. Any possible sequence of four different models is MMS-feasible. The BY approach returns a $1 : 4$ rule. Since model 1 and 2 require the virtual option, no CS-feasible sequence exists. The MSR approach leads to rules $1 : 2$ and $2 : 6$. Therefore, all sequences where model 1 directly follows model 2 and vice versa, are CS-unfeasible. Figure 5 shows a sequence that is MMS-feasible but CS-unfeasible for both the BY and MSR approach. □

For the BY approach, both AVG and MIN aggregation lead to CS-feasible$\not\leftarrow$MMS-feasible. For MSR and AVG aggregation, CS-feasible$\not\leftarrow$MMS-feasible. Combining MSR with MIN aggregation results into CS-feasible$\leftarrow$MMS-feasible. Since CS-feasible$\leftarrow$MMS-feasible holds for the case with only two processing times, it also holds for the virtual option case since all options $m \in M^+$ and $m \in M^-$ have processing times larger than p^{v+} and p^{v-}, respectively.

For the three aggregation possibilities, we study the classification quality as the num-



(a) CS-feasible$\nleftarrow$MMS-feasible (b) CS-feasible$\nrightarrow$MMS-feasible

Figure 6: Number of misclassified sequences (in percent) for $T = 10$

ber of sequences in percent for which CS-feasible$\nleftarrow$MMS-feasible and CS-feasible$\nrightarrow$MMS-feasible holds, respectively. Experimental design follows Section 3. For either three, four, or five different processing times (and thus models) and various station lengths $T \in \{10, 15, 20\}$, we generate $1,000$ random problem instances according to the parameters given in Table 1. For $T = 10$, we consider all possible sequences; for larger T, at least $1,000,000$ random sequences for each instance are sampled. Sampling size is iteratively increased by a factor of $1,000,000$ until it contains at least 10 MMS-feasible sequences.

For each sequence, we determine MMS-feasibility. Furthermore, for the virtual option approach with MAX, AVG, and MIN aggregation, we determine CS-feasibility using either the BY or MSR approach. For $T = 10$, Figure 6(a) shows the number of sequences (in percent) that are MMS-feasible but not CS-feasible (CS-feasible$\nleftarrow$MMS-feasible) in relation to the number of processing times p. Table 2 lists complete results for different T and p.

Using MAX aggregation, the number of feasible solutions that are wrongly classified as CS-unfeasible is highest. For MAX and AVG aggregation, the percentage of wrongly classified sequences increases the more processing times p are considered; for MIN it decreases. In comparison to BY, MSR misclassifies a lower percentage of sequences. As mentioned before, combining MSR and MIN leads to CS-feasible$\leftarrow$MMS-feasible.

Analogously, Figure 6(b) presents the number of sequences (in percent) that are CS-feasible but not MMS-feasible (CS-feasible$\nrightarrow$MMS-feasible) over p for $T = 10$. In comparison to AVG aggregation, MIN aggregation leads to a higher percentage of wrongly classified sequences. The BY approach correctly classifies a higher number of sequences than the MSR approach.

A tradeoff can be observed between CS-feasible$\nleftarrow$MMS-feasible and CS-feasible$\nrightarrow$MMS-feasible. Aggregations that misclassify a large number of MMS-feasible sequences as CS-unfeasible, misclassify a lower number of MMS-unfeasible sequences as CS-feasible and vice versa. This tradeoff applies to both the BY and MSR approach.

Wrongly classifying an MMS-unfeasible sequence as CS-feasible is problematic when searching for feasible sequences. Executing such an MMS-unfeasible sequence would lead to unforeseen overhead and cause large additional cost. Therefore, for multiple processing times, we only find MAX aggregation useful since it guaranties that every CS-feasible

			BY approach			MSR approach		
p	agg.	classification error	T=10	T=15	T=20	T=10	T=15	T=20
3	MAX	CS-feasible $\nleftarrow$ MMS-feasible	40,23 %	48,57 %	53,98 %	30,24 %	36,86 %	41,43 %
	AVG	CS-feasible $\nleftarrow$ MMS-feasible	29,10 %	36,35 %	41,22 %	16,26 %	21,21 %	24,96 %
		CS-feasible $\nrightarrow$ MMS-feasible	4,42 %	6,17 %	7,50 %	8,64 %	12,11 %	14,82 %
	MIN	CS-feasible $\nleftarrow$ MMS-feasible	8,71 %	12,28 %	15,29 %	0,00 %	0,00 %	0,00 %
		CS-feasible $\nrightarrow$ MMS-feasible	23,11 %	27,91 %	31,13 %	29,14 %	35,29 %	39,30 %
4	MAX	CS-feasible $\nleftarrow$ MMS-feasible	52,83 %	62,21 %	67,41 %	45,76 %	54,57 %	59,87 %
	AVG	CS-feasible $\nleftarrow$ MMS-feasible	34,38 %	43,11 %	48,96 %	22,44 %	29,21 %	33,95 %
		CS-feasible $\nrightarrow$ MMS-feasible	6,61 %	9,59 %	11,09 %	12,60 %	17,73 %	21,73 %
	MIN	CS-feasible $\nleftarrow$ MMS-feasible	4,65 %	6,95 %	8,64 %	0,00 %	0,00 %	0,00 %
		CS-feasible $\nrightarrow$ MMS-feasible	36,52 %	44,13 %	49,13 %	41,97 %	50,36 %	55,57 %
5	MAX	CS-feasible $\nleftarrow$ MMS-feasible	59,12 %	68,70 %	73,99 %	52,93 %	62,48 %	68,04 %
	AVG	CS-feasible $\nleftarrow$ MMS-feasible	36,27 %	45,47 %	51,60 %	24,41 %	31,89 %	37,16 %
		CS-feasible $\nrightarrow$ MMS-feasible	7,93 %	11,14 %	13,55 %	14,41 %	20,11 %	24,67 %
	MIN	CS-feasible $\nleftarrow$ MMS-feasible	2,73 %	3,83 %	4,89 %	0,00 %	0,00 %	0,00 %
		CS-feasible $\nrightarrow$ MMS-feasible	42,67 %	51,61 %	57,47 %	46,89 %	56,24 %	62,04 %

Table 2: Number of sequences where CS-feasible$\nleftarrow$MMS-feasible and CS-feasible$\nrightarrow$MMS-feasible, respectively

sequence is also MMS-feasible. Combining MAX aggregation with MSR finds a larger number of MMS-feasible solutions in comparison to the BY approach. Again this comes for the price of an enlarged rule set.

6 Conclusion

This paper investigates the classification quality resulting from existing and novel procedures for generating car sequencing rules, where quality is measured by the fraction of sequences for which a generated rule set properly predicts whether or not work overload occurs. Analytical and empirical results show a superior classification quality of our novel MSR approach for both the two and multiple processing times case. However, this comes for the price of additional sequencing rules to be introduced per instance. Thus, to benefit from more accurate rules solution procedures are required which are able to handle large rule sets. Furthermore, future research should investigate the optimization version of minimizing work overload. Here, special rule generation procedures are required which additionally derive option-specific penalty values weighting rule violations according to the resulting amount of work overload.

References

Bolat, A., Yano, C., 1992. Scheduling algorithms to minimize utility work at a single station on a paced assembly line. Production Planning & Control 3 (4), 393–405.

Boysen, N., Fliedner, M., Scholl, A., 2009. Sequencing mixed-model assembly lines: Survey, classification and model critique. European Journal of Operational Research 192 (2), 349–373.

Drexl, A., Kimms, A., 2001. Sequencing jit mixed-model assembly lines under station-load and part-usage constraints. Management Science 47 (3), 480–491.

Fliedner, M., Boysen, N., 2008. Solving the car sequencing problem via branch & bound. European Journal of Operational Research 191 (3), 1023–1042.

Parrello, B., Kabat, W., Wos, L., 1986. Job-shop scheduling using automated reasoning: A case study of the car-sequencing problem. Journal of Automated Reasoning 2, 1–42.

Solnon, C., Cung, V., Nguyen, A., Artigues, C., 2008. 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.

Wester, L., Kilbridge, M. D., 1964. The assembly line model-mix sequencing problem. In: Proceedings of the Third International Conference on Operations Research. Oslo, Paris, 247–260.