

Briskorn, Dirk; Stephan, Konrad; Boysen, Nils

**Article — Published Version**

## Minimizing the makespan on a single machine subject to modular setups

Journal of Scheduling

**Provided in Cooperation with:**

Springer Nature

*Suggested Citation:* Briskorn, Dirk; Stephan, Konrad; Boysen, Nils (2021) : Minimizing the makespan on a single machine subject to modular setups, Journal of Scheduling, ISSN 1099-1425, Springer US, New York, NY, Vol. 25, Iss. 1, pp. 125-137, <https://doi.org/10.1007/s10951-021-00704-8>

This Version is available at:

<https://hdl.handle.net/10419/286997>

**Standard-Nutzungsbedingungen:**

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

**Terms of use:**

*Documents in EconStor may be saved and copied for your personal and scholarly purposes.*

*You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.*

*If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.*



<https://creativecommons.org/licenses/by/4.0/>



# Minimizing the makespan on a single machine subject to modular setups

Dirk Briskorn<sup>1</sup> · Konrad Stephan<sup>2</sup> · Nils Boysen<sup>2</sup>

Accepted: 10 August 2021 / Published online: 14 September 2021  
© The Author(s) 2021

## Abstract

Single machine scheduling with sequence-dependent setup times is one of the classical problems of production planning with widespread applications in many industries. Solving this problem under the min-makespan objective is well known to be strongly NP-hard. We consider a special case of the problem arising from products having a modular design. This means that product characteristics, (mass-)customizable by customers, are realized by separate components that can freely be combined. If consecutive products differ by a component, then a setup is necessary. This results in a specially structured setup matrix that depends on the similarities of product characteristics. We differentiate alternative problem cases where, for instance, the setup operations for multiple components either have to be executed sequentially or are allowed to be conducted in parallel. We analyze the computational complexity of various problem settings. Our findings reveal some special cases that are solvable in polynomial time, whereas most problem settings are shown to remain strongly NP-hard.

**Keywords** Scheduling · Modular products · Sequence-dependent setups · Complexity

## 1 Introduction

We consider a production scheduling problem where jobs are to be sequenced on a single machine subject to sequence-dependent setup times under the min-makespan objective. This problem, which according to the famous 3-field notation introduced by Graham et al. (1979) is denoted by  $[1|s_{ij}|C_{\max}]$ , is closely related to the traveling salesman problem (TSP). The processing times of jobs merely constitute a constant portion of the makespan, and minimizing the makespan is thus equivalent to minimizing the total setup time. Processing jobs and setup times in the scheduling

problem then directly correspond to visiting customers and distances in the TSP. A wide range of results for the TSP with special structures of distance matrices is documented by the survey papers of Boysen and Stephan (2016), Burkard et al. (1998), and Oda and Ota (2001) as well as Stephan and Boysen (2017).

While the scheduling problem with sequence-dependent setup times is strongly NP-hard in general, this paper considers special structures of setup times arising in the context of modular product architectures. We address the question whether under this special structure the problem is still hard to solve or can be solved in polynomial time.

The relevance of specially structured setup matrices in the context of modular products came to our mind when visiting a production plant of Germany's market leader for sectional garage doors (see Fig. 1a). The sections of garage doors, (mass-)customizable by customers selecting the interior and exterior surface coating as well as the kind of polyurethane (PU) foam insulation, are produced in the continuous production process schematically depicted in Fig. 1b. Since interior and exterior steel coatings are unrolled from their coils and continuously pulled along the three main production stages, the whole process can be modeled by a single machine. Sequence-dependent setups occur whenever two subsequent jobs, which correspond to customer orders for

---

✉ Dirk Briskorn  
briskorn@uni-wuppertal.de  
http://www.prodlog.uni-wuppertal.de

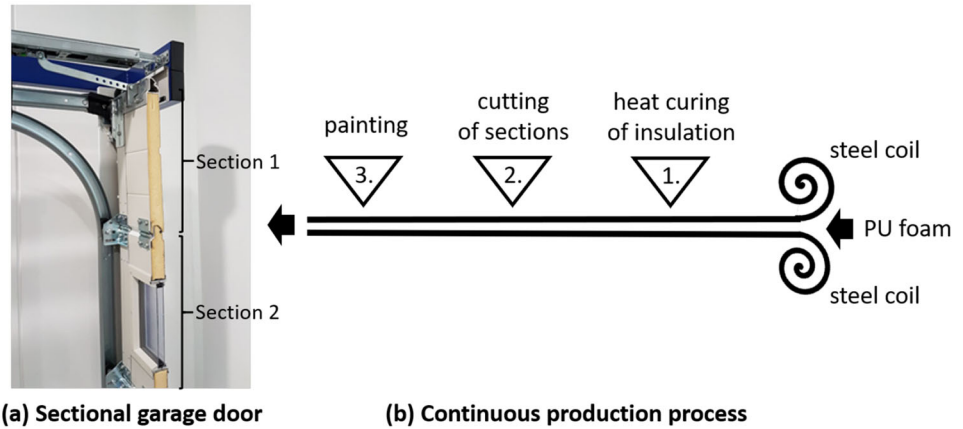
Konrad Stephan  
konrad.stephan@uni-jena.de

Nils Boysen  
nils.boysen@uni-jena.de  
http://www.om.uni-jena.de

<sup>1</sup> Bergische Universität Wuppertal, Professur für BWL, insbesondere Produktion und Logistik, Gaußstr. 20, 42119 Wuppertal, Germany

<sup>2</sup> Friedrich-Schiller-Universität Jena, Lehrstuhl für Operations Management, Carl-Zeiß-Straße 3, 07743 Jena, Germany

**Fig. 1** A modular product and its production process



garage doors, vary in their individual specification according to insulation, interior, and/or exterior coating, so that one or both of the steel coils and/or the cartridge of the PU foam is to be changed. Vice versa, our three resources (i.e., insulation, interior, and exterior coating) remain unchanged if subsequent jobs share the specific product characteristic. Thus, we have a specially structured setup matrix that depends on the similarity of product characteristics of subsequent jobs.

Modular product architectures, providing “a one-to-one mapping from functional elements in the function structure to the physical components of the product” with “de-coupled interfaces between components” (Ulrich 1995) are widely aspired “to provide customized products or services through flexible processes in high volumes and at reasonably low costs” (Da Silveira et al. 2001) in mass-customization environments. Typically, there is also a one-to-one mapping not only from product characteristic to component, but also to a resource required for the assembly of the respective component. Whenever the resource required depends on the product and needs to be exchanged if products differ in the resource, modular products share the peculiarities of our garage door manufacturer, and we have specially structured setup times that depend on the similarity of product characteristics.

For these modular setups, we want to investigate whether the specially structured setup matrices allow solving problem  $[1|s_{ij}|C_{\max}]$  in polynomial time. In doing so, we differentiate problem variants according to three features:

- Setup operations of resources to be altered between two successive jobs can be executed *in parallel*, so that the maximum time over all current resource setups is to be considered, or *sequentially*, so that the sum of single resource setups constitutes the setup time.
- While we always consider operations *equipping* the machine with a resource type, *removal operations* caused by removing the resource type required by the previous job may be relevant or not.

- Setup times per resource can all be *equal*, e.g., changing any type of exterior steel coil, interior steel coil, and PU cartridge always takes the same amount of time, *characteristic-dependent*, e.g., changing any type of exterior steel coil takes identical setup time, which may, however, differ from the constant setup time for changing any type of PU cartridges, or *value-dependent*, e.g., each specific type of steel coil and PU cartridge may require a different setup time.

We combine all manifestations of these features and, hence, have twelve problem variants in total. We determine the computational complexity of all but one of these variants.

There are several other papers that provide polynomial time algorithms for special single machine scheduling problems with setups when minimizing makespan  $C_{\max}$ . They arise in case of family setups where products are subdivided into families and only changes between families cause setup operations, e.g., see Bai et al. (2012), Huang et al. (2011), Lee and Wu (2010), Wang et al. (2008, 2012), Wu and Lee (2008), Xu et al. (2014), Yang (2011), and Yang and Yang (2010) and if setups are applied to model learning effects and their duration decreases with increasing cumulated production time, e.g., see Cheng et al. (2010), Huang et al. (2013), Koulamas and Kyparisis (2008), Lai et al. (2011), Soroush (2012) and Yin et al. (2010, 2012). However (to the best of the authors’ knowledge), none of these existing approaches considers our case of modular setups where the similarity of product characteristics influences setup times. Our own finding (after a thorough literature search) is also supported by the manifold and up-to-date survey papers on scheduling with setups provided in Allahverdi (2015), Allahverdi et al. (1999), Allahverdi et al. (2008) and Stefansdottir et al. (2017) where modular setups are not mentioned.

The remainder of the paper is structured as follows. Section 2 defines the problem setting and classifies the different problem cases investigated in this paper. Our complexity

results are presented in Sect. 3, and finally, Sect. 4 concludes the paper.

## 2 Problem definition and classification

We have a modular product, which is mass-customizable by customers specifying a given set  $C = \{c_1, \dots, c_m\}$  of  $m$  different product characteristics. For each characteristic  $c \in C$ , a customer can select from a given set of integer values  $V_c$  reflecting the alternative options. For our garage door example, the insulation is a product characteristic  $c$  and there is a set  $V_c$  of alternative insulation types the customers may choose from. All jobs, each corresponding to an order for a specific product customized according to a customer’s demand, are unified in job set  $J = \{1, \dots, n\}$ . Each job  $j \in J$  is defined by its processing time  $p_j$  and a value  $v_{j,c} \in V_c$  for each characteristic  $c \in C$ . Let  $K_c$  be the number of distinct values for characteristic  $c$  among the set of jobs, and let  $K = \max \{K_c \mid c \in C\}$ .

From now on, a characteristic  $c \in C$  with value  $v \in V_c$  refers to the product characteristic (e.g., the interior coating of a garage door) and its value (e.g., a specific kind of coating), but also to the specific resource (e.g., a steel coil) required by the machine and its specific resource type (e.g., a specific type of coil) to realize the value of the characteristic. Thus, we say processing job  $j$  with  $v_{j,c} \in V_c, c \in C$ , requires the machine to have value  $v_{j,c}$  of characteristic  $c$ . For each value  $v \in V_c, c \in C$ , we have a setup time  $s_{v,c}^e \geq 0$  for equipping characteristic  $c$  with value  $v$  and a setup time  $s_{v,c}^r \geq 0$  for removing value  $v$  from characteristic  $c$ .

A schedule is a sequence  $\sigma$  of jobs with  $\sigma(k)$  being the  $k$ th job in  $\sigma$ . For processing a job  $j$ , we have to configure the machine by setting each characteristic  $c \in C$  to  $v_{j,c} \in V_c$ . The actual setup time  $s_{i,j}$  between two jobs  $i$  and  $j$  depends on the values required by both jobs for each characteristic. The setup time  $s_{i,j,c}$  for characteristic  $c$  of job  $j$  immediately following  $i$  amounts to  $s_{v_i,c}^r + s_{v_j,c}^e$ , if  $v_{j,c} \neq v_{i,c}$  and to 0 otherwise. If setups for characteristics are conducted sequentially, then  $s_{i,j} = \sum_{c \in C} s_{i,j,c}$ . If setups for characteristics are conducted in parallel, then  $s_{i,j} = \max \{s_{i,j,c} \mid c \in C\}$ . The setup time  $s_{0,j}$  before the first job  $j = \sigma(1)$  amounts to  $s_{0,j} = \sum_{c \in C} s_{v_{j,c}}^e$  or  $s_{0,j} = \max \{s_{v_{j,c}}^e \mid c \in C\}$ . The setup time  $s_{j,0}$  after the last job  $j = \sigma(n)$  amounts to  $s_{j,0} = \sum_{c \in C} s_{v_{j,c}}^r$  or  $s_{j,0} = \max \{s_{v_{j,c}}^r \mid c \in C\}$ .

The makespan  $C_{\max}(\sigma)$  of a schedule  $\sigma$  is

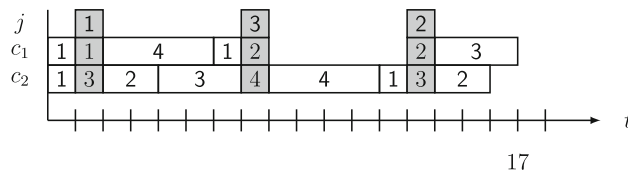
$$C_{\max}(\sigma) = s_{0,\sigma(1)} + \sum_{k=1}^{n-1} s_{\sigma(k),\sigma(k+1)} + s_{\sigma(n),0} + \sum_{j \in J} p_j .$$

**Table 1** Values for jobs and characteristics

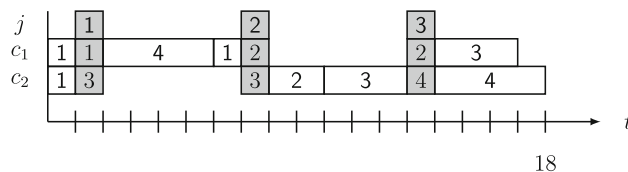
$v_{j,c}$	1	2	3
$c_1$	1	2	2
$c_2$	3	3	4

**Table 2** Setup times

	$V_{c_1}$		$V_{c_2}$	
	1	2	3	4
$s_{v,c}^e$	1	1	1	3
$s_{v,c}^r$	4	3	2	4



**Fig. 2**  $C_{\max} = 17$  for schedule (1, 3, 2) with parallel setups



**Fig. 3**  $C_{\max} = 18$  for schedule (1, 2, 3) with parallel setups

The goal in problem MODSETUP-SEQ and MODSETUP-PAR is to find a minimum makespan schedule when setups are conducted sequentially and in parallel, respectively.

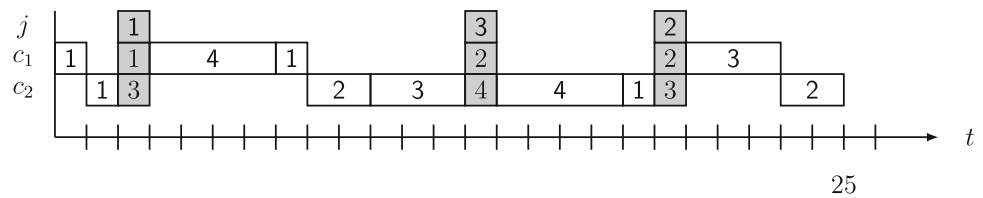
In order to improve intuition about the problem setting, we discuss a small example in the following. We consider job set  $J = \{1, 2, 3\}$ , characteristics  $c_1$  and  $c_2$ , value sets  $V_{c_1} = \{1, 2\}$  and  $V_{c_2} = \{3, 4\}$ , values for jobs and characteristics as given in Table 1, and setup times for equipping and removing as given in Table 2. Each job  $j \in J$  has unit processing time  $p_j = 1$ .

Figures 2, 3, 4 and 5 depict schedules corresponding to schedules (1, 3, 2) and (1, 2, 3) with parallel and sequential setups, respectively. The gray blocks represent jobs with the topmost entry referring to the job number  $j$ , the second entry representing  $v_{j,c_1}$ , and the third entry representing  $v_{j,c_2}$ . Setups are depicted as white blocks immediately before or after the job they belong to. The line a setup is depicted in refers to the corresponding characteristic. The entry of the white blocks reflects the length of the corresponding setup.

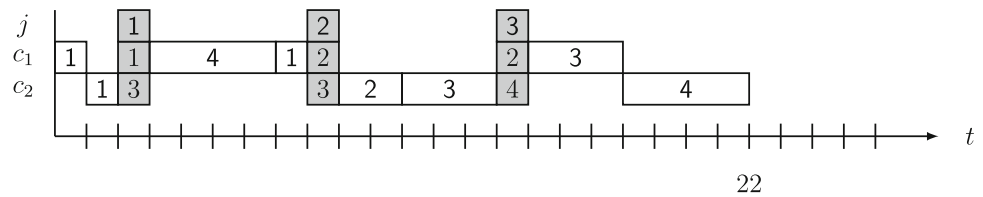
As we can see in the example, schedule (1, 3, 2) yields a smaller makespan for parallel setups, while schedule (1, 2, 3) finds better results for sequential setups.

In general, it is easy to see that the total processing time is a constant, so that, from now on, we will assume that  $p_j = 0$  and neglect processing times. In the following, we elaborate

**Fig. 4**  $C_{\max} = 25$  for schedule (1, 3, 2) with sequential setups



**Fig. 5**  $C_{\max} = 22$  for schedule (1, 2, 3) with sequential setups



how the different problem variants investigated in this paper are derived.

- **Setup aggregation** Our first distinction addresses the aggregation of setup times required for altering the divergent characteristics of two successive jobs. Here, we distinguish the two problem versions MODSETUP-SEQ (e.g., a single worker executes all setups for the different characteristics sequentially) and MODSETUP-PAR (e.g., each characteristic is serviced by a separate worker, so that all relevant setups are executed in parallel) elaborated above.
- **Removing time** Furthermore, we distinguish whether or not there are special setup times for removing values from characteristics required by the previous job. An extra removing time is to be considered, for instance, if removing the steel coils required by the previous jobs takes different amounts of time, e.g., due to varying weights or sizes. Once all removing times for the different values of a characteristic are equal, they can simply be added to the setup times related with equipping the machine for the subsequent job. Note that the removing time of the last job, then, is covered by the setup time for equipping of the first job. Note, furthermore, that due to symmetry the problem setting with setup times for equipping but without setup times for removing is equivalent to the problem setting with setup times for removing but without setup times for equipping.

- **Setup time differences** We call the standard case *value-dependent* setup time. Here, equipping any value  $v$  for characteristic  $c$  may take individual setup times  $s_{v,c}^e \geq 0$ . Analogously, individual setup times  $s_{v,c}^r \geq 0$  for removing  $v$  from  $c$  can occur. Furthermore, we distinguish the following special cases. We investigate the special case where setup times are only *characteristic-dependent*; that is, we have  $s_{v,c}^e = s_c^e$  and  $s_{v,c}^r = s_c^r$  for each characteristic  $c \in C$  and each value  $v \in V_c$ . Furthermore, we analyze the special case where  $s_c^e = 1$  and  $s_c^r = 1$  for each characteristic  $c \in C$ , so that all setup times are *equal*.

Given these distinctions, we receive twelve different problem settings to be investigated in the following section.

### 3 Analysis of computational complexity

This section analyzes the computational complexity of the problem variants defined in Sect. 2. The main results are summarized in Table 3.

We first establish the following properties, which justify a simplifying assumption concerning all problem variants to be treated later on.

**Lemma 1** *In both MODSETUP-SEQ and MODSETUP-PAR, setup times satisfy the triangle inequality.*

**Proof** Let us first consider setup times  $s_{i,j,c}$  and  $s_{j,k,c}$  for an arbitrary job triplet  $i \in J, j \in J$ , and  $k \in J$  and an arbitrary characteristic  $c \in C$ . We have

$$s_{i,j,c} + s_{j,k,c} = \begin{cases} s_{v_{i,c},c}^r + s_{v_{j,c},c}^e + s_{v_{j,c},c}^r + s_{v_{k,c},c}^e \geq s_{v_{i,c},c}^r + s_{v_{k,c},c}^e & \text{if } v_{i,c} \neq v_{j,c} \text{ and } v_{j,c} \neq v_{k,c} \\ s_{v_{j,c},c}^r + s_{v_{k,c},c}^e = s_{v_{i,c},c}^r + s_{v_{k,c},c}^e & \text{if } v_{i,c} = v_{j,c} \text{ and } v_{j,c} \neq v_{k,c} \\ s_{v_{i,c},c}^r + s_{v_{j,c},c}^e = s_{v_{i,c},c}^r + s_{v_{k,c},c}^e & \text{if } v_{i,c} \neq v_{j,c} \text{ and } v_{j,c} = v_{k,c} \\ 0 = s_{i,k,c} & \text{if } v_{i,c} = v_{j,c} = v_{k,c} \end{cases}$$

**Table 3** Subproblems and their complexity status

setup times	MODSETUP-SEQ		MODSETUP-PAR	
	no removing	removing	no removing	removing
equal $s_c^e = s_c^r = 1$	strongly NP-hard even if $m = 2$ (Theorem 1) or if $K = 2$ (Theorem 2)		$\mathcal{O}(n \cdot m \cdot K)$ Theorem 3	
characteristic-dependent $s_c^e \geq 0, s_c^r \geq 0$			open	strongly NP-hard even if $m = 6$ (Theorem 4) or if $K = 3$ (Theorem 5)
value-dependent $s_{v,c}^e \geq 0, s_{v,c}^r \geq 0$				

Note that the above carries over to setups before the first job  $j$  and between the first two jobs  $j$  and  $k$  in a schedule as well as to setups after the last job  $j$  and between the last two jobs  $i$  and  $j$  in a schedule. To see this, in the former case we can set  $s_{v_i,c,c}^r = 0$  and in the latter case, we can set  $s_{v_k,c,c}^e = 0$ .

Now, we consider setup times  $s_{i,j} = \sum_{c \in C} s_{i,j,c}$  and  $s_{j,k} = \sum_{c \in C} s_{j,k,c}$  for an arbitrary job triplet  $i, j$ , and  $k$  in MODSETUP- SEQ. We obtain

$$\begin{aligned}
 s_{i,j} + s_{j,k} &= \sum_{c \in C} s_{i,j,c} + \sum_{c \in C} s_{j,k,c} \\
 &= \sum_{c \in C} (s_{i,j,c} + s_{j,k,c}) \geq \sum_{c \in C} s_{i,k,c} = s_{i,k}.
 \end{aligned}$$

Finally, we consider setup times  $s_{i,j} = \max \{s_{i,j,c} \mid c \in C\}$  and  $s_{j,k} = \max \{s_{j,k,c} \mid c \in C\}$  for an arbitrary job triplet  $i, j$ , and  $k$  in MODSETUP- PAR. We obtain

$$\begin{aligned}
 s_{i,j} + s_{j,k} &= \max \{s_{i,j,c} \mid c \in C\} + \max \{s_{j,k,c} \mid c \in C\} \\
 &\geq \max \{s_{i,j,c} + s_{j,k,c} \mid c \in C\} \geq \max \{s_{i,k,c} \mid c \in C\} = s_{i,k}.
 \end{aligned}$$

Again, this result carries over to setups before the first job  $j$  and between the first two jobs  $j$  and  $k$  in a schedule as well as to setups after the last job  $j$  and between the last two jobs  $i$  and  $j$  in a schedule.  $\square$

The fact that the triangle inequality is satisfied, hence, means that the approximability results by Svensson et al. (2017, 2018) and Traub and Vygen (2020) for the asymmetric TSP (with the triangle inequality being satisfied) carry over to both MODSETUP- SEQ and MODSETUP- PAR. Svensson et al. (2018) show that the asymmetric TSP can be approximated within a factor of 5,500 in polynomial time, and Svensson et al. (2017) refine this result to a factor of 506. Most recently, Traub and Vygen (2020) further reduced the factor to  $22 + \epsilon$  for any  $\epsilon > 0$ .

**Lemma 2** *There is an optimal schedule for MODSETUP- SEQ (MODSETUP- PAR) where jobs coinciding in the value of each characteristic are scheduled consecutively.*

**Proof** Consider an arbitrary optimal schedule and the first job  $i$  that does not immediately follow the previous job  $j$  with identical values for each characteristic. If there is no such job, we are done. Due to Lemma 1, we do not increase total setup time by removing  $i$  from its current position and inserting it immediately after  $j$ . By repeating this step, we find an optimal schedule where jobs coinciding in the value of each characteristic are scheduled consecutively.  $\square$

**Corollary 1** *Both MODSETUP- SEQ and MODSETUP- PAR can be solved in  $\mathcal{O}(n \cdot m \cdot K + (m \cdot K)!)$  time and, thus, in  $\mathcal{O}(n)$  time, if  $m$  and  $K$  are fixed.*

**Proof** We can cluster jobs with identical values for each characteristic by sorting in  $\mathcal{O}(n \cdot m \cdot K)$  time using radix sort. We then have a fixed number  $m \cdot K$  of clusters to be sequenced. Evaluating each of the  $(m \cdot K)!$  sequences can be done in constant time, if  $m$  and  $K$  are fixed.  $\square$

In the following, we will assume that we do not have jobs coinciding in all values of characteristics. We can ensure this by a preprocessing step that clusters jobs accordingly. Such a preprocessing step clustering jobs with identical values for all characteristics can be conducted in  $\mathcal{O}(n \cdot m \cdot K)$  time using radix sort or in  $\mathcal{O}((n \log n) \cdot m)$  time with comparison-based sorting. After the preprocessing step, we handle clusters as jobs.

### 3.1 Sequential setups

This section analyzes the computational complexity of those problem cases where setups are executed sequentially, i.e., we have  $s_{i,j} = \sum_{c \in C} s_{i,j,c}$  for all  $i, j \in J$  with  $i \neq j$ . Unfortunately, the problem turns out to be strongly NP-hard not only in the most restricted case, but also if we additionally restrict the setting to no more than  $m = 2$  characteristics. We show NP-hardness by reduction from the problem to find a Hamiltonian path in a cubic graph, abbreviated to HAM- CUBE, which is strongly NP-complete, see Garey and Johnson (1979).

**Theorem 1** *ModSetup-Seq with  $s_{v,c}^e = 1$  for each characteristic  $c \in C$  and each value  $v \in V_c$  and without setups for removing is strongly NP-hard even if  $m = 2$ .*

**Proof** We consider an arbitrary instance  $I$  of HAM-CUBE given by a cubic graph  $G = (V, E)$ . Let  $v_j^1, v_j^2$ , and  $v_j^3$  be the three nodes adjacent to  $j \in V$ . We first construct another instance  $I'$  of HAM-CUBE specified by a graph  $G' = (V', E')$  as follows.

We replace each node in  $V$  by a triangle in  $G'$  where each of the nodes in a triangle establishes the connection to one other triangle. Connections among triangles correspond to edges in  $E$ . Let  $j^i \in V'$  be the node in the triangle replacing  $j \in V$  that connects its triangle to the triangle replacing  $i \in V$ . Furthermore, we define  $\tilde{E}$ , the set of edges between different triangles. Punim et al. (2007) show that there exists a Hamiltonian cycle in  $G'$ , if and only if there exists one in  $G$ . The idea of the proof can be easily adapted to show that there is a Hamiltonian path in  $G'$ , if and only if there is one in  $G$ .

Now, we construct an instance  $I''$  of MODSETUP-SEQ. We set  $J = V'$  and  $m = 2$ . Jobs  $j^{k_1}, j^{k_2}$ , and  $j^{k_3}$  have  $v_{j^{k_1},1} = v_{j^{k_2},1} = v_{j^{k_3},1} = j$  for each  $j \in J$ . Jobs  $j^i$  and  $i^j$  have  $v_{j^i,2} = v_{i^j,2} = \{j, i\}$  for each  $\{j^i, i^j\} \in \tilde{E}$ . Finally, we have  $s_{v,c}^e = 1$  for each value  $v$  of each characteristic  $c$  and do not consider setups for removing. This completes the reduction which is pseudo-polynomial, see Garey and Johnson (1979, p. 101).

Summarizing, we have a job for each node in  $V'$ . Two jobs coincide in the value of the first characteristic, if they belong to the same triangle in  $G'$  (and, therefore, correspond to the same node in  $G$ ). Two jobs coincide in the value of the second characteristic, if they belong to different triangles and are connected by an edge in  $G'$  (and, therefore, correspond to an edge in  $G$ ).

We claim that there is a schedule with makespan of at most  $|V'| + 1$  for  $I''$ , if and only if there is a Hamiltonian path in  $G'$ .

- If there is a Hamiltonian path in  $G'$ , then we can construct a schedule for  $I''$  by having the jobs in the same order as the corresponding nodes in the Hamiltonian path in  $G'$ . We, then, have setup time of 2 prior to the first job. Furthermore, we have a setup time of 1 between each pair of consecutive jobs. This is true, because two jobs differ in exactly one characteristic's value, if the corresponding nodes in  $G'$  are adjacent. Hence, there is a schedule for  $I''$  with makespan of at most  $2 + (|V'| - 1) = |V'| + 1$ .
- Now, assume there is a schedule with a makespan of at most  $|V'| + 1$ . We have a setup time of 2 prior to the first job, which leaves  $|V'| - 1$  total setup time for setups between pairs of consecutive jobs. Since there are no identical jobs, we have a setup time of at least 1 for each

such pair. Hence, each pair of consecutive jobs corresponds to adjacent nodes in  $G'$ , since non-adjacent nodes in  $G'$  incur a setup time of 2.

Hence, there is schedule with a makespan of at most  $|V'| + 1$  for  $I''$ , if and only if there is a Hamiltonian path in  $G$ .  $\square$

To illustrate the relation between instances  $I, I'$ , and  $I''$  in the proof, we provide an example in the following. Figure 6a depicts an example instance  $I$  with  $V = \{1, \dots, 6\}$ . The modified graph  $G'$  specifying instance  $I'$  is depicted in Fig. 6b. Here, we have a triangle of nodes for each node in  $G$ . Two triangles are connected by an edge in  $E'$ , if and only if the corresponding nodes in  $V$  are connected by an edge in  $E$ . Each node in a triangle connects its triangle to exactly one other triangle.

The Hamiltonian path (1, 6, 5, 2, 3, 4) in  $G$  corresponds to, e.g., the Hamiltonian path

$$(1^4, 1^5, 1^6, 6^1, 6^3, 6^5, 5^6, 5^1, 5^2, 2^5, 2^4, 2^3, 3^2, 3^6, 3^4, 4^3, 4^2, 4^1)$$

in  $G'$ . The other way round, the Hamiltonian path

$$(1^4, 4^1, 4^2, 4^3, 3^4, 3^2, 3^6, 6^3, 6^5, 6^1, 1^6, 1^5, 5^1, 5^6, 5^2, 2^5, 2^4, 2^3)$$

in  $G'$ , where nodes  $1^4, 1^5$ , and  $1^6$  belonging to the same triangle are not visited consecutively, implies a Hamiltonian path

$$(4^1, 4^2, 4^3, 3^4, 3^2, 3^6, 6^3, 6^5, 6^1, 1^6, 1^4, 1^5, 5^1, 5^6, 5^2, 2^5, 2^4, 2^3)$$

in  $G'$ , where nodes of each triangle are visited consecutively. The latter Hamiltonian path in  $G'$  implies Hamiltonian path (4, 3, 6, 1, 5, 2) in  $G$ .

Figure 6c highlights for each job corresponding to a node in  $G'$  the values of the characteristics in instance  $I''$ . We can see that the first value coincides with the corresponding node in  $G$ , while the second value reflects the corresponding edge in  $G$ .

Let us once more consider Hamiltonian path

$$(4^1, 4^2, 4^3, 3^4, 3^2, 3^6, 6^3, 6^5, 6^1, 1^6, 1^4, 1^5, 5^1, 5^6, 5^2, 2^5, 2^4, 2^3)$$

in  $G'$ . If we have the corresponding schedule for  $I''$ , then we obtain a total setup time of  $2 + (|V'| - 1) = 2 + (18 - 1) = 19$  since two consecutive jobs coincide in exactly one characteristic's value.

Note that Theorem 1 states strong NP-hardness for the most restricted version of MODSETUP-SEQ with  $s_{v,c}^e = 1$  for each characteristic  $c \in C$  and each value  $v \in V_c$  and without setups for removing. Hence, all other problem cases MODSETUP-SEQ share this complexity status. However, we add another complexity result that proves strong

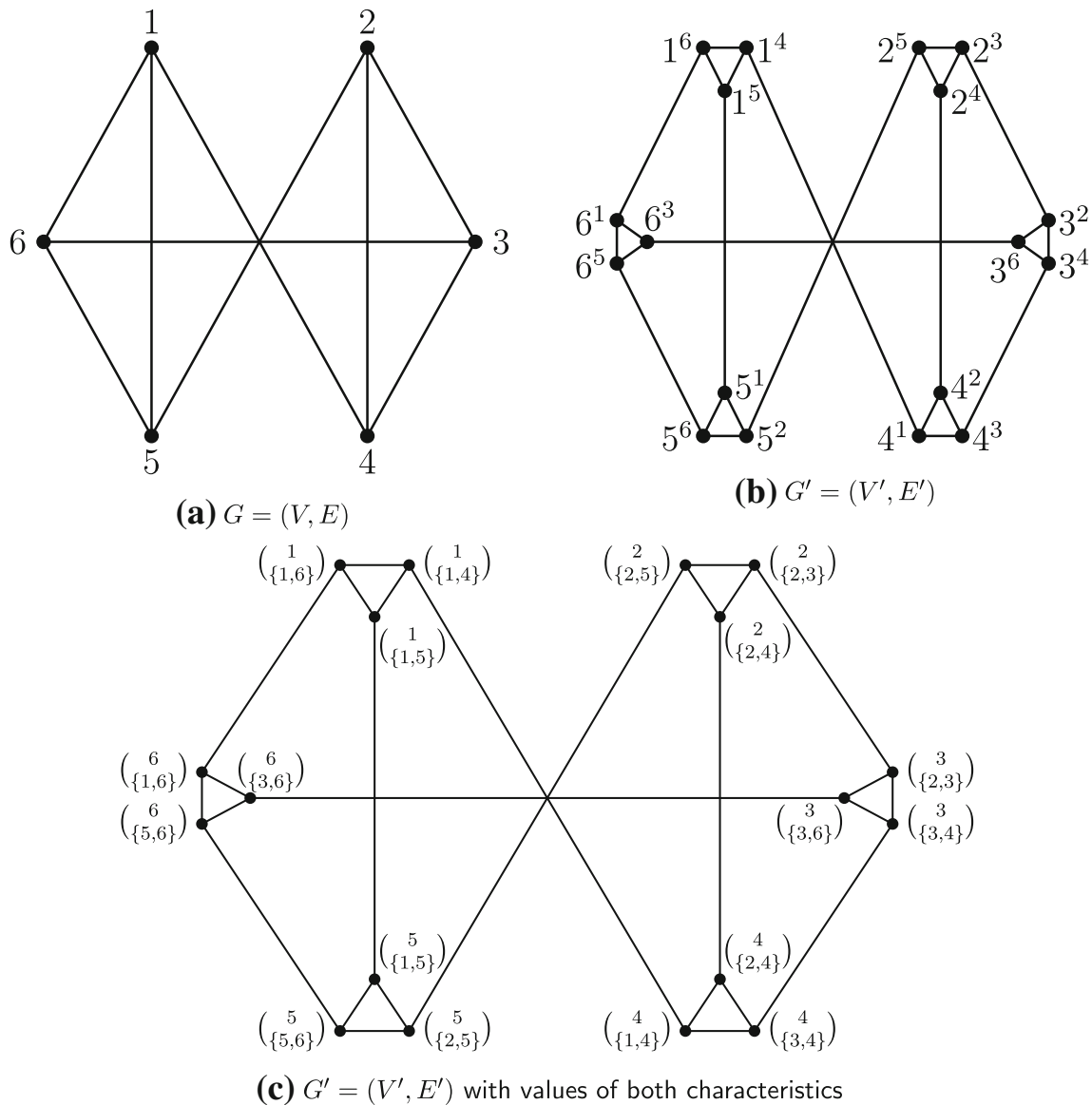


Fig. 6 Example instances  $I, I',$  and  $I''$  in the proof of Theorem 1

NP-hardness, if instead the maximum number  $K$  of distinct values over all characteristics is restricted. Again, we show NP-hardness by reduction from the problem to find a Hamiltonian path in a cubic graph, abbreviated to HAM-CUBE.

**Theorem 2** MODSETUP-SEQ with  $s_{v,c}^e = 1$  for each characteristic  $c \in C$  and each value  $v \in V_c$  and without setups for removing is strongly NP-hard even if  $K = 2$ .

**Proof** We consider an arbitrary instance  $I$  of HAM-CUBE given by a cubic graph  $G = (V, E)$  and construct an instance  $I'$  of MODSETUP-SEQ as follows. We set  $J = V$  and  $m = |E|$ . We consider a one-to-one relationship between edges in  $E$  and characteristics. We refer to the characteristic associated with  $e \in E$  as  $c_e$ . For each  $j \in J$  and each  $e \in E$ , we have  $v_{j,e} = 1$  if  $e$  is incident to  $j$ , and  $v_{j,e} = 0$  other-

wise. Finally, we have  $s_{1,c}^e = s_{0,c}^e = 1$  for each characteristic  $c$  and do not consider setups for removing. This completes the reduction which is pseudo-polynomial.

Summarizing, we have a job for each node in  $V$ . Two distinct jobs  $j$  and  $j'$  have  $v_{j,e} = v_{j',e} = 1$  for a characteristic  $c_e$ , if and only if nodes  $j$  and  $j'$  are adjacent in  $G$  and  $e$  is the connecting edge. Since  $G$  is cubic, each job  $j$  has  $v_{j,e} = 1$  for exactly three characteristics, and we denote the set of these characteristics as  $C_j$ . We refer to them as the characteristics of job  $j$ .

We claim that there is a schedule with a makespan of at most  $5.5 \cdot |V| - 4$  for  $I'$ , if and only if there is a Hamiltonian path in  $G$ .



We, first, have a look at the total setup time for characteristic  $c_e$  with  $e = \{j, j'\}$  in a given schedule. Note that a setup time of 1 is due before the first job in any schedule. Without loss of generality, we assume that  $j$  is scheduled before  $j'$ .

- If  $j$  and  $j'$  are scheduled in the first two positions or in the last two positions, then total setup time for  $c_e$  is 2.
- If  $j$  and  $j'$  are scheduled in consecutive positions  $p$  and  $p + 1$ , respectively, with  $p = 2, \dots, n - 2$  or in positions 1 and  $n$ , then total setup time for  $c_e$  is 3.
- If  $j$  and  $j'$  are scheduled in non-consecutive positions  $p$  and  $p'$ , respectively, with  $p = 1$  and  $2 < p' < n$  or  $1 < p < n - 1$  and  $p' = n$ , then total setup time for  $c_e$  is 4.
- If  $j$  and  $j'$  are scheduled in non-consecutive positions  $p$  and  $p'$ , respectively, with  $1 < p < p' - 1 < n' - 1$ , then total setup time for  $c_e$  is 5.

Now, we analyze the total setup time in a given schedule  $\sigma$  and distinguish two cases.

- If there is a characteristic  $c_e \in C_{\sigma(1)} \cap C_{\sigma(n)}$ , then the total setup time for characteristics in  $C_{\sigma(1)} \cup C_{\sigma(n)}$  is at least

$$LB_1 = \underbrace{1 \cdot 3}_{c_e} + \underbrace{2 \cdot 2}_{LB'_1} + 2 \cdot 4 = 15.$$

Obviously, the total setup time for  $c_e$  is 3. If and only if  $C_{\sigma(1)} \cap C_{\sigma(2)} \neq \emptyset$  ( $C_{\sigma(n-1)} \cap C_{\sigma(n)} \neq \emptyset$ ), the total setup time for the characteristic in  $C_{\sigma(1)} \cap C_{\sigma(2)}$  (in  $C_{\sigma(n-1)} \cap C_{\sigma(n)}$ ) is 2 leading to partial lower bound  $LB'_1$ . The total setup time for at least one characteristic of  $\sigma(1)$  and at least one characteristic of  $\sigma(n)$  is 4 leading to the last term in  $LB_1$ . Hence, total setup time for characteristics in  $C_{\sigma(1)} \cup C_{\sigma(n)}$  equals 15, if  $C_{\sigma(1)} \cap C_{\sigma(2)} \neq \emptyset$  and  $C_{\sigma(n-1)} \cap C_{\sigma(n)} \neq \emptyset$  and exceeds 15 otherwise.

Among the  $1.5 \cdot |V| - 5$  other characteristics at most  $|V| - 3$  can have total setup time of 3. In fact, if  $C_{\sigma(k)} \cap C_{\sigma(k+1)} \neq \emptyset, k = 2, \dots, n - 2$ , then total processing time of  $c_e$  with  $e = \{\sigma(k), \sigma(k + 1)\}$  is 3. This leads to a total setup time for the  $1.5 \cdot |V| - 5$  remaining characteristics of

$$S_1(q) = 3 \cdot (|V| - 3 - q) + 5 \cdot (0.5 \cdot |V| - 2 + q) = 5.5 \cdot |V| - 19 + 2 \cdot q.$$

with  $q$  being the number of pairs of consecutive jobs in  $\sigma(2), \dots, \sigma(n - 1)$  having no characteristic in common. We, thus, obtain a lower bound of the total setup time for all characteristics of

$$LB(q) = LB_1 + S_1(q) = 15 + 5.5 \cdot |V| - 19 + 2 \cdot q$$

$$= 5.5 \cdot |V| - 4 + 2 \cdot q.$$

Furthermore, a total setup time of  $LB(0) = 5.5 \cdot |V| - 4$  is achieved, if and only if  $C_{\sigma(k)} \cap C_{\sigma(k+1)} \neq \emptyset$  for each  $k = 1, \dots, n - 1$ .

- If  $C_{\sigma(1)} \cap C_{\sigma(n)} = \emptyset$ , then the total setup time for characteristics in  $C_{\sigma(1)} \cup C_{\sigma(n)}$  is at least

$$LB_2 = \underbrace{2 \cdot 2}_{LB'_2} + 4 \cdot 4 = 20.$$

Partial lower bound  $LB'_2$  is motivated as  $LB'_1$  above. The total setup time for the two remaining characteristics of  $\sigma(1)$  and  $\sigma(n)$ , respectively, is 4 leading to the last term in  $LB_2$ . Hence, the total setup time for the characteristics in  $C_{\sigma(1)} \cup C_{\sigma(n)}$  equals 20, if  $C_{\sigma(1)} \cap C_{\sigma(2)} \neq \emptyset$  and  $C_{\sigma(n-1)} \cap C_{\sigma(n)} \neq \emptyset$  and exceeds 20 otherwise.

For the  $1.5 \cdot |V| - 6$  remaining characteristics, we can derive a total setup time of

$$S_2(q) = 3 \cdot (|V| - 3 - q) + 5 \cdot (0.5 \cdot |V| - 3 + q) = 5.5 \cdot |V| - 24 + 2 \cdot q.$$

We, thus, obtain a lower bound

$$LB(q) = LB_2 + S_2(q) = 20 + 5.5 \cdot |V| - 24 + 2 \cdot q = 5.5 \cdot |V| - 4 + 2 \cdot q.$$

Furthermore, a total setup time of  $LB(0) = 5.5 \cdot |V| - 4$  is achieved, if and only if  $C_{\sigma(k)} \cap C_{\sigma(k+1)} \neq \emptyset$  for each  $k = 1, \dots, n - 1$ .

For both cases, we see that the lower bound of  $5.5 \cdot |V| - 4$  of the total setup time (and thus of the makespan) is achieved, if and only if  $C_{\sigma(k)} \cap C_{\sigma(k+1)} \neq \emptyset$  for each  $k = 1, \dots, n - 1$ . Hence, a schedule with makespan  $5.5 \cdot |V| - 4$  implies a Hamiltonian path in  $G$  and vice versa. This completes the proof.  $\square$

Summarizing, the most restricted version of MODSETUP-SEQ considered in this paper is strongly NP-hard, even if either  $m = 2$  or  $K = 2$ . Since MODSETUP-SEQ can be solved in polynomial time if both  $m$  and  $K$  are fixed, see Corollary 1, this completes the picture.

### 3.2 Parallel setups

This section analyzes the computational complexity of those problem cases where setups are executed in parallel, i.e., we have  $s_{i,j} = \max \{s_{i,j,c} \mid c \in C\}$  for all  $i, j \in J$  with  $i \neq j$ .

First, we develop a positive result for the special case with  $s_{v,c}^e = s_c^e$  and  $s_{v,c}^r = s_c^r$  for each characteristic  $c \in C$  and each value  $v \in V_c$ . Without loss of generality, we assume that  $s_{c_k}^e + s_{c_k}^r \geq s_{c_{k+1}}^e + s_{c_{k+1}}^r$  for each  $k = 1, \dots, m - 1$ .

**Table 4** Values for jobs and characteristics

$v_{j,c}$	1	2	3	4	5	6
$c_1$	1	2	2	3	3	2
$c_2$	4	6	5	5	6	6
$c_3$	8	8	7	9	9	9

**Lemma 3** *There is an optimal schedule to MODSETUP- PAR with  $s_{v,c}^e = s_c^e$  and  $s_{v,c}^r = s_c^r$ , where all jobs having the same value for  $c_1$  are scheduled consecutively.*

**Proof** Consider an optimal schedule where some jobs with the same value for  $c_1$  are not scheduled consecutively. Let  $j$  be the first job, which is followed by an immediate successor  $s(j)$  with  $v_{j,c_1} \neq v_{s(j),c_1}$  although there is a (non-immediate) successor  $j'$  with  $v_{j,c_1} = v_{j',c_1}$ . If there is more than one such successor, we assume that  $j'$  is the first one. Let  $p(j')$  be the immediate predecessor of  $j'$ .

We modify the schedule as follows. We postpone the subsequence of jobs  $(s(j), \dots, p(j'))$  and move it to the end of the schedule. Now,  $j'$  immediately follows  $j$ . The total setup time changes as follows.

- In the original schedule, the setup time between  $s(j)$  and its immediate predecessor  $j$  and between  $j'$  and its immediate predecessor  $p(j')$  is  $2(s_{c_1}^r + s_{c_1}^e)$ . The last job in the schedule causes  $\max \{s_c^r \mid c \in C\}$  setup time for removing.
- In the new schedule, the setup time between  $s(j)$  and its immediate predecessor is at most  $s_{c_1}^r + s_{c_1}^e$  and the setup time between  $j'$  and its immediate predecessor  $j$  is at most  $s_{c_2}^r + s_{c_2}^e \leq s_{c_1}^r + s_{c_1}^e$ . Here, also, the last job in the schedule causes  $\max \{s_c^r \mid c \in C\}$  setup time for removing.

Since the total setup time does not increase, the makespan does not increase either. Hence, the modification yields an optimal schedule. By repeating this step, we can achieve an optimal schedule as specified in the lemma.  $\square$

To illustrate the basic idea of the proof of Lemma 3, we consider job set  $J = \{1, \dots, 6\}$ , characteristics  $c_1, c_2$ , and  $c_3$ , and value sets  $V_{c_1} = \{1, 2, 3\}$ ,  $V_{c_2} = \{4, 5, 6\}$ , and  $V_{c_3} = \{7, 8, 9\}$ . The values for these jobs and characteristics are given in Table 4, and the setup times for equipping and removing are given in Table 5. Note that we have  $s_{v,c}^e = s_c^e$  and  $s_{v,c}^r = s_c^r$ .

Figure 7 depicts a schedule where not all jobs having the same value for  $c_1$  are scheduled consecutively. In this example, we have  $j = 3$  and  $j' = 6$ .

The schedule in Fig. 8 is obtained from the schedule in Fig. 7 by applying the modification used in the Proof of Lemma 3. We postpone the subsequence  $(s(j), \dots, p(j'))$ , which

**Table 5** Setup times

	$V_{c_1}$			$V_{c_2}$			$V_{c_3}$		
	1	2	3	4	5	6	7	8	9
$s_{v,c}^e$	3			1			4		
$s_{v,c}^r$	4			5			1		

consists of jobs 4 and 5, and move it to the end of the schedule. This modification reduces the makespan by one time unit.

**Theorem 3** *MODSETUP- PAR with  $s_{v,c}^e = s_c^e$  and  $s_{v,c}^r = s_c^r$  for each characteristic  $c \in C$  and each value  $v \in V_c$  can be solved in  $\mathcal{O}(n \cdot m \cdot K)$ .*

**Proof** Lemma 3 allows to cluster jobs according to their values of  $c_1$ . We refer to these clusters as  $c_1$ -clusters. Note that between each pair of  $c_1$ -clusters we have a total setup time of  $s_{v,c_1}^r + s_{v,c_1}^e$ . Before the first  $c_1$ -cluster, we have a setup time of  $\max \{s_{c_k}^e \mid c_k \in C\}$  and after the last  $c_1$ -cluster, we have a setup time of  $\max \{s_{c_k}^r \mid c_k \in C\}$ . Hence, we can sequence  $c_1$ -clusters in an arbitrary order. Then, jobs in each  $c_1$ -cluster can be sequenced independently.

Note that within each  $c_1$ -cluster, we can apply Lemma 3 with respect to  $c_2$  instead of  $c_1$ . Hence, we arrange  $c_2$ -clusters within each  $c_1$ -cluster that can be sequenced in arbitrary order. At this point, we can assume that there is no setup before the first job and after the last job in each  $c_1$ -cluster, because the maximum possible time has been considered already between  $c_1$ -clusters (and before the first and after the last  $c_1$  cluster). Following this line of argument, we arrange  $c_{k+1}$  within  $c_k$ -cluster for  $k = 1, \dots, m-1$ . Due to Lemma 3, this yields an optimal schedule. We can implement this cluster schedule by sorting jobs lexicographically using radix sort in  $\mathcal{O}(n \cdot m \cdot K)$ .  $\square$

For the instance depicted in Figs. 7 and 8, we have an optimal schedule derived by sorting jobs lexicographically illustrated in Fig. 9.

The most general case of MODSETUP- PAR, however, turns out to be NP-hard. We show this result by a transformation from the path version of the geometric traveling salesman problem with maximum metric, abbreviated to PATH- TSP- MAX. Note that the geometric traveling salesman problem with maximum metric is strongly NP-hard, see Garey and Johnson (1979). It follows easily that the problem to find a shortest path with a given start point  $u$  and a given end point  $v$  that visits all other points exactly once is strongly NP-hard, as well.

**PATH-TSP-MAX** : Given a set  $P \subseteq Z \times Z$  of (integer) points in the plane, two specified points  $u, v \in P$ , and a positive integer  $B$ , is there a simple path of length  $B$  or less (with respect to the maximum metric) that starts in  $u$ , ends in  $v$ , and visits every point exactly once?

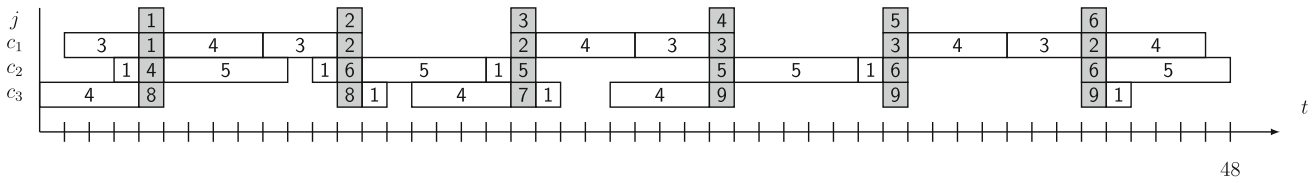


Fig. 7 Schedule with jobs having the same value for \$c\_1\$ not scheduled consecutively

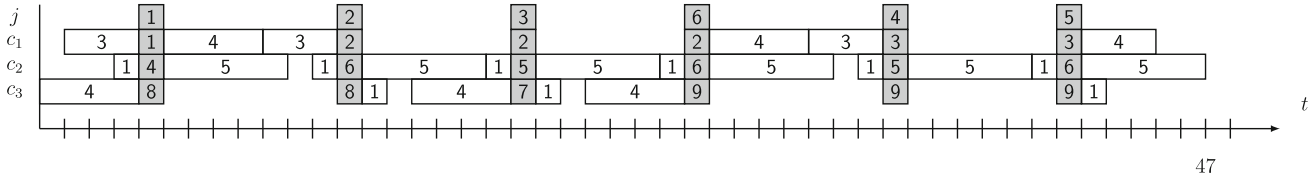


Fig. 8 Schedule with jobs having the same value for \$c\_1\$ scheduled consecutively

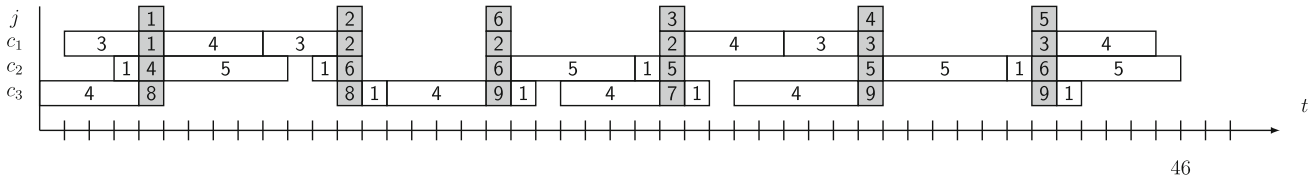


Fig. 9 Optimal schedule

**Theorem 4** MODSETUP- PAR is strongly NP-hard, even if \$m = 6\$.

**Proof** For an arbitrary instance \$I\$ of PATH- TSP- MAX, e.g., a set \$P\$ of points, a start point \$u \in P\$, an end point \$v \in P\$, and an integer \$B\$, we construct an instance \$I'\$ of MODSETUP- PAR with \$m = 6\$ characteristics \$c\_1\$ to \$c\_6\$ as follows. Without loss of generality, we assume the smallest coordinates in \$I\$ in both dimensions to be zero. Additionally, we assume that coordinates are not larger than \$B\$, since otherwise the answer to \$I\$ trivially is no. Summarizing, we assume \$0 \le p\_x \le B\$ and \$0 \le p\_y \le B\$ for each \$p = (p\_x, p\_y) \in P\$.

For each point \$p \in P\$, we introduce a job \$p\$ requiring unique values \$v\_{p,c\_1} = p\_1, v\_{p,c\_2} = p\_2, v\_{p,c\_3} = p\_3, v\_{p,c\_4} = p\_4, v\_{p,c\_5} = p\_5\$, and \$v\_{p,c\_6} = p\_6\$ for characteristics \$c\_1, c\_2, c\_3, c\_4, c\_5\$, and \$c\_6\$, respectively. For each job \$p, p \neq u\$ and \$p \neq v\$, the resulting setup times for equipping and removing are

- \$s\_{p\_1,c\_1}^e = p\_x\$ and \$s\_{p\_1,c\_1}^r = B - p\_x\$ for characteristic \$c\_1\$,
- \$s\_{p\_2,c\_2}^e = B - p\_x\$ and \$s\_{p\_2,c\_2}^r = p\_x\$ for \$c\_2\$,
- \$s\_{p\_3,c\_3}^e = p\_y\$ and \$s\_{p\_3,c\_3}^r = B - p\_y\$ for \$c\_3\$,
- \$s\_{p\_4,c\_4}^e = B - p\_y\$ and \$s\_{p\_4,c\_4}^r = p\_y\$ for \$c\_4\$,
- \$s\_{p\_5,c\_5}^e = B\$ and \$s\_{p\_5,c\_5}^r = 0\$ for \$c\_5\$ and
- \$s\_{p\_6,c\_6}^e = 0\$ and \$s\_{p\_6,c\_6}^r = B\$ for \$c\_6\$.

For jobs \$u\$ and \$v\$, setup times for equipping and removing are

- \$s\_{u\_1,c\_1}^e = 0, s\_{u\_1,c\_1}^r = v\_x, s\_{v\_1,c\_1}^e = B - u\_x\$, and \$s\_{v\_1,c\_1}^r = 0\$ for \$c\_1\$,

- \$s\_{u\_2,c\_2}^e = 0, s\_{v\_2,c\_2}^e = B - v\_x, s\_{u\_2,c\_2}^r = u\_x\$, and \$s\_{v\_2,c\_2}^r = 0\$ for \$c\_2\$,
- \$s\_{u\_3,c\_3}^e = 0, s\_{v\_3,c\_3}^e = v\_y, s\_{u\_3,c\_3}^r = B - u\_y\$, and \$s\_{v\_3,c\_3}^r = 0\$ for \$c\_3\$,
- \$s\_{u\_4,c\_4}^e = 0, s\_{v\_4,c\_4}^e = B - v\_y, s\_{u\_4,c\_4}^r = u\_y\$, and \$s\_{v\_4,c\_4}^r = 0\$ for \$c\_4\$,
- \$s\_{u\_5,c\_5}^e = 0, s\_{v\_5,c\_5}^e = B, s\_{u\_5,c\_5}^r = 0\$, and \$s\_{v\_5,c\_5}^r = 0\$ for \$c\_5\$ and
- \$s\_{u\_6,c\_6}^e = 0, s\_{v\_6,c\_6}^e = 0, s\_{u\_6,c\_6}^r = B\$, and \$s\_{v\_6,c\_6}^r = 0\$ for \$c\_6\$.

This completes the reduction which is pseudo-polynomial. The question we ask is whether there is a schedule with a total setup time of at most \$|P| \cdot B\$.

A parallel setup of characteristics \$c\_1\$ and \$c\_2\$ from a job \$p (p \neq v)\$ to a job \$q (q \neq u)\$ requires \$\max\{B - p\_x + q\_x; p\_x + B - q\_x\} = B + \max\{q\_x - p\_x; p\_x - q\_x\}\$ time units. Analogously, a parallel setup of \$c\_3\$ and \$c\_4\$ from \$p (p \neq v)\$ to \$q (q \neq u)\$ requires \$B + \max\{q\_y - p\_y; p\_y - q\_y\}\$. Thus, a parallel setup of all characteristics \$c\_1\$ to \$c\_6\$ from \$p (p \neq v)\$ to \$q (q \neq u)\$ requires \$B + \max\{q\_x - p\_x; p\_x - q\_x; q\_y - p\_y; p\_y - q\_y; 0\} = B + d\_\infty(p, q)\$ time units with \$d\_\infty(p, q)\$ denoting the distance between the corresponding points \$p\$ and \$q\$ in \$I\$. Consequently, a YES-certificate of \$I\$ with a length at most \$B\$ directly corresponds to a YES-certificate of \$I'\$ with the total setup time of \$|P| \cdot B\$.

For the opposite direction, suppose an arbitrary schedule for \$I'\$ with a total setup time of at most \$|P| \cdot B\$. We, first, establish that this schedule necessarily starts with \$u\$ and ends with \$v\$. The total setup time for both \$c\_5\$ and \$c\_6\$ is \$(|P| - 1) \cdot B\$.

Hence, we have a total setup time for  $c_5$  and  $c_6$  combined of at least  $|P| \cdot B$  (and, thus a total setup time of more than  $|P| \cdot B$  since setup times between jobs for  $c_1$  to  $c_4$  exceed  $B$ ) unless the positive setup times for  $c_5$  and  $c_6$  occur in the same  $|P| - 1$  gaps between jobs. The latter can only be achieved when having  $u$  and  $v$  in the first and the last position, respectively.

Thus, the schedule directly corresponds to a path from  $u$  to  $v$  and due to the analogy between setup times between jobs and distances between points the path has a length of at most  $B$ . This completes the proof.  $\square$

Next, we add another complexity result that proves strong NP-hardness, if instead the maximum number  $K$  of distinct values over all characteristics is restricted. We show NP-hardness by a reduction from the problem to find a Hamiltonian path in a graph, abbreviated to HAM, which is strongly NP-complete, see Garey and Johnson (1979).

**Theorem 5** MODSETUP- PAR is strongly NP-hard, even if  $K = 3$  and  $s_{v,c}^e = s_{v,c}^r$  for all  $c \in C$  and  $v \in V_c$ .

**Proof** We consider an arbitrary instance  $I$  of HAM given by a graph  $G = (V, E)$  with  $V = \{1, \dots, |V|\}$ . We construct an instance  $I'$  of MODSETUP- PAR as follows.

- For each vertex  $i \in V$ , we have a job  $i$  in  $I'$ .
- For each pair  $(j, k)$  of vertices with  $j < k$  of  $I$ , we introduce a characteristic  $c_{j,k}$  in  $I'$ . In the following, we call these characteristics *edge-characteristics*.
  - The set of values for edge-characteristic  $c_{j,k}$  is  $V_{j,k} = \{j, k, 0\}$ .
  - For edge-characteristic  $c_{j,k}$ , job  $j$  receives value  $v_{j,c_{j,k}} = j$ , job  $k$  has value  $v_{k,c_{j,k}} = k$ , and each other job  $i, i \neq j$  and  $i \neq k$ , has value  $v_{i,c_{j,k}} = 0$ .
  - We set  $s_{j,c_{j,k}}^e = s_{j,c_{j,k}}^r = s_{k,c_{j,k}}^e = s_{k,c_{j,k}}^r = 0$ , that is equipping edge-characteristic  $c_{j,k}$  with  $j$  or  $k$  or removing  $j$  or  $k$  from  $c_{j,k}$  takes no time, if  $\{j, k\} \in E$ . If  $\{j, k\} \notin E$ , we set  $s_{j,c_{j,k}}^e = s_{j,c_{j,k}}^r = s_{k,c_{j,k}}^e = s_{k,c_{j,k}}^r = 1$ . Finally, we set  $s_{0,c_{j,k}}^e = s_{0,c_{j,k}}^r = 0$ .
- For each vertex  $i \in V$ , we introduce a characteristic  $c_i$  in  $I'$ . In the following, these characteristics are called *vertex-characteristics*.
  - The set of values for vertex-characteristics  $c_i$  is  $V_i = \{i, 0\}$ .
  - For vertex-characteristic  $c_i$ , job  $i$  receives value  $v_{i,c_i} = i$ , whereas each other job  $j, j \neq i$ , has value  $v_{j,c_i} = 0$ .
  - We set  $s_{0,c_i}^e = s_{0,c_i}^r = 0$ , that is equipping vertex-characteristic  $c_i$  with value 0 or removing value 0 from  $c_i$  takes no time. Furthermore, we set  $s_{i,c_i}^e = s_{i,c_i}^r = 1$ .

Thus, we have  $|V|$  jobs and  $0.5 \cdot |V| \cdot (|V| - 1) + |V|$  characteristics each having at most 3 values. The reduction is pseudo-polynomial.

The question we ask is whether there is a schedule with a makespan of at most  $|V| + 1$ . For each job  $i$ , we have at least a setup time of  $s_{i,c_i}^e = 1$  for equipping characteristic  $c_i$  with value  $i$  and, for the last job  $i'$ , we have at least a setup time of  $s_{i',c_{i'}}^r = 1$  for removing  $i'$  from  $c_{i'}$ . Thus,  $|V| + 1$  is a lower bound for the makespan in  $I'$  due to the vertex-characteristics.

We can achieve a makespan of  $|V| + 1$  only if we do not have two consecutive jobs  $j$  and  $k$ , such that  $\{j, k\} \notin E$ . These jobs would incur a total setup time of  $s_{j,c_{j,k}}^r + s_{k,c_{j,k}}^e = 2$  in between. However, for a pair of consecutive jobs  $j$  and  $k$ , such that the corresponding nodes are adjacent in  $I$ , we have a total setup time of 0 for each edge-characteristic. This completes the proof.  $\square$

Summarizing, similar as for MODSETUP- SEQ problem MODSETUP- PAR is strongly NP-hard, even if either  $m$  or  $K$  are fixed. On the other hand, MODSETUP- PAR can be solved in polynomial time if both  $m$  and  $K$  are fixed, see Corollary 1, or setup times are not value-dependent, see Theorem 3.

### 4 Conclusions and outlook

This paper investigates the well-known machine scheduling problem  $[1|s_{ij}|C_{\max}]$  for the special case of modular products. In a mass-customization environment, modular products allow customers to individually specify their products by selecting a specific value (e.g., a specific PU foam) for different product characteristics (e.g., the insulation of a garage door) and each value is realized by a dedicated component assembled into the product. If each component requires a specific resource (e.g., the cartridge containing the selected PU foam) during the assembly process, which need not be changed whenever two subsequent jobs share the respective value of the product characteristic, then the setup matrix becomes specially structured and depends on the similarity of products. We provide an in-depth analysis of computational complexity for different problem cases with modular products. Specifically, we distinguish whether resource setups are executed sequentially or in parallel and consider removal operations as well as special setup times. We show that for some cases with parallel setups, our detailed view on the special structure of setups is worthwhile and provides an algorithm finding optimal solutions in polynomial time. Other problem cases are shown to be strongly NP-hard, so that the complexity status compared to  $[1|s_{ij}|C_{\max}]$  remains unaltered. Note that the producer of sectional garage doors that brought the peculiarities of modular setups to our attention applies a single worker for setups, so that the

sequential setups render their machine scheduling problem strongly NP-hard.

Future research should investigate the leftover problem case, whose complexity status is still open. This case considers parallel resource setups, no removal operations  $s_{v,c}^r = 0$ , and general setup times  $s_{v,c}^e \geq 0$ . Furthermore, developing exact and heuristic solution procedures for the strongly NP-hard problem cases is a valid task for future research. Such procedures should exploit the special structure of the setup matrices and could, for instance, apply our efficiently solvable problem case as a bounding argument.

**Acknowledgements** The first author has been supported by the German Science Foundation (DFG) through the grant “State Dependent Maintenance Scheduling” (BR 3873/14-1).

**Funding** Open Access funding enabled and organized by Projekt DEAL.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Allahverdi, A. (2015). The third comprehensive survey on scheduling problems with setup times/costs. *European Journal of Operational Research*, 246(2), 345–378.
- Allahverdi, A., Gupta, J. N., & Aldowaisan, T. (1999). A review of scheduling research involving setup considerations. *Omega*, 27(2), 219–239.
- Allahverdi, A., Ng, C., Cheng, T. E., & Kovalyov, M. Y. (2008). A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*, 187(3), 985–1032.
- Bai, J., Li, Z.-R., & Huang, X. (2012). Single-machine group scheduling with general deterioration and learning effects. *Applied Mathematical Modelling*, 36(3), 1267–1274.
- Boysen, N., & Stephan, K. (2016). A survey on single crane scheduling in automated storage/retrieval systems. *European Journal of Operational Research*, 254(3), 691–704.
- Burkard, R. E., Deineko, V. G., van Dal, R., van der Veen, J. A., & Woeginger, G. J. (1998). Well-solvable special cases of the traveling salesman problem: A survey. *SIAM Review*, 40(3), 496–546.
- Cheng, T. E., Lee, W.-C., & Wu, C.-C. (2010). Scheduling problems with deteriorating jobs and learning effects including proportional setup times. *Computers & Industrial Engineering*, 58(2), 326–331.
- Da Silveira, G., Borenstein, D., & Fogliatto, F. S. (2001). Mass customization: Literature review and research directions. *International Journal of Production Economics*, 72(1), 1–13.
- Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. New York: W. H. Freeman and Company.
- Graham, R. L., Lawler, E. L., Lenstra, J. K., & Kan, A. R. (1979). Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, 5, 287–326.
- Huang, X., Wang, M.-Z., & Wang, J.-B. (2011). Single-machine group scheduling with both learning effects and deteriorating jobs. *Computers & Industrial Engineering*, 60(4), 750–754.
- Huang, X., Li, G., Huo, Y., & Ji, P. (2013). Single machine scheduling with general time-dependent deterioration, position-dependent learning and past-sequence-dependent setup times. *Optimization Letters*, 7(8), 1793–1804.
- Koulamas, C., & Kyparisis, G. J. (2008). Single-machine scheduling problems with past-sequence-dependent setup times. *European Journal of Operational Research*, 187(3), 1045–1049.
- Lai, P.-J., Lee, W.-C., & Chen, H.-H. (2011). Scheduling with deteriorating jobs and past-sequence-dependent setup times. *The International Journal of Advanced Manufacturing Technology*, 54(5–8), 737–741.
- Lee, W.-C., & Wu, C.-C. (2010). A note on optimal policies for two group scheduling problems with deteriorating setup and processing times. *Computers & Industrial Engineering*, 58(4), 646–650.
- Oda, Y., & Ota, K. (2001). Algorithmic aspects of pyramidal tours with restricted jump-backs. *Interdisciplinary Information Sciences*, 7(1), 123–133.
- Punnim, N., Saenpholphat, V., & Thaitae, S. (2007). Almost Hamiltonian cubic graphs. *International Journal of Computer Science and Network Security*, 7(1), 83–86.
- Soroush, H. (2012). Solving the single machine scheduling problem with general job-dependent past-sequence-dependent setup times and learning effects. *European Journal of Industrial Engineering*, 6(5), 596–628.
- Stefansdottir, B., Grunow, M., & Akkerman, R. (2017). Classifying and modeling setups and cleanings in lot sizing and scheduling. *European Journal of Operational Research*, 261(3), 849–865.
- Stephan, K., & Boysen, N. (2017). Crane scheduling in railway yards: An analysis of computational complexity. *Journal of Scheduling*, 20(5), 507–526.
- Svensson, O., Tarnawski, J., & Végh, L. A. (2017). A constant-factor approximation algorithm for the asymmetric traveling salesman problem. [arXiv:1708.04215](https://arxiv.org/abs/1708.04215).
- Svensson, O., Tarnawski, J., & Végh, L. A. (2018). A constant-factor approximation algorithm for the asymmetric traveling salesman problem. In *Proceedings of the 50th annual ACM SIGACT symposium on theory of computing* (pp. 204–213). New York, NY, USA, 2018. Association for Computing Machinery.
- Traub, V. Vygen, J. (2020). An improved approximation algorithm for ATSP. In *Proceedings of the 52nd annual ACM SIGACT symposium on theory of computing*, STOC 2020, New York, NY, USA, 2020. Association for Computing Machinery.
- Ulrich, K. (1995). The role of product architecture in the manufacturing firm. *Research Policy*, 24(3), 419–440.
- Wang, J.-B., Lin, L., & Shan, F. (2008). Single-machine group scheduling problems with deteriorating jobs. *The International Journal of Advanced Manufacturing Technology*, 39(7–8), 808–812.
- Wang, J.-B., Huang, X., Wu, Y.-B., & Ji, P. (2012). Group scheduling with independent setup times, ready times, and deteriorating job processing times. *The International Journal of Advanced Manufacturing Technology*, 60(5–8), 643–649.
- Wu, C.-C., & Lee, W.-C. (2008). Single-machine group-scheduling problems with deteriorating setup times and job-processing times. *International Journal of Production Economics*, 115(1), 128–133.

- Xu, Y.-T., Zhang, Y., & Huang, X. (2014). Single-machine ready times scheduling with group technology and proportional linear deterioration. *Applied Mathematical Modelling*, 38(1), 384–391.
- Yang, S.-J. (2011). Group scheduling problems with simultaneous considerations of learning and deterioration effects on a single-machine. *Applied Mathematical Modelling*, 35(8), 4008–4016.
- Yang, S.-J., & Yang, D.-L. (2010). Single-machine group scheduling problems under the effects of deterioration and learning. *Computers & Industrial Engineering*, 58(4), 754–758.
- Yin, N., Wang, J.-B., Wang, D., Wang, L.-Y., & Wang, X.-Y. (2010). Deteriorating jobs and learning effects on a single-machine scheduling with past-sequence-dependent setup times. *The International Journal of Advanced Manufacturing Technology*, 46(5–8), 707–714.
- Yin, Y., Xu, D., Cheng, S.-R., & Wu, C.-C. (2012). A generalisation model of learning and deteriorating effects on a single-machine scheduling with past-sequence-dependent setup times. *International Journal of Computer Integrated Manufacturing*, 25(9), 804–813.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.