Journal of Scheduling (2021) 24:123–135
https://doi.org/10.1007/s10951-020-00645-8

Cyclic lot-sizing problems with sequencing costs

Alexander Grigoriev[1] · Vincent J. Kreuzen[1] · Tim Oosterwijk[1]

Published online: 21 February 2020

Abstract
We study a single-machine lot-sizing problem, where n types of products need to be scheduled on the machine. Each product is associated with a constant demand rate, maximum production rate and inventory costs per time unit. Every time when the machine switches production between products, sequencing costs are incurred. These sequencing costs depend both on the product the machine just produced and on the product the machine is about to produce. The goal is to find a cyclic schedule minimizing total average costs, subject to the condition that all demands are satisfied. We establish the complexity of the problem, and we prove a number of structural properties largely characterizing optimal solutions. Moreover, we present two algorithms approximating the optimal schedules by augmenting the problem input. Due to the high-multiplicity setting, even trivial cases of the corresponding conventional counterparts become highly non-trivial with respect to the output sizes and computational complexity, even without sequencing costs. In particular, the length of an optimal solution can be exponential in the input size of the problem. Nevertheless, our approximation algorithms produce schedules of a polynomial length and with a good quality compared to the optimal schedules of exponential length.

Keywords Lot-sizing problem · Sequencing costs · High multiplicity · Approximation algorithm

1 Introduction

In the current competitive economy, companies need to be aware of multiple objectives such as decreasing costs and enhancing customer service. Among the core activities of many companies and supply chains are mechanisms to match supply with demand, to prevent stock-outs and to cut back unnecessary overhead costs. Production companies are required to conduct extensive research into cost reduction to remain competitive within the market. Consequently, a lot of interest has been shown in problems within the area of operations management.

This paper is motivated by a real-life problem: A multinational textile company posed the problem of optimizing the production schedule of their lycra production operations. The company employed a single machine to produce synthetic fibers of a few different types of thickness, subject to (extremely large) fixed daily output rates. In the setting we have dealt with, there were three to five types of lycra thickness. Every switch from one thickness type to another is associated with a setup of the machine, and corresponding costs occur. The company was interested in finding a cyclic production schedule of minimum *cycle length*. A similar setting can be encountered in the automotive manufacturing, where the cars on the conveyor belt should be colored. Due to cleansing requirements, changing from one color to another does not only take a fixed amount of setup time, but an additional amount of time based on the color sequence, e.g., switching from black to yellow is more costly than switching from yellow to green.

In the aforementioned industrial setting, the problem of finding an optimal cycle in which every product is produced exactly once has been addressed in Wetsels (2012). In the present paper, we generalize this result to cyclic schedules with no restrictions on the number of production periods per product. We arrive at a lot-sizing problem with sequence-dependent setup costs and aim to find a cyclic schedule which minimizes total *average* costs.

The LOT-SIZING PROBLEM is a well-studied problem in operations management, where one machine needs to produce a set of products to minimize average holding and

✉ Tim Oosterwijk
t.oosterwijk@maastrichtuniversity.nl

Alexander Grigoriev
a.grigoriev@maastrichtuniversity.nl

Vincent J. Kreuzen
vjc.kreuzen@avans.nl

[1] Maastricht University, Maastricht, The Netherlands

setup costs. In this problem, the ongoing production can be represented as the repeated scheduling of a single job on the machine, enabling a highly compact encoding of the input. These types of problems are commonly referred to as *high-multiplicity scheduling* problems. Jobs in the high-multiplicity setting are represented by a single job description with a multiplicity, representing the number of individual jobs to be processed. It is different from the conventional scheduling setting where every single job, even though identical to many others, is given as a part of the problem input. In this case, the input length of the traditional setting can be exponentially larger than the length of the high-multiplicity input, resulting in exponentially slower performance of algorithms regularly applicable in traditional scheduling. Due to the compact encoding of the input for the problem at hand, the optimal schedule can have superpolynomial length, even for very restricted cases with only one or two products (see Gabay et al. 2014). Consequently, finding a polynomially sized certificate for these types of problems alone can already prove to be a hard task.

Not only from the computational complexity point of view, it is questionable whether conventional encoding is practical for high-multiplicity problems. For many companies, the high-multiplicity encoding is a natural way to provide input from real-world data, especially if thousands of jobs are identical, and is found in numerous practical applications. In this age of big data, in which processing large amounts of data is becoming more and more important, and in many cases necessary in order to keep up with competitors, companies are often able to compress large amounts of data into smaller-sized input sets. Algorithms need to be equipped to cope with these compressed data in such a way that the original problem is tackled, without resorting to the usage of excessive processing times in order to process the underlying information of the reduced input.

In this paper, we address this issue by incorporating high-multiplicity encoding into an extended version of the aforementioned real-life problem, the LOT- SIZING PROBLEM with sequence-dependent setup costs. In this problem, we have a single machine that is capable of producing a single product at any given time and a set of products that need to be produced. Each product is associated with a demand rate, a maximum production rate and inventory holding costs per unit. The objective is to find a cyclic schedule such that the demand of every product is met, minimizing the average costs per cycle. For any schedule, sequence-dependent setup costs, referred to as *sequencing costs*, are incurred each time the machine switches production between two different products. Moreover, input is provided under high-multiplicity encoding.

We show NP-hardness of the problem, and we largely characterize optimal solutions by proving a number of structural properties, which will be of great use for the algorithm design. Further, we develop an approximation algorithm which slightly perturbs the input instance to get a polynomial running time and, most importantly, polynomial size of the output schedule, where the number of products is fixed. The latter is a reasonable assumption, since in most real-world applications, the number of distinct product types is relatively small, while the demand quantities are substantial. The quality of the resulting schedule is relatively close to that of an optimal schedule.

2 Related work

The earliest research on problems with high-multiplicity encoding dates back to the 1960s; see e.g., Rothkopf (1966) who considers the TRAVELING SALESMAN PROBLEM with multiple visits to cities. Madigan (1968) studies a variant of our problem where setup times are introduced, setup costs do not depend on the sequence, and holding costs are product-independent. He proposes an elegant heuristic for the problem and compares it to the results previously published in the literature. Goyal (1973) studies the variant of the problem posed by Madigan where no setup times are involved and solves the problem to optimality for a fixed time horizon. Boctor (1982) extends the model to incorporate product-dependent holding costs and setup times and considers an infinite time horizon. He presents an exact algorithm for the case of two products. For a historic overview of economic lot-sizing problems, we refer to Holmbom and Segerstedt (2014).

Only in 1991, Hochbaum and Shamir (1991) coined the term *high multiplicity* and underlined the added complexity of such encodings. They study single-machine high-multiplicity scheduling problems with different objective functions and construct algorithms that are strongly polynomial in the number of types of jobs. At the same time, Narro Lopez and Kingsman (1991) discuss basic solution approaches to high-multiplicity scheduling problems and assess their quality and use in practice.

Most papers on high-multiplicity scheduling consider discrete variants, in which time and/or quantities are discretized into units. There has also been some work considering the continuous setting, in which production can start and stop at any time, e.g., with fluids. Bertsimas et al. (2003) consider the high-multiplicity job shop problem without sequencing costs and use this continuous setting as a relaxation for the original discrete job shop problem. They round an optimal solution for the fluid problem to an asymptotically optimal solution for the discrete problem and provide some computational experiments. In another work on the continuous setting, Haase (1996) discusses a problem very closely related to ours, where production rates are fixed. He proposes a local search heuristic and evaluates it by comparing it to optimal solutions for small instances. Haase and Kimms (2000) consider the same problem and, by making additional assumptions on the input instances, solve the problem to optimality. They present a mixed integer programming formulation for their model and a fast enumeration scheme, which they evaluate by a computational study.

Incorporating sequencing costs substantially adds complexity akin to the TRAVELING SALESMAN PROBLEM. The techniques we are using in this paper are closely related to the techniques used in classical single multiplicity scheduling. For instance, Clifford and Posner (2000) provide lower bounds and use these to develop heuristics for minimizing tardiness. They extend the problem to parallel, uniform and unrelated machines in Clifford and Posner (2001), where their objective is to minimize the makespan or the sum of completion times in either the preemptive, or the non-preemptive variant of the problem. They prove NP-hardness, develop polynomial time and pseudopolynomial time algorithms for special cases, and present heuristics. Filippi and Romanin-Jacur (2009) continue their work and present a two-stage approach, in which they first fix most jobs in partial schedules and then solve the residual problem.

Brauner et al. (2005) provide a detailed framework for the complexity analysis of high-multiplicity scheduling problems. We refer the reader to this paper for an excellent survey of related work in this field. They extend their framework in Brauner et al. (2007).

3 The model

We model the general problem for multiple products as follows. We have a single machine that can produce a single type of product at any given time and we are given a set of products $J = \{1, \dots, n\}$. For each product $i \in J$, let p_i be its maximum production rate, i.e., the maximum number of units produced per time unit. Similarly, let d_i be its demand rate and h_i its holding costs per time unit. Furthermore, we are given sequencing costs $s_{i,j}$ that need to be paid when the machine switches from producing product i to producing product j. The problem we consider is to find an optimal cyclic schedule S^* that minimizes the average costs per unit of time $\bar{c}(S^*)$. Note that for each product i, the rates d_i and p_i and costs h_i are assumed to be constant over time and positive. Observe that the input is very compact. Let m be the largest number in the input, then the input size is $\mathcal{O}(n \log m)$, where n is typically a small number, or even a constant.

We distinguish two variants of the problem: The *continuous* case, in which the machine can switch production at any time; and the *discrete* case, in which the machine can switch production only at the end of a fixed unit of time (e.g., a day) and produces some product i at a single rate $r_i \leq p_i$ during each unit of time. Herewith, we assume that production is done in the beginning of the period and demand is satisfied at the end. Without loss of generality, in both variants we assume $d_i, p_i, h_i, s_{ij} \geq 1$ for all $i, j \in J$.

We denote by LSP(A,n) with $A \in \{C, D\}$, $n \in \mathbb{N}$ the lot-sizing problem of scheduling n products in the continuous or discrete setting, respectively. Let $\pi_i^{[a,b)}$ denote the produced amount of product i during time interval $[a, b)$. Let $\pi_i^t = \pi_i^{[t,t+1)}$. Let x_i^t be an indicator function denoting whether product i is produced during time interval $[t, t+1)$. Let q_i^t denote the stock level for product i at time t. We explicitly refer to the stock of product i at time t in a schedule S as $q_i^t(S)$.

Finally, let $H(S)$ denote the total holding costs and $W(S)$ the total sequencing costs of a schedule S, and $c(S) = H(S) + W(S)$ denote the total costs of S. Denote the average costs of a cyclic schedule S of cycle length ℓ by $\bar{c}(S) = \bar{H}(S) + \bar{W}(S)$, where $\bar{H}(S) = H(S)/\ell$ and $\bar{W}(S) = W(S)/\ell$.

Formally, we arrive at the following problem.

Input Let $A \in \{C, D\}$. Let a set of products $J = \{1, \dots, n\}$ be given, and for each product $i \in J$, a demand rate $d_i \geq 1$, a maximum production rate $p_i \geq 1$, and inventory holding costs $h_i \geq 1$. Sequencing costs $s_{i,j} \geq 1$ are given for every pair of products.

Task Find a cyclic schedule S which minimizes the average costs per unit of time, $\bar{c}(S)$ for A.

We represent a cyclic schedule of length ℓ as a sequence:

$$\left[t_0, t_1\right)_{i_0}^{r_0}, [t_1, t_2)_{i_1}^{r_1}, \dots, [t_s, \ell)_{i_s}^{r_s},$$

where $r_\varphi \leq p_{i_\varphi}$ is a production rate of *phase* $\varphi = 0, \dots, s$, i_φ is the product produced in that phase, and $[t_\varphi, t_{\varphi+1})$ is a maximal time interval where only i_φ is produced at a fixed rate r_φ. A maximal sequence of consecutive phases of the same product $i \in J$ is called a *production period*, denoted by $[t, t')_i$ for some $t' > t$. The complete sequence of phases is called the (cyclic) schedule, and we call a schedule a *simple cycle* if there is exactly one production period for each product.

4 Structural properties of optimal solutions

We now prove some structural properties of optimal schedules of the problem. We show that all variants are NP-hard, even when we restrict ourselves to unit demand rates and unit holding costs. Next, we derive a simple necessary and sufficient condition for the existence of a feasible cyclic schedule. Furthermore, we characterize the form of production for the continuous and discrete cases. Also, we show that there is no idle time in an optimal schedule and that every product has at least one point during the schedule where its stock level is zero. Finally, in the last subsection, we present a lower bound on the objective value for the continuous case and an upper bound on the objective value and the maximum stock level for the discrete case. We use these bounds in the approximation.

4.1 Problem complexity

The following lemma follows directly from a reduction from the TRAVELING SALESMAN PROBLEM (TSP).

Lemma 1 (Complexity) *Both the discrete and the continuous variants of the lot-sizing problem are strongly NP-hard.*

Proof We prove NP-hardness for the discrete case by a reduction from the TRAVELING SALESMAN PROBLEM (TSP). Let us consider a TSP instance $I = \{G = (V, E), [c_{i,j}]_{V \times V}\}$. From this given TSP instance, we construct an instance $I' = \{J, (d_i, p_i, h_i)_{i \in J}, [s_{i,j}]_{J \times J}\}$ of the LOT- SIZING PROBLEM as follows.

Identify J with V and let $s_{i,j} = c_{i,j}$ for each $i, j \in J$. Let $d_i = 1$, $p_i = |V| = n$, and $h_i = h = \sum_{j,k} s_{j,k} + 1$ for each $i \in J$. Additionally, let $W_{\max} = \sum_{i,j} s_{i,j}$ and $W_{\min} = n \times \min_{i,j} s_{i,j}$.

Note that for every feasible schedule S, we have sequencing costs $W(S)$ such that $W(S) \geq W_{\min}$. Moreover, for all simple cycles S we have $W(S) \leq W_{\max}$. We claim that there exists a TSP tour of length at most B if and only if the corresponding instance of the lot-sizing problem admits a solution of total cost at most $hn(n-1)/2 + B/n$.

Clearly, since the total demand and production rates match each other, the total stock level is constant over time. Every simple cycle of length n, using the same order of products, can be realized with average holding costs $\bar{H} = hn(n-1)/2$ and average sequencing costs $W_{\min}/n \leq \bar{W} \leq W_{\max}/n$. In fact, this schedule is minimum regarding the holding costs.

Let S' be a feasible non-simple cycle of length $\ell' > n$ with total costs $c(S') = H(S') + W(S')$. Note that there is a product of which two consecutive production periods are separated by at least $n + 1$ time units. Hence, we need at least one additional unit of that product in stock and thus $H(S') \geq h\ell' n(n-1)/2 + h\ell'$. Thus, for every minimal simple cycle S, since $W(S) \leq W_{\max} < h$, we have that the average costs of S' are $\bar{c}(S') \geq H(S')/\ell' > \bar{c}(S)$. Observe that the value of $\bar{H}(S)$ is the same for every minimal simple cycle, and therefore the optimal solution to I is the minimal simple cycle which minimizes $W(S)$.

Let σ be a sequence of visits in the TSP instance with costs B. Producing each product for 1 time unit with the same sequence as σ is a feasible solution for the lot-sizing problem with costs $hn(n-1)/2 + B/n$. Conversely, let σ be a solution for the lot-sizing problem with costs $hn(n-1)/2 + B/n$. This solution is a simple cycle, and therefore the production sequence is a tour with costs B. This proves the NP-hardness of the discrete case.

We prove the continuous case by a similar reduction from the METRIC TSP. For an instance I of the METRIC TSP, we let $J = V$ and $s_{i,j} = c_{i,j}$ for all $i, j \in J$. Let $d_i = 1$, $p_i = n$ and $h_i = 1$ for all $i \in J$.

Let σ be an optimal solution to I with costs $c(\sigma)$. Let S be any feasible schedule for the corresponding instance I' of the lot-sizing problem, and let the length of the schedule be ℓ. Let S^* be the simple cycle of length ℓ^* where the products are produced in the same order as in σ, with production time ℓ^*/n per product.

Since every product needs to be produced at least once in a feasible schedule and the triangle inequality holds for the sequencing costs, S^* is optimal with respect to the sequencing costs, i.e., $W(S^*) \leq W(S)$. Note that compared to the discrete case, the continuous case has a complication: We can choose ℓ^* arbitrarily small. By construction, every production period in schedule S^* consists of one phase of length ℓ^*/n where the product is produced at rate $p_i = n$. Since $h_i = 1$, the total holding costs for every product i are given as (cf. Fig. 1)

$$\int_0^{\ell^*/n} q_i^t \mathrm{d}t + \int_{\ell^*/n}^{\ell^*} q_i^t \mathrm{d}t = \frac{n-1}{2n}(\ell^*)^2. \tag{1}$$

Thus, the total holding costs of S^* are $H(S^*) = (\ell^*)^2(n-1)/2$ and the average holding costs are $\bar{H}(S^*) = \ell^*(n-1)/2$. In particular, since holding costs decrease with the cycle length, we can choose ℓ^* such that $\bar{H}(S^*) \leq \bar{H}(S)$ and $\bar{c}(S^*) \leq \bar{c}(S)$. Thus, we have that the optimal solution to I' is a simple cycle S^* using the sequence of σ.

The total average costs $c(\sigma)/\ell^* + \ell^*(n-1)/2$ are minimized with $\ell^* = \sqrt{2c(\sigma)/(n-1)}$. Hence, we have

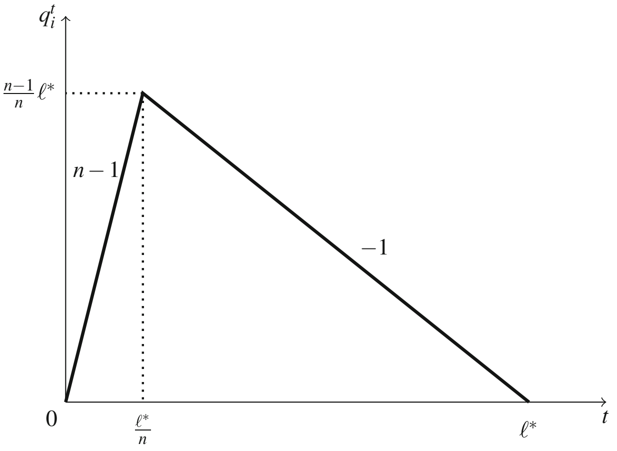

Fig. 1 An illustration to clarify Eq. (1)

$$c(\sigma) = W(S^*) = \frac{n-1}{2}(\ell^*)^2 = H(S^*).$$

Now, σ is an optimal solution for I with costs $c(\sigma)$, if and only if there is an optimal solution for I' with average costs $\sqrt{2(n-1)c(\sigma)}$. □

A closely related problem with setup times was addressed in Gallego and Shaw (1997), where they show NP-hardness for multiple special cases of their problem.

4.2 Feasibility condition

Observe that d_i/p_i is the fraction of time product i needs to be scheduled on the machine, and therefore $\sum_{i\in J} d_i/p_i$ needs to be at most 1. The following lemma shows that this is a sufficient condition for feasible schedules.

Lemma 2 [Feasibility, Gabay et al. (2014)] *For both variants of the problem, there exists a feasible schedule if and only if*

$$\sum_{i\in J}\frac{d_i}{p_i} \leq 1.$$

Remark 1 Note that additionally for LSP(F,n), a schedule of length ℓ is feasible if and only if

$$\frac{\ell d_i}{p_i} = \sum_{t=0}^{\ell-1} x_i^t \in \mathbb{N}, \quad \text{and} \quad \ell \bmod (p_i - d_i) = 0, \quad \forall_{i\in J}.$$

4.3 Characterizing optimal production schedules

In this subsection, we prove several properties about the production in continuous and discrete schedules. We start by showing that if there is some idle time in a schedule, we can already start producing the next product at demand rate during the idle time to decrease the holding costs.

Lemma 3 [No idle time, Gabay et al. (2014)] *Let S^* be an optimal schedule for LSP(C,n) or LSP(D,n), with $n \in \mathbb{N}$. S^* has no idle time.*

We now provide a short proof for the claim that in an optimal schedule for the continuous case, at any time the production rate is always larger than or equal to the demand rate of the produced product.

Lemma 4 (Produce at least the demand rate) *Let S^* be an optimal schedule for LSP(C,n) with $n \in \mathbb{N}$. For each phase $[t,t')_i^r$ in S^*, we have that $r \geq d_i$.*

Proof We prove by contradiction. Let S be a counterexample, i.e., there is at least one phase $[t,t')_i^r$ with $r < d_i$. Since S is feasible, we know that $q_i^t \geq (d_i - r)(t'-t) > 0$. Now, let $\pi_i^{[t',\ell)\cup[0,t)} \leftarrow \pi_i^{[t',\ell)\cup[0,t)} - (d_i - r)(t-t')$ and replace $[t,t')_i^r$ by $[t,t')_i^{d_i}$. Clearly, the schedule is feasible and the costs are decreased, and thus S was not optimal. □

The next property ensures that the machine produces every product i only at rates d_i and p_i to minimize holding costs in the continuous case.

Lemma 5 [Two-phase production, Gabay et al. (2014)] *Consider LSP(C,n) for any $n \geq 2$. There is an optimal cycle S^* such that for every product $i \in J$, every production period of i in S^* consists of at most two phases. For every production period, in the first phase the machine produces i at a rate of d_i. During the second (non-empty) phase i is produced at a rate of p_i.*

Note that in a *tight* schedule, i.e., $\sum_{i\in J} d_i/p_i = 1$, in order to meet demand for each product, the machine needs to continuously produce at maximum speed. Therefore, in an optimal schedule S for a tight instance of the problem, each production period consists of a single phase where product i is produced at rate p_i. Furthermore, the proof of Lemma 5 also proves that in an optimal schedule for LSP(C,n), for each phase $[t,t')_i^r$, we have that $r = d_i$ or $r = p_i$.

Following the same reasoning as in the previous two lemmata, we can achieve a similar result for the discrete case of the problem and prove that in an optimal schedule, production periods consist of at most four phases.

Lemma 6 (Four-phase production) *Consider LSP(D,n) for any $n \geq 2$. There is an optimal cycle S^* such that for every product $i \in J$, every production period of i in S^* consists of at most four phases. For every production period, in the first phase the machine produces i at a rate of $r_1 < d_i$ and this phase has length at most 1. During the second phase i is produced at a rate of d_i. During the third phase, i is produced at rate $d_i < r_2 < p_i$ and this phase again has length at most 1. Finally, during the fourth phase, i is produced at a rate of p_i. Phases can be empty, but the first and third phase cannot occur sequentially.*

Proof We prove by contradiction. We claim, following arguments similar to those in the proofs of the previous two lemmata, that phases within the production period can be ordered such that for every pair of phases $[t_j, t_{j+1})_i^{r_1}, [t_{j'}, t_{j'+1})_i^{r_2}$ with $j' > j$ we have that $r_1 < r_2$ in order to minimize costs while retaining a feasible schedule. To see this, note that a swap similar to the swap in the proof of Lemma 4 yields lower holding costs, as it is always favorable to produce demand at the latest possible time.

Suppose we have an optimal schedule S with two consecutive phases $[t_j, t_{j+1})_i^{r_j}, [t_{j+1}, t_{j+2})_i^{r_{j+1}}$. By definition of a phase, $r_j \neq r_{j+1}$. Since S is optimal, $0 < r_j < r_{j+1} \leq p_i$ must hold. Clearly, if $t_{j+2} = t_{j+1} + 1 = t_j + 2$, the lemma holds. Otherwise, we construct a new schedule S^* and we start this construction by initializing $S^* := S$.

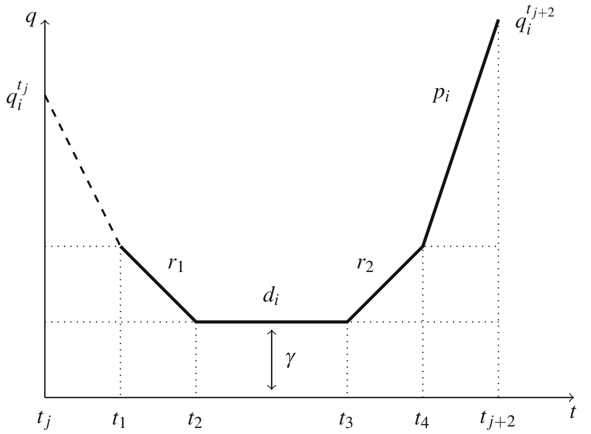

Fig. 2 A depiction of an optimal production period of schedule S^* for LSP(C,n) (where phases $[t_1, t_2)_i^{r_1}$ and $[t_3, t_4)_i^{r_2}$ must be empty) and for LSP(D,n), with $n \geq 2$

We split $[t_j, t_{j+1})_i^{r_j}, [t_{j+1}, t_{j+2})_i^{r_{j+1}}$ in S^* into five new phases as follows. We first deplete the stock by q° and consecutively increase the stock by q^*, where these values depend on the case distinction below. We introduce the indicator function $f_{\mathbb{N}}(x) = \lceil x \rceil - \lfloor x \rfloor$ which takes on the value 1 if $x \notin \mathbb{N}$ and 0 otherwise. The new phases are:

$$[t_j, t_1)_i^0, [t_1, t_2)_i^{r_1}, [t_2, t_3)_i^{d_i}, [t_3, t_4)_i^{r_2}, [t_4, t_{j+2})_i^{p_i},$$

$$\text{where } t_1 = t_j + \left\lfloor \frac{q^\circ}{d_i} \right\rfloor, \text{ and } t_2 = t_1 + f_{\mathbb{N}}\left(\frac{q^\circ}{d_i}\right),$$

$$t_4 = t_{j+2} - \left\lfloor \frac{q^*}{p_i - d_i} \right\rfloor, \text{ and } t_3 = t_4 - f_{\mathbb{N}}\left(\frac{q^*}{p_i - d_i}\right),$$

$$r_1 = d_i - \left(q_i^{t_j} - (t_1 - t_j)d_i\right)$$

$$\text{and } r_2 = d_i + \left(q_i^{t_{j+2}} - (t_{j+2} - t_4)(p_i - d_i)\right).$$

We refer the reader to Fig. 2 for a depiction of the new set of phases.

Firstly, suppose $d_i \leq r_j < r_{j+1}$. Now, let $q^* = (q_i^{t_{j+2}} - q_i^{t_j})$ and $q^\circ = 0$, consequently producing stock, which results in a production period of at most three phases.

Secondly, suppose $r_j < r_{j+1} \leq d_i$. Now, let $q^\circ = (q_i^{t_j} - q_i^{t_{j+2}})$ and $q^* = 0$, consequently depleting stock, which results in a production period of at most three phases.

Lastly, suppose $r_j < d_i < r_{j+1}$. Now, let $q^\circ = q_i^{t_j}$ and $q^* = q_i^{t_{j+2}}$, consequently first depleting and consecutively producing stock, which results in a production period of at most four phases.

If completely depleting and consecutively producing the stock takes longer than the production period, we get $t_2 > t_3$. In this case, denote the total amount of stock which was produced in this production period by $q = (t_{j+2} - t_j)d_i + (q_i^{t_{j+2}} - q_i^{t_j}) = r_j(t_{j+1} - t_j) + r_{j+1}(t_{j+2} - t_{j+1})$. Then, let $t_4 = t_{j+2} - \left\lfloor \frac{q}{p_i} \right\rfloor$ and $t_1 = t_2 = t_3 = t_{j+2} - \left\lceil \frac{q}{p_i} \right\rceil$ and $r_2 = d_i + q - p_i \left\lfloor \frac{q}{p_i} \right\rfloor$, resulting in a production period of at most three phases.

Clearly, in all cases S^* is feasible. If S^* is different from S, then $H(S^*) < H(S)$, and thus S is not optimal. Note that the phase $[t_j, t_1)_i^0$ is idle and can be removed as in the proof of Lemma 3 by extending or introducing demand production for some other product, thereby delaying its stock production, leaving a production period of four phases and proving the lemma. □

Note that the proof of Lemma 6 also shows that in an optimal schedule for LSP(D,n), for each phase $[t, t')_i^r$ with $t' > t + 1$, we have that $r = d_i$ or $r = p_i$.

We now show that in the continuous case, the machine produces product i at rate d_i only if the stock for i is empty.

Lemma 7 (Level production for continuous case) *In an optimal schedule S^* for an instance of LSP(C,n), for any product $i \in J$ there exists a non-empty phase $[t_j, t_{j+1})_i^{d_i}$ (i.e., with $t_{j+1} > t_j$) only if $q_i^{t_j} = 0$.*

Proof We prove by contradiction. Suppose we have an optimal schedule S with a phase $[t_j, t_{j+1})_i^{d_i}$ with $t_{j+1} > t_j$ and $q_i^{t_j} > 0$. Again, we construct a new schedule S^* starting with $S^* := S$. We split $[t_j, t_{j+1})_i^{d_i}$ in S^* into three new phases:

$$[t_j, t_1)_i^0, [t_1, t_2)_i^{d_i}, [t_2, t_{j+1})_i^{p_i},$$

$$\text{where } t_1 = t_j + \frac{q_i^{t_j}}{d_i} \text{ and } t_2 = t_{j+1} - \frac{q_i^{t_{j+1}}}{p_i - d_i}.$$

If the length of the phase is too short to completely deplete the stock and consecutively completely rebuild the stock, i.e., $t_1 > t_2$, then we reduce the stock as much as possible. In this case, let $t_1 = t_2 = t_{j+1} - t^\circ$, where $t^\circ = (t_{j+1} - t_j)\frac{d_i}{p_i}$ denotes the time required to produce when producing at rate p_i in order to meet demand during the original phase.

Clearly, S^* is feasible and now we have that $H(S^*) < H(S)$ and thus S is not optimal. □

We now show a similar result for the discrete case, where the machine produces product i at rate d_i only if the stock for i is empty *or* if the production phase has length 1.

Lemma 8 (Level production for discrete case) *In an optimal schedule S^* for an instance of LSP(D,n), for any product $i \in J$ there exists a non-empty phase $[t_j, t_{j+1})_i^{d_i}$ only if $q_i^{t_j} = 0$ or $t_{j+1} = t_j + 1$.*

Proof We prove by contradiction. Suppose we have an optimal schedule S with a phase $[t_j, t_{j+1})_i^{d_i}$ with $t_{j+1} = t_j + 2$ and $q_i^{t_j} > 0$. Once again, we construct a new schedule S^*

starting with $S^* := S$. We can now split $[t_j, t_{j+1})_i^{d_i}$ in S^* into two new phases:

$$[t_j, t_j + 1)_i^{r_1}, [t_j + 1, t_{j+1})_i^{r_2}.$$

If $q_i^{t_j} < d_i$, let $r_1 = d_i - q_i^{t_j}$ and $r_2 = d_i + q_i^{t_j}$. Otherwise, let $r_1 = \max\{2d_i - p_i, 0\}$ and $r_2 = \min\{p_i, 2d_i\}$. Clearly, S^* is feasible and we have that $H(S^*) < H(S)$ and thus S is not optimal.

Next, suppose $t_{j+1} > t_j + 2$. Now, split $[t_j, t_{j+1})_i^{d_i}$ into the phases $[t_j, t_j+1)_i^{r_1}[t_j+1, t_{j+1}-1)_i^{d_i}[t_{j+1}-1, t_{j+1})_i^{r_2}$, where r_1 and r_2 are defined as above. This process can be iteratively repeated upon the schedule S^* until either the stock level reaches 0, or there is at most one phase left of length 1. □

We can now show that in an optimal schedule, for every product there is a time where its stock level is zero.

Lemma 9 (Zero stock level) *Let S^* be an optimal schedule for an instance of LSP(C,n) or LSP(D,n). Then, for each $i \in J$ there exists a time t such that $q_i^t = 0$.*

Proof The proof is by contradiction. Let S be an optimal schedule of length ℓ with at least one product i such that $q_i^t > 0$ for all t. Let t^* be such that $q_i^{t^*} = \min_{0 \le t \le \ell} q_i^t$. Now, let S^* be a copy of S, where we decrease the stock level for the entire schedule of this product, i.e., $q_i^t \leftarrow q_i^t - q_i^{t^*}$ for all $0 \le t \le \ell$. Since $q_i^{t^*} \le q_i^t$ for all t in S, we know that S^* is feasible. Clearly, $H(S^*) < H(S)$, and thus S is not optimal. Note that the stock level can be decreased by producing at a rate lower than required by the schedule until the desired level is attained. □

4.4 Bounding the average costs

We conclude the basic properties with a lower bound on the average costs of an optimal continuous schedule and an upper bound on the average costs and maximum stock level of an optimal discrete schedule. To obtain these, we first derive optimality conditions for both cases.

Lemma 10 (Continuous cost balancing) *An optimal schedule S for an instance of LSP(C,n) has the property that $H(S) = W(S)$.*

Proof We prove by contradiction. Let S be an optimal schedule s.t. $H(S) \ne W(S)$. Scale the length of each phase in S by a positive factor $\delta \ne 1$, such that for the resulting feasible schedule S' it holds that $H(S') = W(S')$. Let i be a product, where without loss of generality we assume that $h_i = 1$. The holding costs for i in S during a phase $[t_1, t_2)_i^r$ are given as

$$H(i, [t_1, t_2)_i^r) = (t_2 - t_1)q_{\min} + \frac{(t_2 - t_1)^2}{2}(r - d_i) \ ,$$

where $q_{\min}$ is the minimum stock level of i during the phase. When rescaling, we maintain two inequalities. First of all, it is clear that $(t_2' - t_1') \le (t_2 - t_1)\delta$. Moreover, considering Fig. 1, we see that rescaling results in similar triangles of the stock level, implying that $q'_{\min} \le q_{\min}\delta$. Thus, for the corresponding phase $[t_1', t_2')_i^r$ of the scaled schedule S' we have

$$\begin{aligned} H(i, [t_1', t_2')_i^r) &= (t_2' - t_1')q'_{\min} + \frac{(t_2' - t_1')^2}{2}(r - d_i) \\ &\le (t_2 - t_1)\delta^2 q_{\min} + \frac{(t_2 - t_1)^2\delta^2}{2}(r - d_i) \\ &= H(i, [t_1, t_2)_i^r)\delta^2. \end{aligned}$$

Summing over all phases and products, we get

$$H(S') = \sum_{[t_1', t_2')_i^r \in S'} \sum_{i \in J} H(i, [t_1', t_2')_i^r) \le H(S)\delta^2.$$

Observe that due to scaling the schedule, we have $W(S) = W(S') = H(S') \le H(S)\delta^2$, or equivalently, $\delta \ge \sqrt{W(S)/H(S)}$. We now choose δ such that

$$\sqrt{\frac{W(S)}{H(S)}} \le \delta < \frac{1}{2} + \frac{W(S)}{2H(S)}. \tag{2}$$

Observe that for any values of $H(S)$ and $W(S)$ s.t. $H(S) \ne W(S)$, there exists a δ satisfying Eq. (2). Because of this particular choice of δ, we have that

$$\begin{aligned} \bar{c}(S') &= \frac{1}{\ell'}H(S') + \frac{1}{\ell'}W(S') = \frac{1}{\ell\delta}H(S') + \frac{1}{\ell\delta}W(S) \\ &\le \frac{\delta}{\ell}H(S) + \frac{\delta}{\ell}H(S) = \frac{2\delta}{\ell}H(S) \\ &< \frac{1}{\ell}H(S) + \frac{1}{\ell}W(S) = \bar{c}(S), \end{aligned}$$

and thus S was not optimal, proving the lemma. □

We now prove a similar result for the discrete case, taking into account that low values of δ might create infeasible schedules.

Lemma 11 (Discrete cost balancing) *A schedule S for an instance of LSP(D,n) is optimal only if $W(S) \le 4 \cdot H(S)$.*

Proof The lemma follows almost entirely from the proof of Lemma 10. The difference is that in the discrete case, we might introduce infeasible schedules S' by stretching with any factor δ. Therefore, we restrict ourselves to factors $\delta \in \mathbb{N}$, $\delta \ge 2$. The first inequality in Eq. (2) now holds only if $W(S) \le 4 \cdot H(S)$. □

To obtain a lower bound on the average costs, we first fully characterize the optimal continuous schedule for instances where all products are identical.

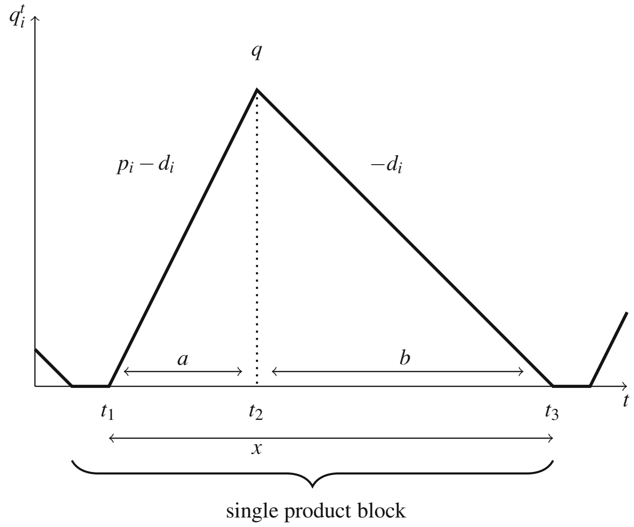

Fig. 3 An example of a single minimum *product block*

Lemma 12 (Identical products) *For LSP(C,n) with n identical products, i.e., $d_i = d$, $p_i = p$, $h_i = h$ and $s_{i,j} = s$ for all $i, j \in J$, the optimal schedule S^* is a simple cycle of average costs $\bar{c}(S^*)$ and length ℓ, where*

$$\bar{c}(S^*) = n\alpha\sqrt{\frac{2s(p-d)dh}{p}} \text{ and } \ell = \frac{1}{\alpha}\sqrt{\frac{2sp}{(p-d)dh}},$$
$$\text{where } \alpha = \left(1 - \frac{1}{n} + \frac{d}{p}\right).$$

Proof Since all products are identical, the optimal schedule is defined by a simple cycle where all products are produced for the same period of time and $H(S) = W(S)$; see Lemma 10.

We first look at a single *product block* $[t_1, t_2, t_3)_i$, which denotes the period for a single item i from the moment it starts a production period, until it starts another production period. Here, $[t_1, t_2)$ denotes the production period for product i and $[t_2, t_3)$ denotes the period during which i is not produced. Note that $q_i^{t_1} = q_i^{t_3} = 0$. See Fig. 3.

The holding costs for a single product i are given as $\frac{xq}{2}h$, where q is the maximum stock level and x is the time where product i is produced at rate p plus the time it is not produced, during the product block. Given a slope of $p - d$ during production and a slope of $-d$ during non-production, since $q = a(p-d) = bd$, we have $a = \frac{dx}{p}$, $b = x - a$, resulting in total holding costs of

$$\frac{xq}{2}h = x\frac{a(p-d)}{2}h = x^2\frac{(p-d)d}{2p}h.$$

For each product, the length of the product block is given as the total length ℓ. Note that $1 - \frac{dn}{p}$ is the fraction of time during which the machine produces any product at rate d. Since all products are produced for an equal amount of time, the fraction of time during which one product is produced at rate d is $\frac{1}{n} - \frac{d}{p}$. Define $\alpha := \left(1 - \frac{1}{n} + \frac{d}{p}\right)$, yielding $x = \ell\alpha$. Note that in a tight schedule, $\alpha = 1$.

The optimal schedule S has total sequencing costs $W(S) = ns$ and total holding costs $H(S) = x^2\frac{(p-d)d}{2p}hn = \ell^2\alpha^2\frac{(p-d)d}{2p}hn$.

Thus, the average costs are given as

$$\frac{ns}{\ell} + \ell\alpha^2\frac{(p-d)d}{2p}hn.$$

We now find the optimal cycle length ℓ using that $W(S) = H(S)$. Given the optimal length, we can calculate the average total costs $\bar{c}(S)$, yielding

$$\ell = \frac{1}{\alpha}\sqrt{\frac{2sp}{(p-d)dh}} \quad \text{and} \quad \bar{c}(S) = n\alpha\sqrt{\frac{2s(p-d)dh}{p}}.$$

□

Using the characterization for identical products, we can construct a lower bound on the average costs of a schedule.

Lemma 13 (Lower bound on average costs) *Consider LSP(C,n) for $n > 1$. Let S^* be the optimal schedule. Let i be the product minimizing $\frac{(p_i-d_i)d_i}{2p_i}h_i$, and let $s^{\min} = \min_{i,j\in J} s_{i,j}$ be the minimum sequencing costs. Then,*

$$\bar{c}(S^*) \geq n\alpha\sqrt{\frac{2s^{\min}(p_i-d_i)d_ih_i}{p_i}}, \text{ where } \alpha = \left(1 - \frac{1}{n} + \frac{d_i}{p_i}\right).$$

Proof Intuitively, we construct a schedule for n identical products with d, p and h equal to the corresponding value of the least costly product. We use the notation from Lemma 12.

In order to lower-bound the holding costs, assume that we produce n times a certain new dummy product k, such that $h_k \leq h_j$ and $\frac{d_k}{p_k} \leq \frac{d_j}{p_j}$ for all $j \in J$. Furthermore, assume that the stock level is zero at the beginning and end of the product block, i.e., $q_k^{t_1} = q_k^{t_3} = 0$. From Lemma 12, we know that the holding costs of a single product block for k are equal to

$$\frac{xq}{2}h_k = x^2\frac{(p_k-d_k)d_k}{2p_k}h_k.$$

Now, let $x^2h^{\min}$ denote the minimum holding costs for each product during the block $[t_1, t_2, t_3)_k$, where

$$h^{\min} = \min_{i\in J}\frac{(p_i-d_i)d_i}{2p_i}h_i.$$

Choosing i as the product minimizing $h^{\min}$ and $s^{\min} = \min_{i,j\neq i\in J} s_{ij}$, we apply Lemma 12 to prove this lemma. □

The following lemma bounds the length of a specific class of feasible instances, which we use to create an upper bound on the average costs in Lemma 15.

Lemma 14 (Upper bound on schedule length for discrete case) *Consider LSP(D,n) for $n \geq 2$. Let S be any minimum length feasible simple cycle such that $c(S) = W(S) + H(S) \leq 5\, H(S)$. The length of S is bounded by*

$$\ell^{\max} = \left(\prod_{i\in J} p_i^-\right)\sqrt{\frac{\sum_{(i,j)\in TSP} s_{ij}}{2\sum_{i\in J}\frac{(p_i^- - d_i)d_i}{p_i^-}h_i}},$$

where $p_i^- = p_i \sum_{j\in J}\frac{d_j}{p_j}$ and TSP is the shortest possible simple cycle.[1]

Proof Since we are interested in a minimum length feasible schedule, we can assume that the schedule is tight by limiting the maximum production rates. To be precise, for each product $i \in J$ we limit production to

$$p_i^- = p_i \sum_{j\in J}\frac{d_j}{p_j},$$

yielding $\sum_{i\in J} d_i/p_i^- = 1$, which constitutes a tight schedule.

Now, since S is a simple cycle, we can calculate the total holding costs $H(S)$ as follows:

$$H(S) = \sum_{i\in J} h_i \int_0^\ell q_i^t \mathrm{d}t = \sum_{i\in J} \ell^2 \frac{(p_i^- - d_i)d_i}{2p_i^-} h_i,$$

where ℓ is the length of S. By definition of TSP, we have that $W(S) \geq \sum_{(i,j)\in TSP} s_{ij}$. Combining the above with the requirement that $c(S) = W(S) + H(S) \leq 5H(S)$, we get:

$$\ell \geq \ell^{\min} = \sqrt{\frac{\sum_{(i,j)\in TSP} s_{ij}}{2\sum_{i\in J}\frac{(p_i^- - d_i)d_i}{p_i^-}h_i}}.$$

Observe that in a feasible tight schedule for the discrete case, it must be that $\ell\frac{d_i}{p_i^-} \in \mathbb{N}_+$ for each product $i \in J$. Note that if ℓ is a multiple of $\prod_{i\in J} p_i^-$, this condition is satisfied. Any feasible schedule of length at least $\ell^{\min}$ now constitutes an upper bound on the length of S. Combining this condition with the above inequality yields an upper bound for ℓ of

$$\ell \leq \ell^{\max} = \left(\prod_{i\in J} p_i^-\right)\sqrt{\frac{\sum_{(i,j)\in TSP} s_{ij}}{2\sum_{i\in J}\frac{(p_i^- - d_i)d_i}{p_i^-}h_i}},$$

proving the lemma. □

We now present an upper bound on the average costs of an optimal discrete schedule.

Lemma 15 (Upper bound on average costs) *For $n \geq 2$ consider LSP(D,n). The average costs of an optimal schedule $\bar{c}(S^*)$ are bounded by*

$$\bar{c}(S^*) \leq \frac{5}{2}\sum_{i\in J} h_i \frac{(p_i - d_i)d_i}{p_i}\ell^{\max},$$

where $\ell^{\max}$ is defined as in Lemma 14.

Proof Let S be any minimum length feasible simple cycle such that $c(S) = W(S) + H(S) \leq 5\, H(S)$. As in Lemma 14, we know that $H(S) \leq \sum_{i\in J} \ell^2 \frac{(p_i - d_i)d_i}{2p_i} h_i \leq \sum_{i\in J} \ell^{\max} \frac{(p_i - d_i)d_i}{2p_i} h_i \ell$.

Recall that by assumption, $c(S) = W(S) + H(S) \leq 5H(S)$ and S has length $\ell \geq n$. Let S^* denote an optimal schedule. Now, observe that

$$\bar{c}(S^*) \leq \bar{c}(S) = \frac{W(S) + H(S)}{\ell} \leq \frac{5H(S)}{\ell}.$$

Substituting $H(S)$ by its upper bound proves the lemma. □

Using the previous lemma, we can now bound the maximum stock level.

Lemma 16 (Maximum Stock level) *Consider LSP(D,n) for $n \geq 2$. The maximum stock level in an optimal schedule S^* is bounded by $Q = 5\left(\sum_{i\in J} h_i \frac{(p_i - d_i)d_i}{p_i}\right)\ell^{\max}$.*

Proof Observe that for this value of Q we have $\frac{Q}{2} \geq \bar{c}(S^*)$, where $\bar{c}(S^*)$ is in Lemma 15. Also, $\bar{c}(S^*) = \frac{H(S^*)+W(S^*)}{\ell} \geq \frac{H(S^*)}{\ell}$, which is trivially lower-bounded by $\frac{1}{2}\max_{t\in S, i\in J} q_i^t$. The lemma follows. □

We now have the necessary lemmata to bound the costs of an optimal discrete schedule in terms of an optimal continuous schedule.

Lemma 17 (Pseudopolynomial ratio) *Given an instance of the lot-sizing problem, let S_D and S_C be the optimal schedules for LSP(D,n) and LSP(C,n), respectively. Then, there is a polynomial $\xi([p_i]_J, [d_i]_J, [h_i]_J, [s_{ij}]_{J\times J})$, such that $\bar{c}(S_D) \leq \xi \cdot \bar{c}(S_C)$.*

Proof From Lemmata 13 and 15, we know that $\bar{c}(S_C) \geq \varphi_1$ and $\bar{c}(S_D) \leq \varphi_2$ for given polynomials φ_1 and φ_2. Hence, $\frac{\bar{c}(S_D)}{\bar{c}(S_C)} \leq \frac{\varphi_2}{\varphi_1}$, which is bounded by a polynomial ξ in $[p_i]_J$, $[d_i]_J$, $[h_i]_J$, and $[s_{ij}]_{J\times J}$. □

[1] Note that for practical purposes, the value of n is typically small; hence, computing the TSP does not pose a problem.

5 Approximation algorithms

Already for two products, the optimal schedule can have pseudopolynomial length (Gabay et al. (2014)). This poses an inherent problem in processing the problem in polynomial time, particularly in outputting the schedule in polynomial time.

In this section, we overcome these difficulties and present two approximation algorithms: First, we *augment* the problem and solve this to optimality, yielding an augmented polynomial time approximation algorithm for the discrete case. Next, we convert the augmented discrete solution into a feasible solution for the continuous case, yielding a polynomial time approximation algorithm. In both cases, the schedule produced has polynomial length. The algorithm constructs solutions in polynomial time given a constant number of products. Observe that the latter is a reasonable assumption: In real-life instances, the number of products is relatively small. Throughout this section, we assume S^* is an optimal cyclic schedule of length ℓ and q_i^t for $i \in J$ and $t = 0, \dots, \ell - 1$ denotes the optimal stock level in S^*.

The general idea is to *augment* the production and demand rates, i.e., we allow for slightly higher production rates and modestly adjusted demand rates. For a given $\delta > 0$, we lift the stock levels q_i^t for all i and t to powers of $(1 + \delta)$ and use augmentation to keep the schedule feasible. For every time unit t, we generate states, which are defined by stock levels q_i^t for each product $i \in J$ and the product being produced. By Lemma 16, the maximum stock level is bounded by Q, yielding a polynomial number of states. With these states, we create a state graph and find a minimum mean cycle using Karp's algorithm in Karp (1978), in order to get an optimal schedule for the augmented version of LSP(D,n). Finally, we balance the resulting schedule such that it becomes a close to optimal solution for LSP(D,n) and a feasible schedule for LSP(C,n), yielding the aforementioned approximation algorithms. See Algorithm 1 for the pseudocode of the algorithm.

Let a state $S_i = (q_1, \dots, q_n)$ be defined as an ordered set of stock levels q_j for each product $j \in J$, where subscript $i \in J$ denotes the last product which has been produced before reaching the current state. Let d_i^t denote the augmented demand for a product $i \in J$ in time unit t.

For each time unit t and a product i which is produced, we allow for augmented production rates r_i^t such that the total augmented production is no more than $(1 + \delta)$ times the total production in a feasible schedule. Specifically, we require that augmented production satisfies the following conditions:

$$r_i^t < p_i + \delta(q_i^t + p_i - d_i), \tag{3}$$

$$\sum_{t=0}^{\ell-1} (r_i^t - d_i^t) < \ell(p_i - d_i)(1 + \delta). \tag{4}$$

Algorithm 1: Augmentation Algorithm AugAlg

```
Data: A set $J$ of $n$ products with demand rates $d_i$, maximum
      production rate $p_i$ and holding costs $h_i$ for all $i \in J$.
Result: Augmented schedule $S_D$ and schedule $S_C$.
Create the set $\mathcal{S}$ of all states $S_i = (q_1, \dots, q_n)$;
Let $E = \emptyset$ be the set of state edges;
foreach pair of states $S_i, S_j \in \mathcal{S}$ do
    if $S_i(q_j) - d_j < S_j(q_j) \leq (S_i(q_j) + p_j - d_j)(1 + \delta)$ then
        if $(S_i(q_k) - d_k)/(1 + \delta) \leq S_j(q_k) \leq (S_i(q_k) - d_k)$for every $k \neq j \in J$ then
            Create directed edge $e = (S_i, S_j)$ with cost $c_e = s_{ij} + \frac{1}{2} \sum_{k \in J} \big(S_i(q_k) + S_j(q_k)\big)$;
Find the minimum mean cycle $C^*$ in $\mathcal{S}$ using Karp's algorithm, cf. Karp (1978), discarding edge progressions which do not admit Eqs. (3) to (6);
Extract augmented schedule $S_D$ from $C^*$;
Let $S_C \leftarrow S_D$ be a continuous schedule with $x^t$ the length of time slot $t$;
Let all demands $d_i^t \leftarrow d_i$ and decrease production rates in $S_C$, such that $r_i^t \leq p_i$ and $\sum_{t=0}^{\ell-1} r_i^t - d_i \leq 0$;
For each product $i$ with $\sum_{t=0}^{\ell-1} r_i^t < \ell d_i$, uniformly increase production rates $r_i^t < p_i$ until $\sum_{t=0}^{\ell-1} r_i^t = \ell d_i$ or $r_i^t \leftarrow p_i$ for all $t$;
foreach $i \in J$ such that $\sum_{t=0}^{\ell-1} r_i^t < \ell d_i$ do
    Simultaneously increase all $x^t$ in $S_C$ where $r_i^t > 0$ and decrease all $x^{t'}$ where $r_j^t > 0$ for all $j \neq j \in J$ such that $\sum_{t=0}^{\ell-1} r_j^t \geq \ell d_j$ remains true, until $\sum_{t=0}^{\ell-1} r_i^t = \ell d_i$;
return $S_D$,$S_C$
```

The first equation ensures for each time unit an upper bound on the augmented production rate, such that the next power of $(1 + \delta)$ can be reached for the stock level. Note that this actually augments the stock level rather than the production rates. In order to limit the total augmentation in terms of the production rates, the latter equation ensures that the total production in the augmented schedule is not more than $(1 + \delta)$ times the maximum possible production in the non-augmented schedule. In practice, we get an augmented schedule which is reasonably achievable with respect to the original input data.

Additionally, for each time unit t with product i that is not produced during t, for augmented demand rates $d_i^t \geq 0$ the following equation must hold:

$$\frac{q_i^t - d_i}{q_i^t - d_i^t} \leq 1 + \delta. \tag{5}$$

This equation ensures that demand rates are not increased more than necessary in order to retain stock levels within a factor of $(1 + \delta)$. Moreover, for all time units t and products i, we require the following to ensure that the total demand in the augmented schedule is not more than $(1 + \delta)$ times the total demand in the non-augmented schedule:

$$\ell d_i \leq \sum_{k=0}^{\ell-1} d_i^k \leq (1+\delta)\ell d_i. \qquad (6)$$

Note that transgressing from one state to the next is equivalent to a single time unit in a schedule for the discrete case. Let each edge (S_i, S_j) have costs $s_{ij} + \frac{1}{2}\sum_{k\in J} h_k\left(S_i(q_k) + S_j(q_k)\right)$.

Note that here $s_{ii} = 0$.

We now describe the algorithm (cf. Algorithm 1). In the first step of AugAlg, an augmented state graph is constructed, with a state for each combination of stock levels q_i for each product i, such that $q_i \leq Q(1+\delta)$ and q_i is a power of $(1+\delta)$, where Q is the maximum stock level as in Lemma 16. Let S_i be a state in the optimal schedule with $S_i(q_j)$ the stock level for product j in S_i.

An edge is added from state S_i to S_j if and only if $S_i(q_j) - d_j < S_j(q_j) \leq (S_i(q_j) + p_j - d_j)(1+\delta)$ and $(S_i(q_k) - d_k)/(1+\delta) \leq S_j(q_k) \leq (S_i(q_k) - d_k)$ for all $k \neq j \in J$.

Recall Karp's algorithm for finding a minimum mean cycle in a digraph. The algorithm uses a dynamic program to compute values $F_k(v)$ for each vertex v and each $0 \leq k \leq n$, where $F_k(v)$ denotes the minimum weight of an edge progression of length k from some arbitrarily chosen vertex s to v. Using these values, the algorithm computes the minimum mean cycle. We slightly adjust the dynamic program in Karp's algorithm: Upon evaluating the computed values $F_k(v)$, discard any of these edge progressions which do not admit Eqs. (3)–(6), ensuring that these conditions hold for the minimum mean cycle returned by the algorithm. Observe that the minimum mean cycle returned by Karp's algorithm constitutes a feasible augmented schedule to the problem.

Lemma 18 *Let S be a schedule for LSP(D,n) and let $\varepsilon > 0$. There exists an augmented schedule S' such that $\bar{c}(S') \leq (1+\varepsilon)\bar{c}(S)$.*

Proof Let S be a schedule for LSP(D,n), and let $q_j^t(\delta) \geq q_j^t(S)$ be the nearest power of $(1+\delta)$ greater than or equal to $q_j^t(S)$. Note that by Lemma 9, each product i in the augmented, as well as in the non-augmented solution must have at least one point where its stock level is zero. Denote this point as time unit 0_i with $q_i^{0_i}(S') = q_i^{0_i}(S) = 0$. For each product $i \in J$, starting at zero stock level $q_i^{0_i}(S)$ successively change rates $r_i^t(S)$ and d_i to $r_i^t(S')$ and $d_i^t(S')$ for each time unit t as follows.

- If $q_i^t(\delta) = q_i^t(S)$, then let $r_i^t(S')$ remain the same as in S.
- Otherwise, if product i is produced, let the production rate be $r_i^t(S') \leftarrow r_i^t(S) + q_i^t(\delta) - q_i^t(S)$. However, since we want to bound the increase in the costs of the augmented schedule, we bound the stock level throughout the augmented schedule. For every product, for each of its production periods, we ensure that the cumulative amount of stock up to that point is no more than $(1+\delta)$ times the corresponding original stock. Thus, if $\sum_{k=0}^{t-1}\left(r_i^k(S') - d_i\right) + (q_i^t(\delta) - q_i^t(S)) \geq t(p_i - d_i)(1+\delta)$, then choose $r_i^t(S')$ such that $q_i^t(S')$ becomes the largest power of $(1+\delta)$ such that $q_i^t(S') < q_i^t(\delta)$.
- If i is not produced, choose the smallest $d_i^t(S') \geq d_i$ such that $q_i^t(S')$ is a power of $(1+\delta)$.

Observe that every stock level in S is a power of $(1+\delta)$ and the schedule is feasible. Since every stock level increased by at most $(1+\delta)$, the total costs for the schedule are increased by at most $c(S)\delta$. Choosing $\varepsilon = \delta$ proves the lemma. □

Applying the above lemma to an optimal schedule and bounding the running time of AugAlg yields the following result. Recall that n is typically a constant, and thus we can assume the number of products to be fixed, yielding a polynomial running time.

Theorem 1 *Let S^* be an optimal schedule for LSP(D,n) and let $\varepsilon > 0$.* AugAlg *finds an* augmented *schedule S_D for LSP(D,n) such that $c(S_D) \leq (1+\varepsilon)c(S^*)$ in running time $O\left(\left(\log_{1+\delta}(Q)\right)^n n^2\right)$.*

Proof Consider AugAlg and observe that the algorithm finds a schedule S_D such that $\bar{c}(S_D) \leq (1+\varepsilon)\bar{c}(S^*)$, where S^* is the optimal schedule, as in Lemma 18.

Note that there are $O((\log_{1+\delta}(Q))^n n)$ states in $\mathcal{S}$, and thus at most $O((\log_{1+\delta}(Q))^n n^2)$ edges. Karp's algorithm works in $O(m+n)$ time, where m is the number of edges in the graph, proving the theorem. □

We can now prove that AugAlg is a polynomial time approximation algorithm for the continuous problem.

Theorem 2 *Let S^* be an optimal schedule for LSP(C,n) and let $\varepsilon > 0$.* AugAlg *finds a feasible schedule S_C for LSP(C,n) of polynomial length such that $c(S_C) \leq (1+\varepsilon)\xi c(S^*)$ in time $O\left(\left(\log_{1+\delta}(Q)\right)^n n^2\right)$.*

Proof Ensure that $\sum_{i\in J}\frac{d_i}{p_i} \leq 1$, otherwise there exists no feasible schedule. Run AugAlg to get an augmented schedule S_D for the corresponding instance of LSP(D,n). Observe that the schedule is a feasible augmented schedule for LSP(C,n).

First, lower the demand and production rates to feasible values, i.e., let all demands $d_i^t \leftarrow d_i$ and decrease production rates such that $r_i^t \leq p_i$. Next, for each product i for which total production does not cover total demand, i.e., for which $\sum_{t=0}^{\ell-1} r_i^t < \ell d_i$, uniformly increase production rates $r_i^t < p_i$ until demand is satisfied, i.e., $\sum_{t=0}^{\ell-1} r_i^t = \ell d_i$, or until $r_i^t = p_i$ for all t. If the former is not the case, we cannot satisfy total demand for product i and the lengths of production periods will need to be adjusted. Denote the schedule obtained after this transformation by S'_D. Since demand and production rates are decreased by at most a factor of $(1+\delta)$, overproduction in S'_D is no more than $(1+\delta)\ell(p_i - d_i)$; therefore, the costs are bounded as $c(S'_D) \leq (1+\delta)c(S_D)$.

Let x^t denote the length of time slot t. Clearly, there are ℓ time slots. For each product $i \in J$ such that $\sum_{t=0}^{\ell-1} r_i^t < \ell d_i$, we will increase all production lengths x^t where $r_i^t > 0$ to meet the demand of product i. To retain feasibility for all products $j \neq i \in J$, we increase production rates and shorten production periods where possible, while keeping the schedule length constant. For each product $j \in J$ such that $\sum_{t=0}^{\ell-1} r_j^t \geq \ell d_j$, we consider the following three numbered categories:

1. For all t where $0 < r_j^t < p_j$, we will increase r_j^t and decrease x^t such that total production in x^t remains unchanged, at most up to the point where $r_j^t = p_j$.
2. If $\sum_{t=0}^{\ell-1} r_j^t > \ell d_j$ and $r_j^t \in \{0, p_j\}$ for all t, we will decrease lengths of production periods x^t with $r_j^t > 0$, at most up to the point where $\sum_{t=0}^{\ell-1} r_j^t = \ell d_j$.
3. If $\sum_{t=0}^{\ell-1} r_j^t = \ell d_j$ and $r_j^t \in \{0, p_j\}$ for all t, the schedule is tight for this product.

For each $i \in J$ such that $\sum_{t=0}^{\ell-1} r_i^t < \ell d_i$, simultaneously increase all x^t where $r_i^t > 0$, increase r_j^t and decrease x^t for all products j as in Category 14, and decrease x^t for all products j as in Category 14, while keeping the schedule length ℓ constant. Note that the category number of a production period can only be increased by applying the transformation. Hence, since $\sum_{i \in J} \frac{d_i}{p_i} \leq 1$, and each production period can be categorized as above, this transformation terminates successfully. Finally, for any product which is produced more than the total demand throughout the cycle, we uniformly decrease production rates for this product—without altering the length of the production period—until demand is met exactly. We denote the resulting schedule by S_C. For the remainder of this proof, we assess the quality of S_C.

First look at a single increment of length $\alpha \leq \delta$ for a time unit t where i is produced and $\sum_{t=0}^{\ell-1} r_i^t(S'_D) < \ell d_i$. Let $c_j^t(S)$ denote the costs for a product $j \in J$ during time slot t in a schedule S. Since the production rate is increased in the transformation by a factor $(1+\alpha)$, the costs for product i at time slot t are bounded by $(1+\alpha)c_i^t(S'_D)$. Similarly, the cost for each product $j \neq i \in J$ is increased to at most $(1+\alpha)c_j^t(S'_D)$. At the end of every production period, stock levels in S_C are not increased compared to stock levels in S_D.

Secondly, look at a single decrement of α for a time unit t where i is produced. Clearly, the costs $c_i^t(S_C)$ do not increase. Furthermore, the costs $c_j^t(S)$ for each product $j \neq i \in J$ are neither increased. Since the production period is shortened, the stock level for each product $j \neq i$ at the end of the production period is increased. In a worst-case scenario, this extra stock needs to be carried throughout the entire schedule. Hence, for each decrement of α, for each product $j \neq i$, total costs for the product in the entire schedule can be increased by at most $\alpha d_j h_j \ell$. Observe that $\alpha d_j h_j \ell \leq \delta c_j(S)$.

Recall that the maximum increment for a single time unit is at most δ. Each product, for which time units are increased, increases the total costs for all products by at most $\delta c(S'_D)$.

Furthermore, for each product for which time units are decreased, costs increase by at most $\delta c(S'_D)$ in total. Thus, AUGALG produces a feasible schedule S for LSP(C,n) of costs at most $(1+n\delta)c(S'_D)$. From Lemma 17, we know that $\bar{c}(S_D) \leq \xi \bar{c}(S^*)$.

Hence, $c(S_C) \leq (1+n\delta)c(S'_D) \leq (1+n\delta)(1+\delta)c(S_D) \leq (1+n\delta)(1+\delta)^2 \xi c(S^*)$. Choosing ε such that $(1+\varepsilon) = (1+n\delta)(1+\delta)^2$ proves the theorem. □

6 Discussion and future work

This article combines the hardness of high-multiplicity encoding with sequence-dependent setup costs, both of which are natural properties of real-life problems. Not only does this introduce hardness akin to the TRAVELING SALESMAN PROBLEM, but due to the compact encoding it is not clear whether or not a polynomially sized certificate can be constructed, even for very restricted cases. We discussed the complexity of the problem and presented structural properties largely characterizing optimal schedules, which can be used for future algorithms and computational experiments. We presented a polynomial time augmented approximation algorithm, which finds $(1+\varepsilon)$-approximate augmented solutions for the discrete variant of the problem and $(1+\varepsilon)\xi$-approximate solutions for the continuous case. In contrast to the known complexity of the problem, the algorithm runs in polynomial time and yields schedules of polynomial length.

It is unclear whether it can be guaranteed that an optimal schedule exists at all. Consider the case of LSP(C,2) in Gabay et al. (2014), where the optimal schedule is already irrational even under rational input values. Now consider LSP(C,3). Is it possible that due to the irrationality of the cost balance, the optimal schedule has infinite length? Can it nevertheless be approximated with a finite schedule? Considering instances with two products, can we characterize the optimal solutions for the discrete case? We conjecture this is possible to achieve using techniques similar to the ones used in this paper.

Alternatively, consider the settings where we explicitly make assumptions concerning the input instances. For instance, if the sequence is given, e.g., using a TSP oracle, is it possible to find an (approximately) optimal solution for both cases in polynomial time? Or if the sequencing costs have a lexicographical ordering (e.g., when the products only differ in color and setting up the machine when switching between two similar colors costs less), can we obtain stronger results?

Regarding the complexity of the problem, we conjecture that this problem is contained in a higher complexity class than NP: Already for LSP(F,1) and LSP(C,2), the optimal schedule can be of non-polynomial length. Although the schedule for these cases can still be represented in polynomial time, it is uncertain if this can be done for arbitrary numbers of products. Furthermore, consider the following decision problem: Does there exist an optimal cyclic schedule of average costs k? It is unclear whether this decision problem is contained in NP and how an adequate polynomial certificate for a NO instance can be constructed.

References

Bertsimas, D., Gamarnik, D., & Sethuraman, J. (2003). From fluid relaxations to practical algorithms for high-multiplicity job-shop scheduling: The holding cost objective. *Operations Research*, *51*(5), 798–813.

Boctor, F. (1982). The two-product, single-machine, static demand, infinite horizon lot scheduling problem. *Management Science*, *28*(7), 798–807.

Brauner, N., Crama, Y., Grigoriev, A., & van de Klundert, J. (2005). A framework for the complexity of high-multiplicity scheduling problems. *Journal of Combinatorial Optimization*, *9*(3), 313–323.

Brauner, N., Crama, Y., Grigoriev, A., & van de Klundert, J. (2007). Multiplicity and complexity issues in contemporary production scheduling. *Statistica Neerlandica*, *61*(1), 75–91.

Clifford, J., & Posner, M. (2000). High multiplicity in earliness–tardiness scheduling. *Operations Research*, *48*(5), 788–800.

Clifford, J., & Posner, M. (2001). Parallel machine scheduling with high multiplicity. *Mathematical Programming*, *89*(3), 359–383.

Filippi, C., & Romanin-Jacur, G. (2009). Exact and approximate algorithms for high-multiplicity parallel machine scheduling. *Journal of Scheduling*, *12*(5), 529–541.

Gabay, M., Grigoriev, A., Kreuzen, V. J. C., & Oosterwijk, T. (2016). High multiplicity scheduling with switching costs for few products. In: M. Lübbecke, A. Koster, P. Letmathe, R. Madlener, B. Peis & G. Walther (Eds.), Operations research proceedings 2014 (pp. 437–443). Springer International Publishing

Gallego, G., & Shaw, D. X. (1997). Complexity of the elsp with general cyclic schedules. *IIE Transactions*, *29*(2), 109–113.

Goyal, S. (1973). Scheduling a multi-product single machine system. *Journal of the Operational Research Society*, *24*(2), 261–269.

Haase, K. (1996). Capacitated lot-sizing with sequence dependent setup costs. *Operations-Research-Spektrum*, *18*(1), 51–59.

Haase, K., & Kimms, A. (2000). Lot sizing and scheduling with sequence-dependent setup costs and times and efficient rescheduling opportunities. *International Journal of Production Economics*, *66*(2), 159–169.

Hochbaum, D., & Shamir, R. (1991). Strongly polynomial algorithms for the high multiplicity scheduling problem. *Operations Research*, *39*(4), 648–653.

Holmbom, M., & Segerstedt, A. (2014). Economic order quantities in production: From Harris to economic lot scheduling problems. *International Journal of Production Economics*, *155*(C), 82–90.

Karp, R. (1978). A characterization of the minimum cycle mean in a digraph. *Discrete Mathematics*, *23*(3), 309–311.

Madigan, J. (1968). Scheduling a multi-product single machine system for an infinite planning period. *Management Science*, *14*(11), 713–719.

Narro Lopez, M., & Kingsman, B. (1991). The economic lot scheduling problem: Theory and practice. *International Journal of Production Economics*, *23*(1–3), 147–164.

Rothkopf, M. (1966). The traveling salesman problem: On the reduction of certain large problems to smaller ones. *Operations Research*, *14*(3), 532–533.

Wetsels, N. (2012). Production cycle problem. Master's thesis, Maastricht University, School of Business and Economics

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