Noname manuscript No.
(will be inserted by the editor)

Lot-Size Scheduling of a Single Product on Unrelated Parallel Machines

Anton V. Eremeev ·
Mikhail Y. Kovalyov ·
Pavel M. Kuznetsov

Received: date / Accepted: date

Abstract We study a problem in which at least a given quantity of a single product has to be partitioned into lots, and lots have to be assigned to the unrelated parallel machines for processing so that the maximum machine completion time or the sum of machine completion times is minimized. Machine dependent lower and upper bounds on the lot size are given. The product can be continuously divisible or discrete. We derive optimal polynomial time algorithms for several special cases of the problem. For other cases we provide NP-hardness proofs and demonstrate existence of fully polynomial time approximation schemes.

Keywords Scheduling · Parallel machines · Lot-sizing · Computational complexity · Approximation

1 Introduction

The following problem has been observed as a special case in the production of chemical granules, see Shaik et al. [18], and fuel supply management, see Austin

The research presented in Section 2 is supported by the Russian Science Foundation grant 15-11-10009. Research presented in Section 3 is supported by the program of fundamental scientific researches of the SB RAS I.5.1., project 0314-2016-0019.

Anton V. Eremeev
Sobolev Institute of Mathematics, Siberian Branch of Russian Academy of Sciences, Novosibirsk, Russia
E-mail: eremeev@ofim.oscsbras.ru

Mikhail Y. Kovalyov
United Institute of Informatics Problems, National Academy of Sciences of Belarus, Minsk, Belarus
E-mail: kovalyov_my@newman.bas-net.by

Pavel M. Kuznetsov
Dostoevsky Omsk State University, Omsk, Russia
E-mail: kuznetsovpm@yandex.ru

and Hogan [2]. Its solution can be employed in metaheuristics for solving a more general problem, similar to that in Borisovsky [3]. There is a demand for at least A units of a single product to be produced on m, $m \geq 2$, unrelated parallel machines in lots. The size of a lot is the number of units of the product this lot includes. It can take either integer value or real value, in which cases the product is called discrete and continuous, respectively. Machine i requires $p_i \cdot x$ time units to produce a lot of size x, $i = 1, \dots, m$. If a lot of size x is assigned to machine i, then the lot size must satisfy the relation $l_i \leq x \leq u_i$, where l_i and u_i are given lower and upper bounds, $i = 1, \dots, m$. The problem is to partition at least A units of the product into lots and assign these lots to the machines so that the maximum machine completion time or the total machine completion time is minimized. Both criteria are related to the fair distribution of the machine workloads. Setup times and costs are assumed to be negligibly small, and therefore, they are not considered. It is also assumed that the values A, u_i and p_i, $i = 1, \dots, m$, are positive integer numbers, and the values l_i, $i = 1, \dots, m$, are non-negative integer numbers.

We denote this problem as $R|1, \alpha, \beta|\gamma$, where $\alpha \in \{lot, GT\}$, $\beta \in \{cntn, dscr\}$, $\gamma \in \{C_\Sigma, C_{\max}\}$. Following scheduling traditions, notation R refers to unrelated parallel machines. Notation 1 in the middle field indicates the case of a single product. Notations *lot* and *GT* are used to distinguish the general case (*lot*) and the case in which at most one lot can be assigned on the same machine (*GT* – Group Technology). Abbreviations *cntn* and *dscr* specify continuous and discrete product, respectively. Maximum machine completion time and total machine completion time are denoted as $C_{\max}$ and C_Σ, respectively.

The following mathematical programming formulation for the problem $R|1, \alpha, \beta|\gamma$ can be given.

$$\min \sum_{i=1}^{m} C_i, \quad \text{if } \gamma = C_\Sigma, \tag{1}$$

$$\min\ C_{\max}, \quad \text{if } \gamma = C_{\max}, \tag{1$'$}$$

$$C_{\max} \geq C_i, \quad i = 1, \dots, m, \text{ if } \gamma = C_{\max}, \tag{1$''$}$$

$$C_i = p_i x_i, \quad i = 1, \dots, m, \tag{2}$$

$$\sum_{i=1}^{m} x_i \geq A, \tag{3}$$

$$z_i l_i \leq x_i \leq z_i u_i, \quad i = 1, \dots, m, \tag{4}$$

$$x_i \in R_+, \quad \text{if } \beta = cntn, \tag{5}$$

$$x_i \in Z_+, \quad \text{if } \beta = dscr, \tag{5$'$}$$

$$z_i \in Z_+, \quad \text{if } \alpha = lot, \tag{6}$$

$$z_i \in \{0, 1\}, \quad \text{if } \beta = GT. \tag{6$'$}$$

The variables are the maximum machine completion time $C_{\max}$, the completion time C_i of machine i, the production volume x_i on machine i, and the number of lots z_i on machine i, $i = 1, \ldots, m$. Equalities (2) link machine completion times to the production volumes on these machines. Relation (3) ensures that the required quantity of the product is assigned to the machines. Relations (4) connect production volume x_i with the number of lots z_i on each machine i. This constraint is met for machine i if and only if there exists a number z_i of lots, whose sizes are between l_i and u_i and the total production volume on the machine is x_i. Conditions (5) and (5′) address the assumptions that the product is either continuously divisible (5) or discrete (5′). Conditions (6) and (6′) define the admissible number of lots on each machine.

Note that if the condition (4) is satisfied and $z_i \geq 1$, then the corresponding lot sizes may be set to x_i/z_i in the case of $\beta = cntn$. In the case $\beta = dscr$, all lot sizes on machine i can be chosen from the set $\{\phi_i, \psi_i\}$ of two numbers, where $\phi_i = \lceil x_i/z_i \rceil$ and $\psi_i = \lfloor x_i/z_i \rfloor$. If $\phi_i = \psi_i$, then the sizes of all lots on machine i can be set to x_i/z_i, alternatively, if $\phi_i = \psi_i + 1$, then $k_i = x_i - z_i\psi_i$ lots will be of size ϕ_i and $z_i - k_i$ lots will be of size ψ_i, which is implied by the equation $k_i\phi_i + (z_i - k_i)(\phi_i - 1) = x_i$.

Problem $R|1, \alpha, \beta|\gamma$ falls into the category of batch scheduling problems (Potts and Kovalyov [16], Allahverdi et al. [1]), for which terminologies "lot-sizing" (Potts and Van Wassenhove [17], Chen et al. [6]) and "job splitting" (Logendran and Subur [12], Tahar et al. [20]) are also used, especially, in the situations where a partition of a group of identical items (a job) into lots (job sections) appears to be more natural than their unification into batches. The specificity of the problem $R|1, \alpha, \beta|\gamma$ distinguishing it from the models in the above references is the presence of lower and upper bounds on the lot sizes. The most closely related problem was studied by Dolgui et al. [7], in which there are several products, machine dependent lot size lower bounds, machine and sequence dependent setup times, and the objective is to minimize $C_{\max}$. This problem is strongly NP-hard because the Traveling Salesman Problem reduces to it. In [7], it was proved NP-hard even if the number of products is $n = 2$. Several dynamic programming algorithms for the special cases were developed. Results from [7] can not be employed for $R|1, \alpha, \beta|C_{\max}$ because the upper bounds u_i on the lot sizes are not considered in [7].

The so-called Supply Scheduling Problem (SSP) is also closely related to the problem in this paper. In SSP, there are m providers that supply a certain product to a manufacturing unit. If provider i is not used, then the corresponding delivered quantity is $x_i = 0$. If provider P_i is used, then the delivered quantity x_i must be between the given lower and upped bounds l_i and u_i. The demand at the manufacturing unit is A. The delivery cost for sending a quantity x_i from provider P_i to the manufacturing unit is $c_i(x_i)$, where $c_i(\cdot)$ is a cost function which can be linear as it is in Chauhan et al. [4] and Eremeev et al. [9], or it can be given by an oracle as it is in Chauhan et al. [5] and Ng et al. [14]. The goal is to minimize the total delivery cost, subject to the condition that the manufacturing demand is satisfied. The SSP is NP-hard in the ordinary sense and several FPTASes are proposed for different versions of this problem in [4,5,9,14]. A *Fully Polynomial Time Approximation Scheme (FPTAS)* is a collection of $(1+\varepsilon)$-*approximation algorithms* $\{A_\varepsilon\}$ such that algorithm A_ε guarantees relative error ε, and it runs in time polynomial in $1/\varepsilon$ and in the problem instance length in binary encoding. FPTASes are theoretically best approaches to handling NP-hard problems (Garey and Johnson [10]). Their experimental verification for knapsack type problems shows good performance and solution quality (see, e.g., Martello and Toth [13] and Kovalyov et al. [11]).

A modification of SSP with a requirement to supply *exactly* A units of the product and piecewise concave cost functions was studied by Shor and Stecuk [19]. An important structural property of optimal solutions was established and a dynamic programming algorithm was developed. A generalization of SSP to the case of concave non-decreasing cost functions and several feasible intervals for the delivered quantity was studied by Eremeev et al. [8]. An FPTAS was developed. It employs a property similar to that in [19].

In the next section, the case of at most one lot on each machine (GT) is studied. Polynomial time algorithms are developed for all variations of the problem of minimizing $C_{\max}$. All variations of the problem of minimizing C_Σ are shown NP-hard, and FPTASes are presented. In Section 3, the general case (lot) is studied. Again, polynomial time algorithms are developed for all variations of the problem of minimizing $C_{\max}$, and all variations of the problem of minimizing C_Σ are shown NP-hard. FPTASes are given for the NP-hard problems. The paper concludes with a table of the results and suggestions for future research.

2 At most one lot on each machine

In this section, we study the problem $R|1, GT, \beta|\gamma$, $\beta \in \{cntn, dscr\}$, $\gamma \in \{C_\Sigma, C_{\max}\}$.

2.1 Minimizing total machine completion time

For the problem $R|1, GT, \beta|C_\Sigma$, we only note that it is equivalent to the earlier studied NP-hard problem SSP with linear cost functions proportional to the delivered quantities. This latter problem is well studied. For example, it admits an FPTAS with running time $O(m^3/\varepsilon)$, see Eremeev et al. [4,9] and Ng et al. [14].

A special case of $R|1, GT, \beta|C_\Sigma$, $\beta \in \{cntn, dscr\}$ in which $u_i = A$, $i = 1, \dots, m$, is trivially solvable in $O(m)$ time by selecting a machine with the minimal p_i value and allocating A product units to this machine. Another special case with $l_i = l$ and $u_i = u$, $i = 1, \dots, m$, is solvable in $O(m \log m)$ time by a greedy algorithm, which starts with $x = (0, \dots, 0)$ and iteratively re-sets production volume $x_k := \max\{l, \min\{u, A - \sum_i x_i\}\}$ to the machine k with the minimal value p_i among those with $x_i = 0$.

2.2 Minimizing maximum machine completion time

In this sub-section, we describe an optimal $O(m \log m)$ time solution algorithm for the problem $R|1, GT, cntn|C_{\max}$.

We begin with computing values $l_i p_i$ and $u_i p_i$, $i = 1, \ldots, m$, and sorting these values in non-decreasing order. Denote the set of distinct values $l_i p_i$ and $u_i p_i$, $i = 1, \ldots, m$, as $\{D_1, \ldots, D_t\}$, where $t \le 2m$. Assume without loss of generality that $D_1 < \cdots < D_t$. Assume that $t \ge 2$ because otherwise the problem is trivial. Denote by C^* the optimal solution value of the problem $R|1, GT, cntn|C_{\max}$. It is easy to see that there exists an index k^*, $1 \le k^* \le t-1$, such that $D_{k^*} \le C^* \le D_{k^*+1}$. For any $k, k = 1, \ldots, t-1$, let us partition machines into the following three sets:

$$\text{Before}_k = \{i | u_i p_i < D_k\},$$

$$\text{Between}_k = \{i | l_i p_i \le D_k \le D_{k+1} \le u_i p_i\},$$

$$\text{After}_k = \{i | D_{k+1} < l_i p_i\}.$$

The above three definitions account for the possible location of the interval $[D_k, D_{k+1}]$ with respect to the interval $[l_i p_i, u_i p_i]$. It is clear that there exists an optimal solution x^* such that $x_i^* = u_i$ for $i \in \text{Before}_{k^*}$, $x_i^* = 0$ for $i \in \text{After}_{k^*}$ and

$$x_i^* p_i = \max\{x_i p_i \mid x_i p_i \le C^*\} \text{ for } i \in \text{Between}_{k^*}.$$

The latter equalities are equivalent to $x_i^* = C^*/p_i$ for $i \in \text{Between}_{k^*}$. We deduce

$$\sum_{i=1}^{m} x_i^* = \sum_{i \in \text{Before}_{k^*}} u_i + C^* \sum_{i \in \text{Between}_{k^*}} (1/p_i) \ge A,$$

and hence,

$$C^* \ge \frac{A - \sum_{i \in \text{Before}_{k^*}} u_i}{\sum_{i \in \text{Between}_{k^*}} (1/p_i)}.$$

Denote $U_k := \sum_{i \in \text{Before}_k} u_i$ and $I_k := \sum_{i \in \text{Between}_k} (1/p_i)$ for all $k = 1, \ldots, t$. Note that either $C^* = D_{k^*}$ or the total production volume in all optimal solutions is exactly A and $C^* = (A - U_{k^*})/I_{k^*}$ (or both conditions hold). Thus, the problem $R|1, GT, cntn|C_{\max}$ can be solved by computing $t-1$ values

$$R_k = \max\left\{D_k, \frac{A - U_k}{I_k}\right\}, \quad k = 1, \ldots, t-1, \tag{7}$$

and determining an index k^* such that

$$R_{k^*} = \min\{R_k \mid R_k \le D_{k+1},\ k = 1, \ldots, t-1\}.$$

The optimal solution value can be calculated as $C^* = R_{k^*}$, and the corresponding optimal values x_i^* can be determined such that $x_i^* = u_i$ for $i \in \text{Before}_{k^*}$, $x_i^* = 0$ for $i \in \text{After}_{k^*}$ and $x_i^* = C^*/p_i$ for $i \in \text{Between}_{k^*}$.

Assume that the values $D_1, \dots, D_t$ are given such that $D_1 < \dots < D_t$. Then all the sets Before_k, Between_k, After_k, $k = 1, ..., t-1$, can be computed in $O(m)$ time. Furthermore, if the values U_k, I_k and R_k are known for some k, then the values U_{k+1}, I_{k+1} and R_{k+1} can be computed in a constant time. Therefore, the index k^* can be computed in $O(t) = O(m)$ time. We obtain the following proposition.

Proposition 1 *The problem $R|1, GT, cntn|C_{\max}$ can be solved in $O(m \log m)$ time.*

A special case of problem $R|1, GT, cntn|C_{\max}$, in which $u_i \geq A$, $i = 1, \dots, m$, is a special case of the problem $R|\Delta s_{lij}, cntn|C_{\max}$ with n products studied in [7]. By using notation from [7], this special case can be represented as $R|\Delta s_{lij}, cntn|C_{\max}$ with extra assumptions that $n = 1$ and an upper bound on the total production is $B \geq \sum_{i=1}^{m} u_i$. The best algorithm proposed for this special case in [7] runs in $O(\tau_\beta 2^m)$ time, were τ_β is the time to solve a supplementary linear programming problem.

It sometimes happen that an optimal solution for the discrete case of a problem can be obtained by an appropriate rounding of the components of an optimal solution of the corresponding continuous case. Let us show that this approach does not work for the problem $R|1, GT, dscr|C_{\max}$. Consider an example in which $m = 3$, $A = 68$, $l_1 = l_2 = l_3 = 0$, $u_1 = u_2 = u_3 = +\infty$, $p_1 = 9$, $p_2 = p_3 = 88$. An optimal solution of the continuous problem is vector x^0 with the components $x_1^0 = C^0/p_1 \approx 56.45$ and $x_2^0 = x_3^0 = C^0/p_3 \approx 5.8$, where $C^0 = Ap_1p_2/(p_2 + p_1 + p_1) \approx 508.1$ is the optimal makespan value. The only way to obtain a feasible solution for the corresponding discrete problem from x^0 is to round up at least two of its components, which gives the makespan value of at least 528. However, the optimal makespan value for the corresponding discrete problem is 522, which is attained for $x^* = (\lceil x_1^0 \rceil + 1, \lfloor x_2^0 \rfloor, \lfloor x_3^0 \rfloor)$.

The discrete problem $R|1, GT, dscr|C_{\max}$ can be solved by the following bisection search algorithm. Let C^* denote the optimal makespan value. Recall that the values A, u_i and p_i, $i = 1, \dots, m$, are positive integer numbers, and the values l_i, $i = 1, \dots, m$, are non-negative integer numbers.

Algorithm $BiSec(GT)$

Step 1 (Search range) Calculate $r = \arg\min_i\{l_i p_i\}$. If $l_r \geq A$, then vector x with $x_r = l_r$ and $x_i = 0$ for $i \neq r$ is an optimal solution. Stop. If $\sum_{i=1}^{m} u_i < A$, then there is no feasible solution. Stop. If $l_r < A \leq \sum_{i=1}^{m} u_i$, then initialize $LB := \min_i\{l_i p_i\}$ and $UB := \max_i\{u_i p_i\}$. Note that $C^* \in [LB + 1, UB]$. Calculate feasible solution $\tilde{x} := (u_1, \dots, u_m)$ with value UB. Set $k := 1$.

Step 2 (New solution $x^{(k)}$) Calculate $F_k := \lfloor (LB + UB)/2 \rfloor$ and $x_i^{(k)} := \lfloor F_k/p_i \rfloor$, $i = 1, \dots, m$. If $x_i^{(k)} < l_i$, then re-set $x_i^{(k)} := 0$, and if $x_i^{(k)} > u_i$, then re-set $x_i^{(k)} := u_i$, $i = 1, \dots, m$. Re-set $F_k := \max_i\{x_i^{(k)} p_i\}$.

Step 3 (Testing feasibility) If $\sum_i x_i^{(k)} \geq A$, then $x^{(k)}$ is feasible. In this case, re-set $UB := F_k$ and $\tilde{x} := (x_1^{(k)}, \dots, x_m^{(k)})$. If $\sum_i x_i^{(k)} < A$, then $x^{(k)}$ is infeasible. In this case, re-set $LB := \lfloor (LB+UB)/2 \rfloor$ and note that there is no feasible solution with value LB or less, because $x^{(k)}$ maximizes the production volume provided that $C_{\max} \le \lfloor (LB+UB)/2 \rfloor$.

Step 4 (Testing optimality) If $UB - LB > 1$, then re-set $k := k+1$ and repeat Step 2. If $UB - LB = 1$ and $x^{(k)}$ is feasible, then $x^{(k)}$ is an optimal solution with value F_k, because the $C_{\max}$ value is integer, there is no feasible solution with value $C_{\max} = LB$ and $x^{(k)}$ with value $C_{\max}(x^{(k)}) = UB$ is feasible. Stop. If $UB - LB = 1$ and $x^{(k)}$ is infeasible, then $\tilde{x}$ is an optimal solution with value UB. Stop.

Observe that, in iteration k of Step 2, there is no feasible solution with the makespan value $C^0_{\max}$ such that $C_{\max}(x^{(k)}) = \max_i\{x_i^{(k)} p_i\} < C^0_{\max} < \lfloor (LB+UB)/2 \rfloor$ because each value $x_i^{(k)}$ is maximum in the domain $0 \cup [l_i, u_i]$ provided that $C_{\max} \le \lfloor (LB+UB)/2 \rfloor$. This observation, together with a standard justification of the bisection search algorithms, leads to the following proposition.

Proposition 2 *The problem $R|1, GT, dscr|C_{\max}$ can be solved in weakly polynomial time $O(m \log \max_i\{u_i p_i\})$.*

3 Arbitrary number of lots on each machine

In this section, we consider the general problem $R|1, lot, \beta|\gamma$, $\beta \in \{cntn, dscr\}$, $\gamma \in \{C_{\max}, C_\Sigma\}$. First of all, observe that if $u_i \ge 2l_i$, $i = 1, \dots, m$, then the intervals $[l_i, u_i], [2l_i, 2u_i], \dots$ of feasible production volumes on machine i merge into a single interval $[l_i, A]$, $i = 1, \dots, m$. In this case, the problem $R|1, lot, \beta|\gamma$ and the problem $R|1, GT, \beta|\gamma$, in which $u_i = A$, $i = 1, \dots, m$, are equivalent for any β and γ. Therefore, the algorithmic results of Section 2 for the problem with GT assumption apply for the problem $R|1, lot, \beta|\gamma$ in this case. In the rest of this section, we assume that the condition $u_i \ge 2l_i$, $i = 1, \dots, m$, is not satisfied.

3.1 Minimizing total machine completion time

Firstly, note that a special case of the problem $R|1, lot, \beta|C_\Sigma$, $\beta \in \{cntn, dscr\}$, in which $u_i = A$, $i = 1, \dots, m$, can be solved in $O(m)$ time in the same way as for the problem $R|1, GT, \beta|C_\Sigma$. A special case with $l_i = l$ and $u_i = u$, $i = 1, \dots, m$, can be solved in $O(m)$ time by selecting a machine with the minimal p_i value and allocating $\max\{l\lceil A/u \rceil, A\}$ units of the product to this machine.

In the general setting, the problem $R|1, lot, \beta|C_\Sigma$ for $\beta = cntn$ or $\beta = dscr$ is NP-hard, because the well-known NP-complete decision problem Integer Knapsack [10] polynomially reduces to its special case in which $u_i = l_i$, $i = 1, \dots, m$. We now show that this problem admits an FPTAS.

Theorem 1 *The problem $R|1, lot, \beta|C_\Sigma$, $\beta \in \{cntn, dscr\}$, admits an FPTAS with the running time $O\left(\frac{m^2}{\varepsilon^2}\log r + m^2 \log r \log\log r\right)$, where $r = \min_i\{p_i(A+l_i)\}/(p_{\min}A)$, $p_{\min} = \min_i\{p_i\}$.*

Proof. Ng et al. [14] presented an FPTAS for the problem SSP, which can be formulated as follows.

$$\min \sum_{i=1}^{n} c_i(x_i),$$

subject to

$$\sum_{i=1}^{n} x_i \geq B,$$

$$x_i \in X_i = \{0\} \cup [a_i, b_i], \quad i = 1, \dots, n.$$

The following assumptions were employed in [14]:

1. Each function $c_i(x)$ is defined on a set $\overline{X}_i \subseteq X_i$, $i = 1, \dots, n$.
2. $b_i \in \overline{X}_i$, $i = 1, \dots, n$.
3. $\sum_{i=1}^{n} b_i \geq B$.
4. $0 \leq c_i(x) < \infty$ for $x \in \overline{X}_i$, $i = 1, \dots, n$.
5. Each function $c_i(x)$ is continuous in each point $x \in \overline{X}_i$, $i = 1, \dots, n$.
6. Given $x \in \overline{X}_i$, the value $c_i(x)$ can be computed in a constant time, $i = 1, \dots, n$.
7. Given a real number t, each value $\max\{x \mid x \in \overline{X}_i, c_i(x) \leq t\}$ can be computed in a constant time, $i = 1, \dots, n$.
8. Each value $c_i^{\min} = \min\{c_i(x) \mid x \in \overline{X}_i, c_i(x) > 0\}$ can be computed in a constant time, $i = 1, \dots, n$.

The problem $R|1, lot, \beta|C_\Sigma$ is a special case of the problem SSP, in which $n = m$, $B = A$, $a_i = l_i$, the cost functions are

$$c_i(x_i) = \begin{cases} p_i x_i & \text{if } x_i \in \overline{X}_i, \\ +\infty, & \text{otherwise}, \end{cases} \qquad i = 1, \dots, m,$$

the sets $\overline{X}_i$ are

$$\overline{X}_i = \begin{cases} \cup_{k=1}^{K_i-1}[kl_i, ku_i] \cup [K_i l_i, A] \cup \{0\}, & \text{if } A \in [K_i l_i, K_i u_i], \\ \cup_{k=1}^{K_i-1}[kl_i, ku_i] \cup \{0\}, & \text{otherwise}, \end{cases}$$

where $K_i = \lceil A/u_i \rceil$, $i = 1, \dots, m$, i.e. K_i is the minimum integer number that satisfies $Ku_i \geq A$. Finally, for each $i = 1, \dots, m$ we assume $b_i = A$ if $A \in [K_i l_i, K_i u_i]$, and $b_i = (K_i - 1)u_i$ otherwise.

For this special case all the assumptions in [14] are satisfied. To calculate the function $c_i(x)$, it is sufficient to recognize the case $x \in \overline{X}_i$, which is equivalent to the existence of an integer number k (the number of lots on machine i) such that $kl_i \leq x \leq ku_i$ and $x \leq A$. The latter condition is equivalent to $\lceil x/u_i \rceil \leq \lfloor x/l_i \rfloor$ and $x \leq A$, which can be recognized in a constant time. To calculate $\max\{x \mid x \in \overline{X}_i, c_i(x) \le t\}$, we can first calculate $\max\{x \mid p_i x \le t\} = t/p_i$, then calculate $K_t = \lceil t/(u_i p_i) \rceil$, which is the minimum integer number such that $K_t u_i \ge t/p_i$. After that there are two cases: a) if $t/p_i \in [K_t l_i, K_t u_i]$, then $x_i(t) = t/p_i$ or $x_i(t) = \lceil t/p_i \rceil$ depending on whether the problem is continuous (*cntn*) or discrete (*dscr*), and b) if $t/p_i \notin [K_t l_i, K_t u_i]$, then $x_i(t) = (K_t - 1)u_i$. All these calculations can be performed in a constant time.

The FPTAS in [14] runs in $O\left(\frac{n^2}{\varepsilon^2}\log(U/L) + n^2\log(U/L)\log\log(U/L)\right)$ time, where L and U are lower and upper bounds, respectively, on the optimal solution value. Let us determine these bounds for our special case. The upper bound can be equal to the value of any feasible solution, for example, the best solution in which all items are produced on the same machine i, $i = 1, \ldots, m$: $U = \min_i\{p_i(A + l_i)\}$, because either A is a feasible production volume on machine i, or A falls between two intervals $[kl_i, ku_i]$ and $[(k+1)l_i, (k+1)u_i]$ of feasible production volumes on machine i for some number of lots k. In the latter case, $A + l_i \in [(k+1)l_i, (k+1)u_i]$ and $A + l_i$ is a feasible production volume on machine i. The lower bound L can be established as follows. Let x_i^* be optimal production volume on machine i, $i = 1, \ldots, m$. We have

$$\sum_{i=1}^{m} p_i x_i^* \ge p_{\min} \sum_{i=1}^{m} x_i^* \ge p_{\min} A := L.$$

We deduce that the problem $R|1, lot, \beta|C_\Sigma$ admits an FPTAS with the running time indicated in the statement of the theorem. □

Eremeev et al. [8] studied a modification of the problem $R|1, lot, cntn|C_\Sigma$, in which, instead of the lower and upper bounds on the lot size, a set of *admissible lot size intervals* $[l_i^{(t)}, u_i^{(t)}]$, $t = 1, \ldots, I^{(i)}$, is given for each machine i. An FPTAS with the running time $O((\log\log r + 1/\varepsilon)I_{\max}m^3)$ was presented in [8] for this modification of the problem, where $I_{\max} = \max_{i=1,\ldots,m}\{I^{(i)}\}$ and r is the same as in Theorem 1. An obvious reduction of $R|1, lot, cntn|C_\Sigma$ to the problem with a set of admissible lot size intervals allows to employ the FPTAS from [8] as an approximation algorithm for $R|1, lot, cntn|C_\Sigma$. However, the algorithm from [8] requires the set of admissible lot size intervals to be explicitly presented, and their number is exponential if A/u_i is exponential for some i. Therefore, the approximation algorithm from [8] is exponential for the problem $R|1, lot, cntn|C_\Sigma$. With respect to $1/\varepsilon$, the running time of the FPTAS in Theorem 1 is $\Omega(1/\varepsilon^2)$, while the approximation algorithm from [8] is linear in $1/\varepsilon$.

3.2 Minimizing maximum machine completion time

The problem $R|1, lot, dscr|C_{\max}$ can be solved in *weakly* polynomial time by a modification of the bisection search algorithm $BiSec(GT)$. The required changes are that the initial upper bound is re-set in Step 1 as $UB :=$

$\min_i\{p_i(A+l_i)\}$, the candidate solutions are re-set in Step 2 as $x_i^{(k)} := x_i(t)$, where $t = F_k$, $i = 1, \ldots, m$, and $x_i(t)$ is calculated as it is described in the proof of Theorem 1. We denote the modified algorithm $BiSec(GT)$ as $BiSec(lot,dscr)$.

Proposition 3 *The problem $R|1, lot, dscr|C_{\max}$ can be solved in weakly polynomial time $O(m\log(\min_i\{p_i(A+l_i)\}))$.*

Consider the continuous problem $R|1, lot, cntn|C_{\max}$. Denote by x^* and $C^*_{\max}$ an optimal solution and its makespan value, respectively. Assume that $x^* \in X_0$, where X_0 is a subset of all feasible solutions. If

$$C_{\max}(x^{(1)}) - C_{\max}(x^{(2)}) \geq 1/\Delta \tag{8}$$

for any $x^{(1)} \in X_0$ and $x^{(2)} \in X_0$ with distinct $C_{\max}$ values, and a positive integer number Δ, then the bisection search algorithm $BiSec(lot,dscr)$ can be modified to solve this problem. The modification is that, in Step 2, we understand operator $\lfloor (LB+UB)/2 \rfloor$ as rounding down to the nearest rational number X/Δ, where X is an integer number, and in Step 4, the inequality $UB-LB > 1$ and the equality $UB-LB = 1$ are replaced by $UB-LB > 1/\Delta$ and $UB - LB = 1/\Delta$, respectively. We denote the modified algorithm as $BiSec(lot,cntn)$. The running time of this algorithm is $O(m\log(\Delta\min_i\{p_i(A+l_i)\}))$.

Let us show that the relation (8) is satisfied for a certain set X_0 and

$$\Delta = (\sum_{i=1}^{m} \prod_{j\in\{1,\ldots,m\}, j\neq i} p_j)^2. \tag{9}$$

If $C^*_{\max}$ is integer, then we can define X^0 as the set of all optimal solutions and $\Delta = 1$. Assume that $C^*_{\max}$ is not integer. Consider an optimal solution $x^0 = (x_1^0, \ldots, x_m^0)$. Denote by $I(x^0)$ and $N(x^0)$ the sets of indices of its integer and non-integer components, respectively, $N(x^0) \cup I(x^0) = \{1, \ldots, m\}$. Without loss of generality, we can assume $p_i x_i^0 = C^*_{\max}$ for all $i \in N(x^0)$ because otherwise a non-integer value x_i^0 can be increased up to an integer value or such a value that $p_i x_i^0 = C^*_{\max}$, and the optimality will be maintained. Next, we can also assume that $\sum_{i=1}^m x_i^0 = A$, because otherwise any non-integer value x_i^0 can be decreased down to an integer value or to $A - \sum_{j\neq i} x_i^0$ and the optimality will be maintained. We define X^0 as the set of solutions x^0 such that $p_i x_i^0 = C^*_{\max}$, $i \in N(x^0)$, and $\sum_{i=1}^m x_i^0 = A$. For any $x^0 \in X^0$ we have $x_i^0 = C^*_{\max}/p_i$, $i \in N(x^0)$, and

$$\sum_{i\in N(x^0)} x_i^0 + \sum_{i\in I(x^0)} x_i^0 = C^*_{\max} \sum_{i\in N(x^0)} (1/p_i) + \sum_{i\in I(x^0)} x_i^0 = A,$$

from where it follows that

$$C^*_{\max} = \frac{A - \sum_{i\in I(x^0)} x_i^0}{\sum_{i\in N(x^0)}(1/p_i)} = \frac{\prod_{i\in N(x^0)} p_i(A - \sum_{i\in I(x^0)} x_i^0)}{\sum_{i\in N(x^0)} \prod_{j\in N(x^0), j\neq i} p_j}.$$

We deduce that (8) are satisfied for any $x^{(1)} \in X_0$, $x^{(2)} \in X_0$, and Δ defined in (9). Therefore, the following proposition holds.

Proposition 4 *The problem $R|1, lot, cntn|C_{\max}$ can be solved in weakly polynomial time $O(m \log(\Delta \min_i\{p_i(A + l_i)\}))$, where Δ is defined in (9).*

Existence of a strongly polynomial algorithm for any of the problems $R|1, lot, cntn|C_{\max}$ and $R|1, lot, dscr|C_{\max}$ remains unknown.

4 Conclusion

The computational complexity and algorithmic results obtained in this paper are summarized in Table 1.

Table 1 Computational complexity and algorithmic results

Problem	Complexity	Reference
$R\|1, GT, \beta\|C_{\Sigma}$	Optimal solution NP-hard, $O(m^3/\varepsilon)$	Section 2, [4,9,14]
$R\|1, GT, cntn\|C_{\max}$	$O(m \log m)$	Proposition 1
$R\|1, GT, dscr\|C_{\max}$	$O(m \log \max_i\{u_i p_i\})$	Proposition 2
$R\|1, GT, \beta\|C_{\Sigma}$, $u_i = A$	$O(m)$	Section 2
$R\|1, GT, \beta\|C_{\Sigma}$, $l_i = l$, $u_i = u$	$O(m \log m)$	Section 2
$R\|1, lot, \beta\|C_{\Sigma}$	Optimal solution NP-hard, $O\left(\frac{m^2}{\varepsilon^2} \log r + m^2 \log r \log \log r\right)$	Subsection 3.1 Theorem 1
$R\|1, lot, dscr\|C_{\max}$	$O(m \log(\min_i\{p_i(A + l_i)\}))$	Proposition 3
$R\|1, lot, cntn\|C_{\max}$	$O(m \log(\Delta \min_i\{p_i(A + l_i)\}))$	Proposition 4
$R\|1, lot, \beta\|\gamma$, $u_i \geq 2l_i$	Reduction to $R\|1, GT, \beta\|\gamma$ with $u_i = A$, $i = 1, \ldots, m$	Section 3
$R\|1, lot, cntn\|C_{\Sigma}$, $u_i = A$	$O(m)$	Subsection 3.1
$R\|1, lot, dscr\|C_{\Sigma}$, $l_i = l$, $u_i = u$	$O(m)$	Subsection 3.1

Further research can be undertaken in the following directions:

- establishing computational complexity and developing efficient algorithms for the case of more than one product and other generalizations of the problem $R|1, \alpha, \beta|\gamma$;
- developing strongly polynomial algorithms for the problems $R|1, GT, dscr|C_{\max}$ and $R|1, lot, \beta|C_{\max}$;
- establishing computational complexity and developing efficient algorithms for the problem $R|1, \alpha, \beta|\gamma$, in which $l_i = l$ or $u_i = u$ for all i.

References

1. Allahverdi, A., Ng, C.T., Cheng, T.C.E., Kovalyov, M.Y.: A survey of scheduling problems with setup times or costs. European Journal of Operational Research **187** 985–1032 (2008)

2. Austin, L.M., Hogan., W.W.: Optimizing the procurement of aviation fuels. Management Science **22** (5) 515–527 (1976)
3. Borisovsky, P.A.: A genetic algorithm for one scheduling problem with changeovers. In: Proceedings of XIV Baikal International School-Seminar “Optimization methods and their applications”, Vol. 4, pp. 166-173. Melentiev Energy Systems Institute SB RAS, Irkutsk (2008) (In Russian)
4. Chauhan, S.S., Eremeev, A.V., Romanova, A.A., Servakh, V.V.: Approximation of linear cost supply management problem with lower-bounded demands. In: Proceedings of Discrete Optimization Methods in Production and Logistics, DOM'2004, pp. 16–21. Nasledie Dialog-Sibir Pbs., Omsk (2004)
5. Chauhan, S.S., Eremeev, A.V., Romanova, A.A., Servakh, V.V., Woeginger, G.J.: Approximation of the supply scheduling problem. Oper. Res. Lett. **33** (3) 249–254 (2005)
6. Chen, B., Ye, Y., Zhang, J. Lot-sizing scheduling with batch setup times. Journal of Scheduling. **9** (3), 299–310 (2006)
7. Dolgui, A., Eremeev, A.V., Kovalyov, M.Y., Kuznetsov, P.M.: Multi-product lot sizing and scheduling on unrelated parallel machines. IIE Transactions. **42** (7), 514–524 (2010)
8. Eremeev, A.V., Kovalyov, M.Y., Kuznetsov, P.M.: Approximate solution of the control problem of supplies with many intervals and concave cost functions. Automation and Remote Control. **69** (7), 1181–1187 (2008)
9. Eremeev, A.V., Romanova, A.A., Servakh, V.V., Chauhan, S.S.: Approximation solution of the supply management problem. Journal of Applied and Industrial Mathematics. **1** (4), 433–441 (2007)
10. Garey, M.R., Johnson, D.S.: Computers and intractability. A guide to the theory of NP-completeness. W.H. Freeman and Company, San Francisco (1979)
11. Kovalyov, M.Y., Portmann, M.-C., Oulamara, A. Optimal testing and repairing a failed series system. Journal of Combinatorial Optimization. **12**, 279–95 (2006)
12. Logendran, A., Subur, F.: Unrelated parallel machine scheduling with job splitting. IIE Transactions. **36** (4), 359–372 (2004)
13. Martello, S., Toth, P.: Knapsack problems: algorithms and computer implementations. Wiley (1990).
14. Ng, C.T., Kovalyov, M.Y., Cheng, T.C.E.: An FPTAS for a supply scheduling problem with non-monotone cost functions. Naval Research Logistics. **55**, 194–199 (2008)
15. Papadimitriou, C., Steiglitz, K.: Combinatorial optimization: algorithms and complexity. Prentice-Hall (1982)
16. Potts, C.N., Kovalyov, M.Y.: Scheduling with batching: a review. European Journal of Operational Research. **120**, 228–249 (2000)
17. Potts, C.N., Van Wassenhove, L.N.: Integrating scheduling with batching and lot-sizing: A review of algorithms and complexity. The Journal of the Operational Research Society. **43** (5), 395–406 (1992)
18. Shaik, M.A., Floudas, C.A., Kallrath, J., Pitz, H.-J.: Production scheduling of a large-scale industrial continuous plant: short-term and medium-term scheduling. Computers & Chemical Engineering, **33** (8), 670–686 (2009)
19. Shor, N.Z., Stecuk, P.I.: Piecewise concave problem of knapsack type. In: Mikhalevich, V.S. (ed) Methods of research of extremal problems, pp. 21–28. Institute of Cybernetics n. a. V.M. Glushkov, Kiev (1994) (In Russian)
20. Tahar, D.N., Yalaoui, F., Chu C., Amodeo L. A linear programming approach for identical parallel machine scheduling with job splitting and sequence-dependent setup times. Int. J. Production Economics, **99**, 63–73 (2006)