Journal of Combinatorial Optimization (2022) 43:79–97
https://doi.org/10.1007/s10878-021-00746-w

Minmax common flow-allowance problems with convex resource allocation and position-dependent workloads

Baruch Mor[1]

Accepted: 10 April 2021 / Published online: 29 April 2021

Abstract

We study *minmax* due-date based on *common flow-allowance* assignment and scheduling problems on a single machine, and extend known results in scheduling theory by considering *convex resource allocation*. The total cost function of a given job consists of its earliness, tardiness and flow-allowance cost components. Thus, the common flow-allowance and the actual jobs' processing times are decision variables, implying that the due-dates and actual processing times can be controlled by allocating additional resource to the job operations. Consequently, our goal is to optimize a cost function by seeking the optimal job sequence, the optimal job-dependent due-dates along with the actual processing times. In all addressed problems we aim to minimize the maximal cost among all the jobs subject to a constraint on the resource consumption. We start by analyzing and solving the problem with *position-independent* workloads and then proceed to *position-dependent* workloads. Finally, the results are generalized to the method of *common due-window*. For all studied problems closed form solutions are provided, leading to polynomial time solutions.

Keywords Single machine scheduling · Minmax · Common flow-allowance · Convex resource allocation · Position-dependent workloads

1 Introduction

In this paper, we address and combine two popular topics in scheduling theory: *due-date assignment* problems and controllable processing times. One influential effect of the Just-In-Time (JIT) production methodology on scheduling theory was the vast research on due-date assignment and scheduling problems. In this class of problems, the due-dates are decision variables and we seek to balance the inherent trade-off

✉ Baruch Mor
baruchm@ariel.ac.il

[1] Department of Economics and Business Administration, Ariel University, 40700 Ariel, Israel

between early competitive due-dates on the one hand and late and, consequently, undemanding due-dates on the other. Three significant models were introduced: (i) CON (Common)—where all the jobs share a common due-date (Panwalkar et al. 1982); (ii) DIF (Different)—the unrestricted due date assignment method, in which each job can be assigned a different due date with no restrictions (Seidmann et al. 1981); and (iii) SLK (Slack)—where the job-dependent due-dates are often linear functions of the job processing times (Adamopoulos and Pappis 1996).

The vast majority of studies addressed the *minsum* versions, i.e., minimizing the total cost of earliness, tardiness and due-date of all jobs. These objective functions were studied with various machine environments, position-independent/dependent job processing times, constant/controllable job processing times, problems involving optional job-rejection, and with single/multi-agent settings. Recent articles include, among others, Wang et al. (2016), Yin et al. (2016), Gerstl et al. (2017), Mor (2017, 2019a), Mor and Mosheiov (2017), Cheng and Cheng (2018), Gao et al. (2018), Xiong et al. (2018), Ji et al. (2019), Geng et al. (2019), Liu et al. (2020), Liu and Jiang (2020) and Mor et al. (2020).

In this study, we combine very important classes of scheduling theory—*due-date assignment* and *scheduling problems* and *convex resource allocation*, and provide *optimal algorithm solutions* for very practical problems. We concentrate on the SLK model, i.e., due-date assignments based on *common flow-allowance*. Unlike minsum, the *minmax* version has not received appropriate research attention. In minmax problems, we aim to minimize the maximal cost among all jobs. The importance of these problems is threefold: (i) realize a value that demonstrates the utilization of the production system that can be compared to the competition; (ii) estimate the worst case performance of the production system; and (iii) treat all clients in an equal way. Typically, in minmax scheduling problems with common flow-allowance there are three cost components: earliness, tardiness, and flow-allowance. The first to consider single machine minmax scheduling and assignment problems with due-date based on flow-allowance were Mor and Mosheiov (2012a). Later, Mor and Mosheiov (2012b, 2016, 2017) extended the fundamental setting to identical parallel machines, proportionate flow-shop, and two competing agents, respectively.

In this paper, we combine the setting of due-date based on common flow-allowance and the very important and practical model of controllable processing times. Scheduling models with controllable processing times are very widespread and influential. Unlike most studies in scheduling theory that presume constant job processing times, research with controllable processing times assumes that the processing times may be controlled through allocating extended resources to job operations. Thus, the updated approach aims to address various real-life problems where the improved deployment of inputs in the production system is achieved by allocating added resource to the job processing. There are two dominant methodologies when addressing controllable processing times: (i) actual job processing times as a linear function of the amount of resource allocated to the processing of the job and (ii) actual job processing times as a convex decreasing function of the amount of resource allocated to the processing of the job. In their wide-ranging survey, Shabtay and Steiner (2007) state that "For many resource allocation problems in physical or economic systems, however, they do not use a linear resource consumption function, since it fails to reflect the law of diminishing marginal returns. This law states that productivity increases at a decreasing rate with the amount of resource employed." Thus, in this paper, we assume that the job processing time is a convex decreasing function of the amount of resource allocated to the processing of the job. Recent articles with convex resource allocation include Liu et al. (2016, 2017), Sun et al. (2016), Wang et al. (2017) and Sun et al. (2019).

Many scheduling researchers have studied scheduling problems with variable job processing times, i.e., starting-time-dependent and/or position-dependent. The importance of this method is reflected in Gawiejnowicz (2008) with numerous variable processing time problems, different objective functions, and diverse machine settings. Other articles include Mor and Mosheiov (2011, 2012c, 2018), Rudek (2012), Sun et al. (2013), Agnetis et al. (2014), Agnetis and Mosheiov (2017), Gerstl et al. (2017), Pei et al. (2019) and Zhang et al. (2018). In general, the extension to variable job processing times significantly increases the computational complexity of the problems, whether the processing times are (i) restricted to monotone functions of the job starting times; (ii) position-dependent, i.e., representing aging or learning effects; and (iii) specific functions, e.g., linear or exponential.

Oron (2016) was the first to introduce the realistic and challenging generalization of *position-independent workloads* to *position-dependent workloads*. The author restricted the functional form of the workloads to reflect learning effect and studied minimizing makespan and total flowtime on a single machine, as well as minimizing total flowtime on parallel machines. Lu and Liu (2018) analyzed bi-criteria problems, where the first criterion is a scheduling measure and the second criterion is the total resource consumption cost. Mor (2019b) considered single machine minmax common due-window assignment and scheduling problems with convex resource allocation. Recently, Mor (2020) extended the results of Oron (2016) by omitting the restriction on the functional form of the workloads, and provided a unified approach for single machine scheduling with position-dependent workloads and positional penalties.

Following Mor and Mosheiov (2012a), we focus on the minmax problem with due-date based on flow-allowance on a single machine, and extend their results by considering controllable job processing times. Firstly, we study *position-independent workloads*, and then proceed to the model of *position-dependent workloads*. Secondly, we generalize the due-date method to the due-window method. For the problems with position-independent workloads, we provide $O(n)$ procedure solutions, whereas for the problems with position-dependent workloads, we introduce solution procedures with computational effort of $O(n^3)$. Thus, in combining the minmax approach and controllable processing times, this study addresses theoretical and practical aspects in scheduling theory.

In Sect. 2, we provide the formulation. Sections 3 and 4 are dedicated to the setting of common due-date with position-independent and position-dependent workloads, respectively. In Sects. 5 and 6, we generalize the results of Sects. 3 and 4 to the method of common due-window, respectively. In Sect. 7, we provide conclusions and future research topics.

2 Formulation

A set of jobs $\mathcal{J}$ needs to be processed on a single machine. All the jobs are available at time zero, and preemption is not allowed. The actual processing time of the job allocated to position $j, j \in \mathcal{J}$ is defined as $p_j = \left(\frac{w_j}{u_j}\right)^k$, where w_j is a positive value reflecting the workload of the processing operation, u_j is a decision variable demonstrating the amount of resource allocated to the processing operation, and k is a constant positive parameter. We also denote by w_{max} the largest workload among all jobs, $w_{max} = \max_{j \in \mathcal{J}}\{w_j\}$.

We assume a common flow-allowance denoted by q, which is a decision variable. The due-date (DD) of job j is d_j and defined as its processing time plus the job-independent constant q:

$$d_j = p_j + q, j \in \mathcal{J}.$$

For a given schedule, the completion time of job $j, j \in \mathcal{J}$ is denoted by C_j. We denote the makespan, i.e., the completion time of the last job to leave the system by C_{max}. Jobs completed before/after the due-date are considered as early/tardy jobs and incur penalties. The earliness of the job j is $E_j = \max\{0, d_j - C_j\}$ and the tardiness of job j is defined as $T_j = \max\{0, C_j - d_j\}, j \in \mathcal{J}$. There are three cost components: for earliness, tardiness, and to motivate the scheduler to set competitive due-dates there is an additional cost for the common flow-allowance. Let α denote the unit earliness cost, β denote the unit tardiness cost, and γ denote the cost of delaying the due-dates by one time unit. We note that in a given schedule a specific job is either early or tardy, implying that its cost is $\max\{\alpha E_j + \gamma q, \beta T_j + \gamma q\} = \max\{\alpha E_j, \beta T_j\} + \gamma q$.

We seek to find the optimal sequence and flow-allowance, such that the maximal cost among all jobs is minimized. Formally, the scheduling measure in this case is $Z = \max_{j \in \mathcal{J}}\{\max\{\alpha E_j, \beta T_j\} + \gamma q^1 + \delta D\}$.

First, we focus on minimizing the scheduling measure, given that the total resource consumption cannot exceed an upper bound denoted by U. We denote the problem by $\boldsymbol{P1}$ and, utilizing the three-field notation introduced by Graham et al. (1979), the first problem studied is,

$$\boldsymbol{P1}: \ 1\left|SLK\,DD, conv, w_j, \sum u_j \leq U\right|Z.$$

Next, we extend the setting of common flow-allowance with position-independent workloads to the setting of position-dependent workloads, w_{jr}, implying that the workload of job j if assigned to position r is given by $p_{jr} = \left(\frac{w_{jr}}{u_j}\right)^k, j, r = 1, \ldots, n.$

Thus, the second problem, denoted by $\boldsymbol{P2}$, is:

$$\boldsymbol{P2}: \ 1\left|SLK\,DD, conv, w_{jr}, \sum u_j \leq U\right|Z.$$

Finally, we generalize the above results and address the method of due-window (DW) based on common flow-allowance. Thus, jobs that are completed within a time interval (a due-window) are not penalized, whereas jobs that are completed prior to or after the window, are penalized according to their earliness/tardiness. In due-window assignment problems, we seek simultaneously the start time and the size of the window. We address job-dependent due-windows based on common flow-allowance. The job-dependent due-window is denoted by D_j, such that the window of job $j, j \in \mathcal{J}$ is given by $D_j = [d_j^1, d_j^2]$, where

$$d_j^1 = p_j + q^1, j \in \mathcal{J}.$$
$$d_j^2 = p_j + q^2, j \in \mathcal{J}.$$

Similar to the due-date method, the common flow-allowance constants q^1 and q^2 are decision variables, such that $q^2 \geq q^1$. Since $D_j = d_j^2 - d_j^1 = p_j + q^2 - \left(p_j + q^1\right) = q^2 - q^1 = D, j \in \mathcal{J}$, the size of the due-window is identical for all jobs. The earliness of j is $E_j = \max\left\{0, d_j^1 - C_j\right\}$ and the tardiness of job j is $T_j = \max\left\{0, C_j - d_j^2\right\}, j \in \mathcal{J}$. To motivate the scheduler to set reduced due-window size, there are four unit costs: for earliness, tardiness, the cost of the first flow-allowance, and extending the size of the due-window. Let α denote the unit earliness cost, β denote the unit tardiness cost, γ denote the unit cost of the flow-allowance (q^1), and δ denote the unit cost of extending the size of the due-window (D). Our aim is to find the optimal common constants and the optimal size of the due-window, as well as the sequence, that minimize the maximal cost among all jobs. Formally, the scheduling measure in this case is $Z = \max_{j\in\mathcal{J}}\left\{\max\left\{\alpha E_j + \gamma q^1 + \delta D, \beta T_j + \gamma q^1 + \delta D\right\}\right\} = \max_{j\in\mathcal{J}}\left\{\max\left\{\alpha E_j, \beta T_j\right\} + \gamma q^1 + \delta D\right\}$.

The third and fourth problems, denoted by ***P*3** and ***P*4**, respectively, are:

$$\boldsymbol{P3} : 1\left|SLK\,DW, conv, w_j, \sum u_j \leq U\right|Z,$$
$$\boldsymbol{P4} : 1\left|SLK\,DW, conv, w_{jr}, \sum u_j \leq U\right|Z.$$

3 $\boldsymbol{P1 : 1\left|SLK\,DD, conv, w_j, \sum u_j \leq U\right|Z}$

Mor and Mosheiov (2012a) proved that the same problem with constant job processing time, $1|SLK\,DD|Z$, can be formulated as the following Linear Program (LP), denoted by LP3.1:

$$LP3.1$$
$$\min Z$$
$$\text{s.t.}$$
$$\begin{aligned} &Z \geq aE_1 + \gamma q = q(\alpha + \gamma),\\ &Z \geq \beta T_n + \gamma q = q(\gamma - \beta) + \beta\left(C_{max} - p_n\right),\\ &Z, q \geq 0. \end{aligned}$$

In this linear program, the subscript is the index of the job assigned to the jth position in a given sequence. Analyzing LP3.1, the authors concluded that an optimal schedule exists in which the job with the largest processing time is scheduled last and the sequence of the remaining $n-1$ jobs is immaterial.

Based on the above, the authors provided a closed form solution, where there are only two possible cases, and the optimal flow-allowance and cost function are given by:

Case 1: $\gamma > \beta$

$$q^* = 0 \text{ and } Z^* = \beta\left(C_{max} - p_n\right) = \beta \sum_{j=1}^{n-1} p_j.$$

Case 2: $\gamma \leq \beta$

$$q^* = \frac{\beta}{\alpha+\beta}\left(C_{max} - p_n\right) = \frac{\beta}{\alpha+\beta}\sum_{j=1}^{n-1} p_j \text{ and } Z^* = (\alpha+\gamma)q^* = \frac{\beta(\alpha+\gamma)}{\alpha+\beta}\sum_{j=1}^{n-1} p_j.$$

In the context of this study, the actual processing time of each job is a function of the amount of resource allocated to it, i.e., $p_j = \left(\frac{w_j}{u_j}\right)^k$. Therefore, and following the analysis of Cases 1–2 above, we conclude that the job with the largest workload should be scheduled last and, consequently, does not contribute to the cost function. Moreover, since the coefficients in both cases are constants, it suffices to minimize the term $\sum_{j=1}^{n-1} p_j = \sum_{j=1}^{n-1} \left(\frac{w_j}{u_j}\right)^k$.

We conclude that the following observations are adequate:

Observation 3.1 *An optimal schedule exists in which the job with the largest workload is scheduled last and the sequence of the remaining $n-1$ jobs is immaterial.*

Observation 3.2 *An optimal schedule exists in which no resource is allocated to the job scheduled last.*

From Observations 3.1–3.2, we conclude that $p_n = w_{max}$.

Kaspi and Shabtay (2004) proved the following properties for an optimal solution of the problem $1/w_j, \sum u_j \leq U/C_{max}$:

- The entire available resource is exploited.
- The optimal resource allocated to job $j, j \in \mathcal{J}$ is given by $u_j = U\frac{w_j^{\frac{k}{k+1}}}{\sum_{j\in\mathcal{J}} w_j^{\frac{k}{k+1}}}$.
- The actual processing time of job $j, j \in \mathcal{J}$ is given by $p_j = \left(\frac{\sum_{j=1}^{n} w_j^{\frac{k}{k+1}}}{U}\right)^k w_j^{\frac{k}{k+1}}$.

Combining Observation 3.1–3.2 and the results of Kaspi and Shabtay (2004), we can summarize the properties of an optimal solution for ***P*1**:

Property 3.1 *The entire available resource is consumed to reduce the processing time of jobs* $j = 1, \dots, n - 1$.

For ease of reading, we define a constant W, such that $W = \sum_{j=1}^{n-1} w_j^{\frac{k}{k+1}}$.

Property 3.2 *The optimal resource allocated to job* j,$j = 1, \dots, n - 1$ *is given by* $u_j = \frac{U}{W} w_j^{\frac{k}{k+1}}$.

Property 3.3 *The actual processing time of job* j,$j = 1, \dots, n - 1$ *is given by* $p_j = \left(\frac{W}{U}\right)^k w_j^{\frac{k}{k+1}}$.

Property 3.4 *The optimal flow-allowance and cost function for* ***P*1** *are given by*

Case 1: $\gamma > \beta$

$$q^* = 0 \text{ and } Z^* = \beta\left(\frac{W}{U}\right)^k \sum_{j=1}^{n-1} w_j^{\frac{k}{k+1}}.$$

Case 2: $\gamma \leq \beta$

$$q^* = \frac{\beta}{\alpha + \beta}\left(\frac{W}{U}\right)^k \sum_{j=1}^{n-1} w_j^{\frac{k}{k+1}} \text{ and } Z^* = \frac{\beta(\alpha + \gamma)}{\alpha + \beta}\left(\frac{W}{U}\right)^k \sum_{j=1}^{n-1} w_j^{\frac{k}{k+1}}.$$

Next, we present the formal solution procedure:

Solution procedure to problem ***P*1**

Step 1 Find the job with the largest workload and assign it to the last position.

Step 2 Use Property 3.2 to calculate the amount of resource allocated to jobs $1, \dots, n - 1$.

Step 3 Use Property 3.3 to calculate the actual processing times to jobs $1, \dots, n - 1$.

Step 4 Use Property 3.4 to determine the relevant case and calculate the optimal common flow-allowance and cost.

Theorem 3.1 *The computational complexity of the proposed solution is* $O(n)$ *time*.

Proof Step 1: Determining the job with the largest workload requires $O(n)$ time. Steps 2–3: Computing the amount of resource allocated to each job and the actual processing time of each job, respectively, is done in $O(n)$. Step 4: Calculating the optimal common flow-allowance and cost is done in $O(n)$. Hence we conclude that the computational complexity is $O(n)$. □

Numerical example 3.1 We solve the following 8-job problem:

The jobs workloads are: $w_j = (22, 24, 49, 50, 21, 14, 14, 6)$.

The cost parameters are: $\alpha = 8$, $\beta = 10$ and $\gamma = 5$.

The available total resource is, $U = 12$ and $k = 0.5$.

Since $w_{max} = 50$, switching the positions of jobs 4 and 8, we obtain one optimal sequence $(1, 2, 3, 8, 5, 6, 7, 4)$.

Solution:

Since $\gamma \le \beta$, the relevant case is Case 2.

The amount of resource allocated to each job is:

$$u_j = (1.794, 1.847, 2.343, 1.163, 1.766, 1.543, 1.543, 0).$$

The actual processing times are:

$$p_j = (3.502, 3.605, 4.573, 2.271, 3.448, 3.012, 3.012, 50).$$

The optimal common flow-allowance is $q^* = 13.013$, implying that the due-dates are:

$$d_j = (16.515, 16.618, 17.586, 15.284, 16.461, 16.025, 16.025, 63.013).$$

The optimal cost is: $Z^* = 169.165$.

4 $\boldsymbol{P2: 1\left|SLK\ DD, conv, w_{jr}, \sum u_j \le U\right|Z}$

In this section, we extend the due-date setting by considering *position-dependent workloads*. The solution of Problem $\boldsymbol{P2}$ requires us to establish the optimal job sequence and then to evaluate the effect of allocating resource to each job on the optimal due-date and cost function. Let w_{jr} denote the workload of job j if processed in position r, $j, r = 1, \dots n$. Thus, the input for the problem contains three cost parameters, a matrix of $n \times n$ job-position workloads, the upper bound on the total amount of resource consumption and the constant parameter. Consequently, the flow allowance and the derived due-dates are defined for any given realization of a job sequence, and we have to simultaneously seek the optimal job sequence and the optimal flow allowance. We conclude that for any given job sequence, Observations 3.1–3.2 and Properties 3.1–3.4 are still valid.

Oron (2016) solved the problem, $1/w_{jr}, conv, \sum_{j \in \mathcal{J}} u_j \le U/C_{max}$ and proved that the objective function can be rewritten as:

$$f = \sum_{j=1}^{n} \sum_{r=1}^{n} \left(\frac{w_{jr}}{u_j} \right)^k X_{jr} = \frac{1}{U^k} \left(\sum_{j=1}^{n} \left(\sum_{r=1}^{n} w_{jr}^k X_{jr} \right)^{\frac{1}{1+k}} \right)^{1+k}.$$

Since $U, k > 0$, the author proved that minimizing f is equal to minimizing $\left(\sum_{r=1}^{n} w_{jr}^k X_{jr} \right)^{\frac{1}{1+k}}$.

Subsequently, the author proved that obtaining the optimal job sequence is reduced to solving the following linear Assignment Problem (AP):

$$\min \sum_{j=1}^{n}\sum_{r=1}^{n} w_{jr}^{\frac{k}{1+k}} X_{jr}$$

$$s.t.$$

$$\begin{aligned}
&\sum_{j=1}^{n} X_{jr} = 1, \quad r = 1,\dots,n,\\
&\sum_{r=1}^{n} X_{jr} = 1, \quad j = 1,\dots,n,\\
&X_{jr}, \quad binary,\ j = 1,\dots,n, \quad r = 1,\dots,n.
\end{aligned}$$

where

$$X_{jr} = \begin{cases} 1, & \text{if job } j \text{ is assigned to position } r \\ 0, & \text{otherwise} \end{cases}.$$

After the optimal sequence is realized, the makespan can be calculated based on the effect of the additional resource allocation on the actual processing times.

From Property 3.4, the coefficient in the cost function is given by $\rho = \begin{cases} \beta, & Case\,1 \\ \frac{\beta(\alpha+\gamma)}{\alpha+\beta}, & Case\,2 \end{cases}$.

Since, ρ is a *job-independent* and *position-independent* constant, then the cost function can be formulated as follows,

$$Z = \rho \sum_{j=1}^{n}\sum_{r=1}^{n-1} \left(\frac{w_{jr}}{u_j}\right)^{k} X_{jr}.$$

Thus, to minimize the cost function it suffices to minimize the term $\sum_{j=1}^{n}\sum_{r=1}^{n-1}\left(\frac{w_{jr}}{u_j}\right)^{k} X_{jr}$.

Moreover, the latter implies that to solve Problem $\boldsymbol{P}2$, we can use the assignment problem provided by Oron (2016) with a minor modification. Specifically, to obtain the optimal job sequence we need to solve the following special form $n \times (n-1)$ AP (denoted AP4.1):

$$AP4.1$$

$$\min \sum_{j=1}^{n}\sum_{r=1}^{n-1} \left(\frac{w_{jr}}{u_j}\right)^{k} X_{jr}$$

$$s.t.$$

$$\begin{aligned}
&\sum_{j=1}^{n} X_{jr} = 1, \quad r = 1,\dots,n-1,\\
&\sum_{r=1}^{n-1} X_{jr} \le 1, \quad j = 1,\dots,n,\\
&X_{jr}, \quad binary,\ j = 1,\dots,n,\ r = 1,\dots,n-1.
\end{aligned}$$

where

$$X_{jr} = \begin{cases} 1, & \text{if job } j \text{ is assigned to position } r \\ 0, & \text{otherwise} \end{cases}.$$

Now, we are ready to present the formal solution procedure.

Solution procedure to problem ***P2***:

Step 1 Apply AP4.1 to obtain the optimal sequence of the jobs assigned to positions $1, \dots, n-1$.

Step 2 Schedule the unassigned job to position n.

Step 3 Use Property 3.2 to calculate the amount of resource allocated to jobs $1, \dots, n-1$.

Step 4 Use Property 3.3 to calculate the actual processing times to jobs $1, \dots, n-1$.

Step 5 Use Property 3.4 to determine the relevant case and calculate the optimal common flow-allowance and cost.

Theorem 4.1 *The proposed solution solves* ***P2*** *in* $O(n^3)$ *time.*

Proof Step 1: The computational complexity required for solving the assignment problem is known to be $O(n^3)$. Step 2: Scheduling the unassigned job to position n is done in constant time. Steps 3–4: Computing the amount of resource allocated to each job and the actual processing time of each job, respectively, is done in $O(n)$. Step 5: The calculation of the optimal cost and due-dates is done in $O(n)$. Thus, the total complexity of the solution procedure is dominated by Step 1, and is $O(n^3)$. □

Numerical example 4.1 We solve the 8×8 job-position workloads problem given in Table 1, where the cost parameters are: $\alpha = 2$, $\beta = 10$, and $\gamma = 15$.

The available total resource is $U = 14$ and $k = 1$.

Solution:

Applying AP4.1, we obtain an optimal sequence $(6, 2, 7, 3, 5, 8, 4, 1)$.

Since $\gamma > \beta$ the relevant case is Case 1.

The amount of resource allocated to each job is:

$$u_j = (1.865, 1.865, 1.668, 0.834, 4.086, 1.180, 2.502, 0).$$

The actual processing times are:

$$p_j = (2.681, 2.681, 2.398, 1.199, 5.874, 1.696, 3.597, 47.000).$$

The optimal common flow-allowance is $q^* = 0$, implying that $d_j = p_j, j = 1, \dots, n$

.

The optimal cost is: $Z^* = 201.248$.

Table 1 Example 2—job-position workloads

Job	Position							
	1	2	3	4	5	6	7	8
1	28	10	14	44	42	13	35	47
2	30	5	36	19	43	35	25	36
3	29	17	17	1	21	6	31	20
4	44	21	38	18	34	38	9	22
5	39	50	47	14	24	38	29	39
6	5	36	15	29	49	47	35	20
7	19	44	4	8	44	35	17	50
8	3	16	32	18	46	2	47	9

5 $P3: 1\left|SLKDW, conv, w_j, \sum u_j \leq U\right|Z$

In this section, we generalize the results of the due-date method (Sect. 3) to the method of due-window. Mor and Mosheiov (2012a) proved that the problem with constant processing times can be formulated by the following LP, denoted LP5.1:

$$
\begin{aligned}
&LP5.1\\
&\min Z\\
&\text{s.t.}\\
&\quad Z \geq \alpha E_1 + \gamma q^1 + \delta\left(q^2 - q^1\right) = q^1(\alpha + \gamma - \delta) + q^2\delta,\\
&\quad Z \geq \beta T_n + \gamma q^1 + \delta\left(q^2 - q^1\right) = q^1(\gamma - \delta) + q^2(\delta - \beta) + \beta\left(C_{max} - p_n\right),\\
&\quad q^2 \geq q^1,\\
&\quad Z, q^1, q^2 \geq 0.
\end{aligned}
$$

Examining LP5.1, the authors observed that similar to the due-date case, the job sequence is immaterial, except for the last job, which must have the largest processing time. Furthermore they proved several properties of an optimal solution and concluded that there are exactly four candidates for the optimal values of q^1 and q^2:

Case 1: The window has the maximum possible size.

$$q^1 = 0, \quad q^2 = C_{max} - p_n \text{ and } Z = \delta q^2.$$

Case 2: The window is reduced to a due-date.

In this case, $q^1 = q^2 = q$, and three sub-cases must be solved and compared.

Sub-Case 2.1:

$$q^1 = q^2 = 0, \quad Z = \beta\left(C_{max} - p_n\right).$$

Sub-Case 2.2:

$$q^1 = q^2 = q = \beta\frac{C_{max} - P_{max}}{\alpha + \beta} \quad \text{and} \quad Z = \beta\frac{\alpha + \gamma}{\alpha + \beta}\left(C_{max} - P_{max}\right).$$

Sub-Case 2.3:

$$q^1 = q^2 = q = C_{max} - p_n \quad \text{and} \quad Z = (\alpha + \gamma)\left(C_{max} - P_{max}\right).$$

Property 5.1 *The optimal cost function is given by* $Z = min\left\{\delta, \beta, \beta\frac{\alpha+\gamma}{\alpha+\beta}, \alpha + \gamma\right\}(C_{max} - P_{max})$.

Based on the above, we conclude that Observations 3.1–3.2 and Properties 3.1–3.3 and 5.1, still hold for Problem ***P***3, and update the above results accordingly:

Property 5.2 *The optional values of* q^1 and q^2 *and cost function for* ***P*3** *are given by*,

Case 1: The window has the maximum possible size.

$$q^1 = 0, \ \ q^2 = \left(\frac{W}{U}\right)^k \sum_{j=1}^{n-1} w_j^{\frac{k}{k+1}} \ \text{and} \ Z = \delta q^2 = \delta\left(\frac{W}{U}\right)^k \sum_{j=1}^{n-1} w_j^{\frac{k}{k+1}}.$$

Case 2: The window is reduced to a due-date.

In this case, $q^1 = q^2 = q$*, and we have to solve and compare three sub-cases.*

Sub-Case 2.1:

$$q^1 = q^2 = 0, \quad Z = \beta\left(\frac{W}{U}\right)^k \sum_{j=1}^{n-1} w_j^{\frac{k}{k+1}}.$$

Sub-Case 2.2:

$$q^1 = q^2 = q = \frac{\beta}{\alpha+\beta}\left(\frac{W}{U}\right)^k \sum_{j=1}^{n-1} w_j^{\frac{k}{k+1}} \quad \text{and} \quad Z = (\alpha+\gamma)q = \beta\frac{\alpha+\gamma}{\alpha+\beta}\left(\frac{W}{U}\right)^k \sum\nolimits_{j=1}^{n-1} w_j^{\frac{k}{k+1}}.$$

Sub-Case 2.3:

$$q^1 = q^2 = q = \left(\frac{W}{U}\right)^k \sum_{j=1}^{n-1} w_j^{\frac{k}{k+1}} \quad \text{and} \quad Z = (\alpha+\gamma)q = (\alpha+\gamma)\left(\frac{W}{U}\right)^k \sum_{j=1}^{n-1} w_j^{\frac{k}{k+1}}.$$

Property 5.3 *The optimal cost function is given by* $Z = min\left\{\delta, \beta, \beta\frac{\alpha+\gamma}{\alpha+\beta}, \alpha+\gamma\right\}\left(\frac{W}{U}\right)^k \sum_{j=1}^{n-1} w_j^{\frac{k}{k+1}}$.

Solution procedure to problem ***P3***

Step 1 Find the job with the largest workload and assign it to the last position.

Step 2 Use Property 3.2 to calculate the amount of resource allocated to jobs $1, \dots, n-1$.

Step 3 Use Property 3.3 to calculate the actual processing times to jobs $1, \dots, n-1$.

Step 4 Use Properties 5.2 and 5.3 to determine the relevant case and to calculate the optimal values of the flow-allowance constants and cost.

Theorem 5.1 *The computational complexity of the proposed solution is $O(n)$ time.*

The proof is identical to that given in Sect. 3 for Theorem 3.1 and is thus omitted.

Numerical example 5.1 Consider the following data:

The number of jobs is $n = 8$, the total resource limitation is $U = 13$ and $k = 0.75$.

The workloads are $(33, 22, 44, 6, 49, 41, 33, 2)$.

We solve the given input with four different sets of cost parameters to reflect Case 1 and Subcases 2.1–2.3 (see Property 5.1).

Solution:

The optimal amount of resource allocated to each job, the actual processing time of each job, and the optimal flow-allowance values, due-dates and cost are provided in Table 2.

Table 2 Numerical example 5.1—resource allocation, actual processing times, and the optimal flow-allowance values, due-dates and cost

Job	1	2	3	4	8	6	7	5
Case 1: $\alpha = 5, \beta = 15, \gamma = 9, \delta = 8$								
u_j	2.222	1.867	2.513	1.070	0.668	2.438	2.222	0
p_j	7.566	6.359	8.559	3.644	2.276	8.304	7.566	49.000
q_1^*	0.000							
q_2^*	44.275							
d_1^*	7.566	6.359	8.559	3.644	2.276	8.304	7.566	49.000
d_2^*	51.841	50.634	52.834	47.919	46.550	52.579	51.841	93.275
Z^*	354.197							
Job	1	2	3	4	8	6	7	5
Case 2, Subcase 2.1: $\alpha = 8, \beta = 5, \gamma = 9, \delta = 11$								
u_j	2.222	1.867	2.513	1.070	0.668	2.438	2.222	0
p_j	7.566	6.359	8.559	3.644	2.276	8.304	7.566	49.000
$q_1^* = q_2^*$	0.000							
$d_1^* = d_2^*$	7.566	6.359	8.559	3.644	2.276	8.304	7.566	49.000
Z^*	221.373							
Job	1	2	3	4	8	6	7	5
Case 2, Subcase 2.2: $\alpha = 15, \beta = 8, \gamma = 5, \delta = 9$								
u_j	2.222	1.867	2.513	1.070	0.668	2.438	2.222	0
p_j	7.566	6.359	8.559	3.644	2.276	8.304	7.566	49.000
$q_1^* = q_2^*$	15.400							
$d_1^* = d_2^*$	22.966	21.759	23.959	19.044	17.676	23.704	22.966	64.400
Z^*	307.998							
Job	1	2	3	4	8	6	7	5
Case 2, Subcase 2.3: $\alpha = 0, \beta = 9, \gamma = 5, \delta = 9$								
u_j	2.222	1.867	2.513	1.070	0.668	2.438	2.222	0
p_j	7.566	6.359	8.559	3.644	2.276	8.304	7.566	49.000
$q_1^* = q_2^*$	44.275							
$d_1^* = d_2^*$	51.841	50.634	52.834	47.919	46.550	52.579	51.841	93.275
Z^*	221.373							

6 $P4: 1\left|SLK\ DW, conv, w_{jr}, \sum u_j \leq U\right|Z$

To present a solution for Problem ***P*4**, we combine the results achieved in Sects. 4 and 5. The input for the problem contains four unit costs, a matrix of $n \times n$ job-position workloads, and an upper bound on the total resource.

From Property 5.1, the coefficient in the cost function is

$$\rho = \begin{cases} \delta, & Case\,1 \\ \beta, & Subcase\,2.1 \\ \frac{\beta(\alpha+\gamma)}{\alpha+\beta}, & Subcase\,2.2 \\ \alpha+\gamma, & Subcase\,2.3 \end{cases}$$

Similar to the due-date setting, ρ is a *job-* and *position-independent* constant and the cost function can be further simplified,

$$Z = \rho \sum_{j=1}^{n} \sum_{r=1}^{n-1} \left(\frac{w_{jr}}{u_j} \right)^k X_{jr}.$$

Thus, to solve Problem ***P*4**, we can utilize AP4.1, presented in Sect. 4 for solving Problem ***P*2**, and the formal solution procedure is as follows.

Solution procedure to problem **P4**

Step 1 Apply AP4.1, to obtain the optimal sequence of the jobs assigned to positions $1, \ldots, n-1$.

Step 2 Schedule the unassigned job to position n.

Step 3 Use Property 3.2 to calculate the amount of resource allocated to jobs $1, \ldots, n-1$.

Step 4 Use Property 3.3 to calculate the actual processing times to jobs $1, \ldots, n-1$.

Step 5 Use Properties 5.2 and 5.3 to determine the relevant case and to calculate the optimal values of the flow-allowance constants and cost.

Theorem 6.1 *The proposed solution solves* **P4** *in* $O(n^3)$ *time.*

The proof is exactly like the one given in Sect. 4 for Theorem 4.1.

Numerical example 6.1 We solve the 8×8 job-position processing times problem given in Table 3, where $U = 16$ and $k = 0.6$. As in Numerical example 4.1, we solve this input with four different sets of cost parameters.

Table 3 Example 6.1—job-position workloads

Job	Position							
	1	2	3	4	5	6	7	8
1	13	12	1	27	4	35	19	40
2	11	33	22	47	34	46	3	8
3	44	46	44	17	24	42	48	25
4	49	3	48	23	30	8	37	46
5	14	27	15	36	40	25	31	49
6	2	7	45	7	20	46	36	28
7	47	24	37	23	5	32	37	31
8	43	32	44	25	35	39	1	24

Table 4 Numerical example 6.1—resource allocation, actual processing times, and the optimal flow-allowance values, due-dates and cost

Job	6	4	1	3	7	5	8	2
Case 1: $\alpha = 5, \beta = 17, \gamma = 10, \delta = 9$								
u_j	1.612	1.877	1.243	3.596	2.273	4.156	1.243	0
p_j	1.138	1.325	0.878	2.539	1.605	2.935	0.878	8.000
q_1^*	0.000							
q_2^*	11.298							
d_1^*	1.138	1.325	0.878	2.539	1.605	2.935	0.878	8.000
d_2^*	12.436	12.623	12.175	13.837	12.902	14.232	12.175	19.298
Z^*	101.678							
Job	6	4	1	3	7	5	8	2
Case 2, Subcase 2.1: $\alpha = 9, \beta = 6, \gamma = 10, \delta = 17$								
u_j	1.612	1.877	1.243	3.596	2.273	4.156	1.243	0
p_j	1.138	1.325	0.878	2.539	1.605	2.935	0.878	8.000
$q_1^* = q_2^*$	0.000							
$d_1^* = d_2^*$	1.138	1.325	0.878	2.539	1.605	2.935	0.878	8.000
Z^*	67.785							
Job	6	4	1	3	7	5	8	2
Case 2, Subcase 2.2: $\alpha = 17, \beta = 9, \gamma = 5, \delta = 10$								
u_j	1.612	1.877	1.243	3.596	2.273	4.156	1.243	0
p_j	1.138	1.325	0.878	2.539	1.605	2.935	0.878	8.000
$q_1^* = q_2^*$	3.911							
$d_1^* = d_2^*$	5.049	5.236	4.788	6.450	5.516	6.845	4.788	11.911
Z^*	86.035							
Job	6	4	1	3	7	5	8	2
Case 2, Subcase 2.3: $\alpha = 0.01, \beta = 10^6, \gamma = 5, \delta = 9$								
u_j	1.612	1.877	1.243	3.596	2.273	4.156	1.243	0
p_j	1.138	1.325	0.878	2.539	1.605	2.935	0.878	8.000
$q_1^* = q_2^*$	11.298							
$d_1^* = d_2^*$	12.436	12.623	12.175	13.837	12.902	14.232	12.175	19.298
Z^*	56.601							

Solution:

The optimal amount of resource allocated to each job, the actual processing time of each job, and the optimal flow-allowance values, due-dates and cost are provided in Table 4.

7 Conclusion

We studied minmax due-dates based on common flow-allowance assignment and scheduling problems on a single machine. We extended the basic model by considering controllable processing times and focused on a model where the job processing time is a convex function of the amount of resource allocated to it. The results of the due-date method were then generalized to that of common due-window. In both settings, the problems with position-independent workloads were shown to be solved in $O(n)$, whereas the problems with position-dependent workloads were solved in $O\left(n^3\right)$.

Given the similarities between the single machine and the proportionate flow-shop environments, interesting subjects for future research are the extensions of the addressed problems to the setting of proportionate flow-shop. Another challenging extension is to the setting of parallel machines. However, since even minimizing the makespan with constant processing times is ordinary NP-hard, a different approach must be applied.

Funding This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data availability Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Declaration

Conflict of interest The authors declare that they have no conflict of interest.

References

Adamopoulos GI, Pappis CP (1996) Single machine scheduling with flow allowances. J Oper Res Soc 47(10):1280–1285

Agnetis A, Mosheiov G (2017) Scheduling with job-rejection and position-dependent processing times on proportionate flowshops. Optim Lett 11(4):885–892

Agnetis A, Billaut JC, Gawiejnowicz S, Pacciarelli D, Soukhal A (2014) Multiagent scheduling. Springer, Berlin. https://doi.org/10.1007/978-3-642-41880-8

Cheng B, Cheng L (2018) Single machine slack due-window assignment and scheduling of linear time-dependent deteriorating jobs and a deteriorating maintenance activity. Open Access Library J 5(10):1–9

Gao F, Liu M, Wang JJ, Lu YY (2018) No-wait two-machine permutation flow shop scheduling problem with learning effect, common due date and controllable job processing times. Int J Prod Res 56(6):2361–2369

Gawiejnowicz S (2008) Time-dependent scheduling. Springer, Berlin

Geng XN, Wang JB, Bai D (2019) Common due date assignment scheduling for a no-wait flowshop with convex resource allocation and learning effect. Eng Optim 51(8):1301–1323

Gerstl E, Mor B, Mosheiov G (2017) Minmax scheduling with acceptable lead-times: extensions to position-dependent processing times, due-window and job rejection. Comput Oper Res 83:150–156

Graham RL, Lawler EL, Lenstra JK, Kan AR (1979) Optimization and approximation in deterministic sequencing and scheduling: a survey. Ann Discrete Math 5:287–326

Ji M, Zhang W, Liao L, Cheng TCE, Tan Y (2019) Multitasking parallel-machine scheduling with machine-dependent slack due-window assignment. Int J Prod Res 57(6):1667–1684

Kaspi M, Shabtay D (2004) Convex resource allocation for minimizing the makespan in a single machine with job release dates. Comput Oper Res 31(9):1481–1489

Liu W, Jiang C (2020) Due-date assignment scheduling involving job-dependent learning effects and convex resource allocation. Eng Optim 52(1):74–89

Liu L, Wang JJ, Wang XY (2016) Single machine due-window assignment scheduling with resource-dependent processing times to minimise total resource consumption cost. Int J Prod Res 54(4):1186–1195

Liu L, Wang JJ, Liu F, Liu M (2017) Single machine due window assignment and resource allocation scheduling problems with learning and general positional effects. J Manuf Syst 43:1–14

Liu W, Yao Y, Jiang C (2020) Single-machine resource allocation scheduling with due-date assignment, deterioration effect and position-dependent weights. Eng Optim 52(4):701–714

Lu YY, Liu JY (2018) A note on resource allocation scheduling with position-dependent workloads. Eng Optim 50(10):1810–1827

Mor B (2017) Minmax common due-window assignment and scheduling on a single machine with two competing agents. J Oper Res Soc. https://doi.org/10.1057/s41274-017-0253-0

Mor B (2019a) Minmax scheduling problems with common due-date and completion time penalty. J Combin Optim 38(1):50–71

Mor B (2019b) Single-machine minmax common due-window assignment and scheduling problems with convex resource allocation. Eng Optim 51(7):1251–1267

Mor B (2020) A unified approach for single machine scheduling with position-dependent workloads and positional penalties. SN Appl Sci 2(2):214

Mor B, Mosheiov G (2011) Total absolute deviation of job completion times on uniform and unrelated machines. Comput Oper Res 38(3):660–665

Mor B, Mosheiov G (2012a) Minmax scheduling problems with common flow-allowance. J Oper Res Soc 63(9):1284–1293

Mor B, Mosheiov G (2012b) Parallel machine scheduling problems with common flow-allowance. Int J Prod Econ 139(2):623–633

Mor B, Mosheiov G (2012c) Heuristics for scheduling problems with an unavailability constraint and position-dependent processing times. Comput Ind Eng 62(4):908–916

Mor B, Mosheiov G (2016) Minsum and minmax scheduling on a proportionate flowshop with common flow-allowance. Eur J Oper Res 254(2):360–370

Mor B, Mosheiov G (2017) A two-agent single machine scheduling problem with due-window assignment and a common flow-allowance. J Combin Optim 33(4):1454–1468

Mor B, Mosheiov G (2018) A note: Minimizing total absolute deviation of job completion times on unrelated machines with general position-dependent processing times and job-rejection. Ann Oper Res 271(2):1079–1085

Mor B, Mosheiov G, Shapira D (2020) Flowshop scheduling with learning effect and job rejection. J Sched 23:631–641. https://doi.org/10.1007/s10951-019-00612-y

Oron D (2016) Scheduling controllable processing time jobs with position-dependent workloads. Int J Prod Econ 173:153–160

Panwalkar SS, Smith ML, Seidmann A (1982) Common due date assignment to minimize total penalty for the one machine scheduling problem. Oper Res 30(2):391–399

Pei J, Cheng B, Liu X, Pardalos PM, Kong M (2019) Single-machine and parallel-machine serial-batching scheduling problems with position-based learning effect and linear setup time. Ann Oper Res 272(1–2):217–241

Rudek R (2012) Scheduling problems with position dependent job processing times: computational complexity results. Ann Oper Res 196(1):491–516

Seidmann A, Panwalkar SS, Smith ML (1981) Optimal assignment of due-dates for a single processor scheduling problem. Int J Prod Res 19(4):393–399

Shabtay D, Steiner G (2007) A survey of scheduling with controllable processing times. Discrete Appl Math 155(13):1643–1666

Sun LH, Cui K, Chen JH, Wang J, He XC (2013) Research on permutation flow shop scheduling problems with general position-dependent learning effects. Ann Oper Res 211(1):473–480

Sun LH, Cui K, Chen JH, Wang J (2016) Due date assignment and convex resource allocation scheduling with variable job processing times. Int J Prod Res 54(12):3551–3560

Sun X, Geng XN, Wang JB, Liu F (2019) Convex resource allocation scheduling in the no-wait flowshop with common flow allowance and learning effect. Int J Prod Res 57(6):1873–1891

Wang DJ, Yin Y, Cheng SR, Cheng TCE, Wu CC (2016) Due date assignment and scheduling on a single machine with two competing agents. Int J Prod Res 54(4):1152–1169

Wang D, Yin Y, Cheng TCE (2017) A bicriterion approach to common flow allowances due window assignment and scheduling with controllable processing times. Naval Res Logist (NRL) 64(1):41–63

Xiong X, Wang D, Edwin Cheng TC, Wu CC, Yin Y (2018) Single-machine scheduling and common due date assignment with potential machine disruption. Int J Prod Res 56(3):1345–1360

Yin Y, Wang DJ, Wu CC, Cheng TCE (2016) CON/SLK due date assignment and scheduling on a single machine with two agents. Naval Res Logist 63(5):416–429

Zhang X, Liao L, Zhang W, Cheng TCE, Tan Y, Ji M (2018) Single-machine group scheduling with new models of position-dependent processing times. Comput Ind Eng 117:1–5