



Optimal policies for minimizing total job completion times and deviations from common due dates in unrelated parallel machine scheduling

Oğuzhan Ahmet Arık¹

Accepted: 4 February 2024 / Published online: 1 March 2024

© The Author(s), under exclusive licence to Operational Research Society of India 2024

Abstract

This paper considers an unrelated parallel machine scheduling problem with a common due date where the objective is to minimize the sum of completion times, earliness, and tardiness durations of the jobs. This paper discovers some properties of the considered problem for restricted and unrestricted common due dates. By using these properties, the paper shows the existence of some optimal policies for the problem. Furthermore, optimal policies for the problem are found. According to our main findings, optimal policies for the problem are to increase the number of early jobs, to use the shortest processing time dispatching rule for tardy jobs, and to distribute tardy jobs on machines with tardy workloads that differ from each other as much as possible. These strategies for the problem are new to the literature and the problem has the potential to be extended. Using these strategies, the paper presents a novel heuristic algorithm for the problem that is rarely studied in the literature. Furthermore, the proposed heuristic is compared with two existing algorithms by generating some test instances for experimental study. The experimental study shows that the proposed heuristic outperforms other algorithms. Furthermore, the paper discusses which components of our proposed solution approach have impacts on the solution quality.

Keywords Unrelated parallel machine · Common due date · Earliness · Tardiness · The sum of completion times · Optimal policy

Mathematics Subject Classification 90B35 · 90C59

List of symbols

n Number of jobs
 m Number of machines

✉ Oğuzhan Ahmet Arık
oguzhanahmet@erciyes.edu.tr

¹ Industrial Engineering Department, Erciyes University, 38280 Kayseri, Turkey

| | |
|---|--|
| j | Index of jobs $j \in \{1, 2, \dots, n\}$ |
| i | Index of machines $i \in \{1, 2, \dots, m\}$ |
| d | The common due date for all jobs |
| E_j | Earliness of job j |
| T_j | Tardiness of job j |
| C_j | Completion time of job j |
| f_j^E | Cost effect of early job j in the objective function |
| f_j^T | Cost effect of tardy job j in the objective function |
| Δ | The sum of processing times that are processed after the common due date and until completion of the job j |
| $C_j^\pi, C_k^\pi, C_j^{\pi'}$ and $C_k^{\pi'}$ | Completion times of jobs (j, k) in schedules (π, π') |

1 Introduction, theoretical framework, and literature review

Although, minimizing the weighted sum of earliness/tardiness (E/T) costs or minimizing the weighted sum of completion times in parallel machine scheduling problems have been investigated since the 1970s, the problem that tries to minimize both of the sums of E/T durations and the sum of completion times simultaneously addressed in this paper has not got so much attention from the literature. According to the reported literature, Biskup and Cheng [1] only investigated the problem and they showed the problem is NP-hard and presented a polynomially solvable special case in which the processing times of all jobs are equal. Also, they proposed a heuristic that tries to assign jobs more evenly to have equal workloads for machines as much as possible. The main observation of Biskup and Cheng [1] in their study was that “in an optimal schedule, the numbers of jobs on the machines differ by as few as possible”. Up to the nature of the problem, if jobs are distributed machines more evenly, the completion times of jobs may decrease and other costs such as earliness and tardiness may be evenly shared among all jobs. There are similar solution approaches that try to share jobs evenly to machines. Arık and Toksarı [2] proposed a serpentine algorithm to distribute jobs evenly to machines in parallel where the objectives are to minimize the total tardiness penalty cost, to minimize the earliness penalty cost, and to minimize the cost of setting due dates. The objective of the proposed problem is interesting since it includes three different costs (the cost of completion time, the cost of earliness, and the cost of tardiness). Biskup and Cheng [1] gave an example of an application of the problem for readers.

The contemporary industrial landscape necessitates meticulous optimization of operational processes, especially in complex environments like parallel machine scheduling. Amidst evolving market dynamics, companies face multifaceted challenges requiring a delicate balance between minimizing completion times, earliness, and tardiness penalties while optimizing resource utilization. The overarching goal of this study is to delve into an unexplored realm of parallel machine scheduling, specifically focusing on the minimization of completion times, earliness, and

tardiness costs simultaneously—an area significantly underrepresented in current literature. Existing research has extensively explored variants of parallel machine scheduling, primarily concentrating on either minimizing weighted sums of earliness/tardiness costs or completion times in isolation. However, the critical interplay between these factors has received scant attention. The dearth of studies addressing the simultaneous optimization of completion times alongside earliness and tardiness costs in parallel machine environments forms the core gap that this research aims to fill. This study presents a compelling contribution by not only identifying this critical research gap but also by introducing optimal policies tailored to the unique complexities of this unexplored problem domain. By outlining strategies to increase early jobs, implement effective tardy job sequencing, and distribute tardy jobs efficiently across machines, this research lays the groundwork for an efficient heuristic algorithm—a key contribution to the field. Moreover, the comparative analysis against existing algorithms reveals the superior performance of the proposed heuristic. This underlines the practical relevance and applicability of the identified optimal policies, showcasing their efficacy in real-world scenarios. In addition to immediate applications in parallel machine environments, the implications of this study extend to diverse industries where timely delivery and efficient resource utilization are paramount. This research paves the way for strategic decision-making, aiding managers in devising scheduling strategies that optimize operations, minimize costs, and enhance overall operational efficiency. The limitations of existing studies and the unexplored potential for further enhancements, such as integrating additional constraints like release dates and sequence-dependent setup times, present promising avenues for future research. By extending the scope of the investigated problem and refining the identified optimal policies, future studies can further enrich the field of parallel machine scheduling. This study shows some properties of the problem and optimal policies for finding an optimal schedule for the problem where the objective is to minimize the completion times and earliness/tardiness durations.

For a close literature review of the investigated problem, there are similar papers considering both E/T durations and the sum of completion times. Su [3] addressed the identical parallel machine scheduling problem in which the total earliness and tardiness about a common due date are minimized subject to minimum total flowtime. He showed how to transform the problem into a simplified version of the parallel machine problem to minimize makespan subject to minimum total flowtime. Sivrikaya and Ulusoy [4] proposed a genetic algorithm for parallel machine E/T scheduling with sequence-dependent setup times and distinct due dates. Weng et al. [5] proposed several heuristics for unrelated parallel machine E/T scheduling problems with setup consideration and a total weighted completion time objective. Bilge et al. [6] considered uniform parallel machine tardiness scheduling with non-identical due dates, arrival times, and sequence-dependent setups. Liaw et al. [7] considered unrelated parallel machine weighted tardiness scheduling problems. They developed efficient lower and upper bounds for the problem. Radhakrishnan and Ventura [8] proposed simulated annealing for parallel machine scheduling with E/T penalties and sequence-dependent set-up times. Cheng et al. [9] proposed genetic

algorithms for minimax E/T scheduling in identical parallel machines. Balakrishnan et al. [10] presented a mixed-integer formulation for E/T scheduling with sequence-dependent setups on uniform parallel machines. Cheng and Chen [11] considered an identical parallel machine E/T scheduling problem where the common due date is a decision variable. They also showed that the special case in which all jobs have an equal processing time for the problem is polynomially solvable. Toksarı and Güner [12] introduced mixed nonlinear integer programming formulation for parallel machine E/T scheduling with simultaneous effects of learning and linear deterioration, sequence-dependent setups, and a common due date for all jobs. Biskup et al. [13] proposed a heuristic approach for identical parallel machines to minimize total tardiness. Shim and Kim [14] developed dominance properties and lower bounds for parallel identical machine problems where the objective is to minimize total tardiness. Furthermore, they developed a branch and bound algorithm using their properties. Bank and Werner [15] proposed heuristic algorithms for unrelated parallel machine scheduling with a common due date, release dates, and linear earliness and tardiness penalties. Yin et al. [16] considered a two-agent unrelated parallel machine scheduling problem where the overall objective is to minimize the total completion time of the jobs of one agent while keeping the weighted number of tardy jobs of another agent within a given limit. They introduced a novel column generation scheme for the problem. Arik [17] compared three different metaheuristic algorithms to determine what kind (swarm intelligence-based, evolutionary or single solution) of metaheuristics is effective in solving unrelated parallel machine E/T scheduling problems. Computational results in Arik's study showed that the artificial bee algorithm outperforms its opponents in view of solution quality as a swarm intelligence-based metaheuristic algorithm. Arik et al. [18] investigated an unrelated parallel machine weighted earliness/tardiness scheduling problem with the common due date. They proposed four solution construction algorithms using the V-shaped property of the problem and two well-known dispatching rules. Then they proposed five solution improvement heuristics including different local search mechanisms. They compared their algorithms with Arik's [17] proposed algorithms and other algorithms in the literature. The experimental study reveals that their solution improvement algorithm with swap and reinsertion-based local search mechanisms outperforms other algorithms in the literature.

Arik [19] considered the common due date assignment for single machine weighted earliness/tardiness scheduling problem with distinct weights to minimize the cost of weighted earliness/tardiness and due date assignment cost. He analyzed optimal properties like V-shaped property of similar problems and applied some of them to generate an efficient heuristic algorithm to solve the problem. Kordmostafapour et al. [20] investigated metaheuristic algorithms for unrelated parallel machine batch scheduling problems in case of that the processing time of jobs is controllable on resource allocation. They proposed also new mathematical model for their problem. Chen et al. [21] presented a polynomial algorithm to solve single machine scheduling problem with due date assignment under group technology environment. Geng et al. [22] proposed a dynamic programming algorithm to solve proportionate

flow shop scheduling problems allowing job rejection where the objective function is to minimize the earliness, tardiness, due date cost, and rejection cost. They also presented analyses of the algorithm complexity. Pan et al. [23] designed efficient and fast polynomial time algorithms to solve the single machine scheduling problem with due date assignments, deteriorating jobs, and past sequence-dependent delivery times by using some optimal policies of the problem. Mosheiov and Sarig [24] studied a common due date assignment problem for two parallel uniform machines where jobs have identical processing times and job-dependent and asymmetric earliness and tardiness costs. They showed the existence of a polynomial algorithm for the problem. Wang [25] proved that there are optimal properties of single machine due date assignment problem with past sequence-dependent setup times. Wang also presented a polynomial time algorithm for the problem. Nasrollahi et al. [26] investigated the common due date problem of constrained two-agent scheduling of jobs in a two-machine flow shop environment to minimize the weighted sum of maximum earliness and maximum tardiness. They presented a branch and bound algorithm based on efficient lower and upper bounds and dominance rules for the problem. Falq et al. [27] single machine weighted earliness/tardiness scheduling with unrestrictive common due date. They investigated neighborhood-based dominance properties. Qian and Han [28] considered a single machine scheduling problem with distinct due dates and jobs under deterioration effect. They proved that there are polynomial time algorithms to solve their problems.

Today's complex and challenging market conditions make the company consider multiple penalty costs such as completion time, tardiness, and earliness penalties because the company has to satisfy its customers by sending them their goods and services on time and it has to increase its operational efficiency. The common due date for all jobs may indicate that all jobs are going to be delivered at the same time and completing any job before/after the shipment date may bring additional costs like insurance costs for an early completed job and shipment delaying costs for a tardy job. Furthermore, the pressure on the company to use its operational resources such as machines, production lines, and facilities efficiently makes considering completion time penalties compulsory for the company. Thus, a natural problem with earliness, tardiness, and completion time penalties arises for companies that have parallel machine environments for manufacturing. This study investigates this problem with an unrelated parallel machine environment and reveals some optimal policies for its solution. Furthermore, this study proposes a solution construction and improvement heuristic algorithm using optimal policies of the problem.

The remainder of the paper is organized as follows: Sect. 2 introduces the properties of the problem. The optimal policies of the problem with four different theorems are shown in Sect. 2. Using these optimal policies, a solution construction and improvement heuristic is also designed for the problem in Sect. 2. In Sect. 3, some test instances are generated for comparison of our proposed heuristics with two existing heuristics. Section 4 discusses the effect of the components of the proposed heuristic. Section 5 concludes the study and gives a future direction of the problem for the reader.

2 Methodology

2.1 Properties of the problem

This section shows some properties of the problem. Let us assume there are n jobs that are ready to be processed on m machines in parallel. Each job j ($j \in \{1, 2, \dots, n\}$) can be processed on machine i ($i \in \{1, 2, \dots, m\}$) with the processing time P_{ij} . There is a common due date d for all jobs. If a job is completed before the common due date, then that job is early and it has a cost E_j as follows:

$$E_j = \text{Max}(0, d - C_j) \tag{1}$$

where C_j is the completion time of the job j . If a job is completed after the common due date, then that job is tardy and it has a cost T_j as follows:

$$T_j = \text{Max}(0, C_j - d) \tag{2}$$

The objective function is to minimize the sum of completion times, earliness, and tardiness durations $\left(\sum_{j=1}^n (C_j + E_j + T_j)\right)$ with equal weights. Since the machines in parallel have different capacities, velocities, and properties; the processing times of each job are different for machines. Thus the problem is an unrelated parallel machine scheduling problem with the common due date where the objective is to minimize the sum of completion times, earliness, and tardiness durations. If the problem is notated with the triple scheduling notations $(\alpha/\beta/\gamma)$, then the problem is $R|d_j = d|\sum_{j=1}^n (C_j + E_j + T_j)$.

Lemma 1 *Each early job’s effect on the objective function is equal to d .*

Let j be a job and its completion time C_j is less than d . Therefore job j is an early job. The effect (f_j^E) of job j on the objective function of the problem of $R|d_j = d|\sum_j(C_j + E_j + T_j)$ is follows:

$$f_j^E = C_j + E_j + T_j \tag{3}$$

$$f_j^E = C_j + \max(0, d - C_j) + \max(0, C_j - d) \tag{4}$$

$$f_j^E = C_j + d - C_j \tag{5}$$

$$f_j^E = d \tag{6}$$

Lemma 2 *Each tardy job’s effect on the objective function is equal to $2C_j - d$.*

Let j be a job and its completion time C_j is more than d . Therefore job j is a tardy job. The effect (f_j^T) of job j on the objective function of the problem of $R|d_j = d|\sum_j(C_j + E_j + T_j)$ is follows:

$$f_j^T = C_j + E_j + T_j \quad (7)$$

$$f_j^T = C_j + \max(0, d - C_j) + \max(0, C_j - d) \quad (8)$$

$$f_j^T = C_j + C_j - d \quad (9)$$

$$f_j^T = 2C_j - d \quad (10)$$

Lemma 3 *A tardy job's effect is more than an early job's effect on the objective function of the problem of $R|d_j = d|\sum_j(C_j + E_j + T_j)$.*

Since C_j is more than d for a tardy job j , it can be assumed that $C_j = d + \Delta$ where Δ is the sum of processing times that are processed after the common due date and until completion of the job j . The effect of a tardy job on the objective function can be rewritten as follows:

$$f_j^T = 2C_j - d \quad (11)$$

$$f_j^T = 2(d + \Delta) - d \quad (12)$$

$$f_j^T = d + 2\Delta \quad (13)$$

Since $\Delta > 0$, f_j^T is greater than f_j^E .

Lemma 4 *Zero start times of machines decrease the total cost of the given schedule more than the non-zero start times of machines.*

The proof of Lemma 4 is trivial because non-zero start times increase (decrease) completion times of jobs (the number of early jobs) so the zero start time of a machine for the problem is a property of the optimal schedule.

2.2 Optimal policies for the problem of $R|d_j = d|\sum_j(C_j + E_j + T_j)$

This section shows optimal policies for the problem of $R|d_j = d|\sum_j(C_j + E_j + T_j)$. Three theorems are used to demonstrate the policies for the optimal schedule.

Theorem 1 For the problem of $R|d_j = d|\sum_j(C_j + E_j + T_j)$, if d is unrestricted ($d > \sum_j P_{ij} \forall i$), the optimal schedule is the schedule where all jobs are early. Thus the objective function of the optimum schedule is equal to $(n - m)d$ where n is the number of jobs and m is the number of machines.

Proof Since all C_j values are less than d , when d is unrestrictive so there is no tardy job and there are m jobs completed on the common due date. Thus, the optimum objective function value is equal to $\sum_j f_j^E = (n - m)d$. This completes the proof of Theorem 1.

Theorem 2 For the problem of $R|d_j = d|\sum_j(C_j + E_j + T_j)$, if d is restrictive ($d < \frac{\sum_j P_{ij}}{h}$ for any machine i) where h is the restrictive factor), the optimal schedule is the schedule where all tardy jobs on the same machine are ordered in increasing order of their processing time.

Proof Let us have two tardy neighboring jobs (j, k) and these jobs are to be processed on machine i among all machines in parallel. There are two alternative schedules. These are π and π' , respectively. In schedule π , job j precedes job k in machine i . $C_j^\pi, C_k^\pi, C_j^{\pi'}$ and $C_k^{\pi'}$ are completion times of jobs (j, k) in schedules (π, π') and these are greater than d . These tardy jobs' costs for π and π' are f^π and $f^{\pi'}$, respectively. Our assumption is $P_{ik} > P_{ij}$ and let M_i is the time that machine i can start processing jobs (j, k) .

For schedule π ,

$$C_j^\pi = M_i + P_{ij} \tag{14}$$

$$C_k^\pi = C_j^\pi + P_{ik} \tag{15}$$

$$C_k^\pi = M_i + P_{ij} + P_{ik} \tag{16}$$

$$f^\pi = C_j^\pi + C_k^\pi + (C_j^\pi - d) + (C_k^\pi - d) \tag{17}$$

$$f^\pi = M_i + P_{ij} + M_i + P_{ij} + P_{ik} + M_i + P_{ij} - d + M_i + P_{ij} + P_{ik} - d \tag{18}$$

$$f^\pi = 3M_i + 4P_{ij} + 2P_{ik} - 2d \tag{19}$$

For schedule π' ,

$$C_k^{\pi'} = M_i + P_{ik} \tag{20}$$

$$C_j^{\pi'} = C_k^{\pi'} + P_{ij} \quad (21)$$

$$C_j^{\pi'} = M_i + P_{ik} + P_{ij} \quad (22)$$

$$f^{\pi'} = C_k^{\pi'} + C_j^{\pi'} + (C_k^{\pi'} - d) + (C_j^{\pi'} - d) \quad (23)$$

$$f^{\pi'} = M_i + P_{ik} + M_i + P_{ik} + P_{ij} + M_i + P_{ik} - d + M_i + P_{ik} + P_{ij} - d \quad (24)$$

$$f^{\pi'} = 3M_i + 4P_{ik} + 2P_{ij} - 2d \quad (25)$$

If the shortest processing time dispatching rule assures optimum for tardy jobs then $f^{\pi'} - f^{\pi}$ must be greater than zero.

$$f^{\pi'} - f^{\pi} = 3M_i + 4P_{ik} + 2P_{ij} - 2d - 3M_i - 4P_{ij} - 2P_{ik} + 2d \quad (26)$$

$$f^{\pi'} - f^{\pi} = 2P_{ik} - 2P_{ij} \quad (27)$$

Since $P_{ik} > P_{ij}$, $2P_{ik} - 2P_{ij}$ is greater than zero. This completes the proof for Theorem 2.

Theorem 3 *If the number of early jobs increases for the problem of $R|d_j = d| \sum_j (C_j + E_j + T_j)$, the objective function value of the schedule decreases.*

Proof Let us have n jobs. The schedule π includes x early jobs and y tardy jobs. It is known that $x + y = n$. The schedule π' includes $x + 1$ early jobs and $y - 1$ tardy jobs. The objective functions of these schedules are f^{π} and $f^{\pi'}$, respectively. If Theorem 3 holds, then $f^{\pi} - f^{\pi'} > 0$.

For schedule π ,

$$f^{\pi} = xd + \sum_{l=1}^y (d + \Delta_l^{\pi}) \quad (28)$$

For schedule π' ,

$$f^{\pi'} = (x + 1)d + \sum_{l=1}^{y-1} (d + \Delta_l^{\pi'}) \quad (29)$$

Δ_l^{π} is the duration between the common due date and completion time of a tardy job l in schedule π . Let us have three tardy jobs in a machine. These jobs are k, t , and l . The

job k is the first tardy job in machine i . The job t is the second tardy job in machine i . The job l is the last tardy job in machine i . Thus $\Delta_k^\pi = d + P_{ik}$, $\Delta_t^\pi = d + P_{ik} + P_{it}$ and $\Delta_l^\pi = d + P_{ik} + P_{it} + P_{il}$. Any of these tardy jobs is selected to make it an early job by arranging the schedule so a new schedule π' is obtained. As seen, there is an extra early job in the schedule π' than the schedule π has.

$$f^\pi - f^{\pi'} = xd + \sum_{l=1}^y (d + \Delta_l^\pi) - (x + 1)d - \sum_{l=1}^{y-1} (d + \Delta_l^{\pi'}) \tag{30}$$

$$f^\pi - f^{\pi'} = xd + yd + \sum_{l=1}^y \Delta_l^\pi - (x + 1)d - (y - 1)d - \sum_{l=1}^{y-1} \Delta_l^{\pi'} \tag{31}$$

$$f^\pi - f^{\pi'} = \sum_{l=1}^y \Delta_l^\pi - \sum_{l=1}^{y-1} \Delta_l^{\pi'} \tag{32}$$

Since $\sum_{l=1}^y \Delta_l^\pi$ includes an extra tardy job than $\sum_{l=1}^{y-1} \Delta_l^{\pi'}$, the expression of $\sum_{l=1}^y \Delta_l^\pi - \sum_{l=1}^{y-1} \Delta_l^{\pi'}$ is greater than zero. Thus, Theorem 3 is proved.

Theorem 4 For $R|d_j = d| \sum_j(C_j + E_j + T_j)$, if tardy jobs are distributed on machines considering equal tardy workloads on machines as much as possible, and then the objective function of the problem decreases.

Proof Biskup and Cheng [1] indicate the importance of disturbing jobs evenly to the identical machines or making workloads of the identical machines as equal as possible. Since each job has different processing times for machines in case of unrelated parallel machine problems, it should be considered the workloads of machines instead of the number of jobs on machines. It is also known that any early job’s effect on the objective function is the same and it is independent of its position, machine, and processing time, so the number of early jobs must be increased to decrease tardiness cost. Thus our first case for proving Theorem 4 investigates the earliness of the schedule instead of tardiness. Since each job’s processing time is dependent on machines, it should be considered the machine workload for tardy jobs by assigning each tardy job to the machine with minimum cost or tardiness. Minimizing overall tardiness and distributing the tardy workload to machines serve the same goal which is to minimize the objective function. Our second case investigates the effects of having equal tardy workloads on machines as much as possible.

Case 1 Let us assume that there are n jobs and two unrelated machines in parallel. The indexes for machines are k and l . The sum of processing times for the machine k is Wk and the sum of processing times for the machine l is Wl . The starting times of machines to process jobs are MK and ML where $MK \geq 0$ and $ML \geq 0$. The common due date (d) is a restrictive due date such as $d < Wk/2$ or $d < Wl/2$.

If all jobs are assigned to the machine k , the workload of the machine k and the completion time of the last job in machine k will be Wk and $Wk + MK$, respectively. Since all jobs are assigned to machine k , the workload of the machine l will be zero. If all jobs are assigned to machine l , the machine workload of machine l and the completion time of the last job in machine l will be Wl and $ML + Wl$, respectively. Thus the machine workload of the machine k will be zero. In both cases, some of the jobs will be early jobs and some of the jobs will be tardy jobs.

Let us have two different schedules named π^1 , and π^2 , respectively. Schedule π^1 is the schedule where all jobs are assigned to machine k . Schedule π^2 is the schedule where $n - 1$ jobs are assigned to machine k and machine l has only one job. In schedule π^1 , there are x early (and an on-due date job at most) and y tardy jobs ($x + y = n$). The processing time of the job on position j ($j \in \{1, 2, \dots, n\}$) in machine i ($i \in \{k, l\}$) is P_{ij} . The total cost functions of schedules are f^{π^1} , and f^{π^2} respectively. If the theorem holds, then $f^{\pi^1} > f^{\pi^2}$.

$$Wl = \sum_{j=1}^{j=n} P_{jl} \tag{33}$$

$$Wk = \sum_{j=1}^{j=n} P_{jk} \tag{34}$$

$$f^{\pi^1} = \sum_{j=1}^{j=n} C_{jk} + \sum_{j=1}^{j=n} E_{jk} + \sum_{j=1}^{j=n} T_{jk} \tag{35}$$

This expression can be simplified by using Lemma 1 and Lemma 2 as follows:

$$f^{\pi^1} = xd + \sum_{jt=1}^{jt=y} 2C_{jt,k} - d \tag{36}$$

where jt is the index for current tardy jobs in machine k and $C_{jt,k}$ is the completion time of a tardy job jt in machine k . If the last job from the machine k in the schedule π^1 is moved to machine l and the start time of that job is to be completed on the common due date, then the new schedule's (π^2) total cost function will be as follows:

$$f^{\pi^2} = (x + 1)d + \sum_{jt=1}^{jt=y-1} 2C_{jt,k} - d \tag{37}$$

The last job (possibly a tardy job) from machine k is removed to machine l by making it an on-date (completed on the common due date) job. This decreases the objective function value of π^1 schedule if the processing time of the removed job lets itself be completed on the common due date. Thus, $f^{\pi^1} > f^{\pi^2}$ and $f^{\pi^1} - f^{\pi^2} > 0$.

$$f^{\pi^1} - f^{\pi^2} = xd + \sum_{jt=1}^{jt=y} 2C_{jt,k} - d - (x + 1)d - \sum_{jt=1}^{jt=y-1} 2C_{jt,k} + d \tag{38}$$

$$f^{\pi^1} - f^{\pi^2} = \sum_{jt=1}^{jt=y} 2C_{jt,k} - \sum_{jt=1}^{jt=y-1} 2C_{jt,k} \tag{39}$$

$$f^{\pi^1} - f^{\pi^2} = 2C_{y,k} \tag{40}$$

where $C_{y,k}$ is the completion time of the removed tardy job from position y of machine k and it is known that it is greater than zero. If it is continued to remove tardy jobs and insert them into available machines' time slots between zero and the common due date, the objective function goes on being decreased too. Thus, Theorem 4 for Case 1 is proved.

Case 2 The workload of tardy jobs must be eliminated by increasing the number of early jobs. If it is not possible then the workloads of tardy jobs on machines should be balanced by distributing tardy jobs to have balanced tardy workloads for all machines without increasing total tardiness. The second case with similar assumptions is proposed. Let us assume that there are n jobs and two unrelated machines in parallel. The indexes for machines are k and l . Let us have two different schedules named π^1 , and π^2 , respectively. Schedule π^1 is the schedule where numbers of early jobs on machines l and k are x and y , respectively. In schedule π^1 , there are z tardy jobs ($x + y + z = n$) and these are assigned to machine k . As an assumption, none of these tardy jobs can be an early job in any machine by removing them from their current positions because there is no proper position of machines in the time interval between 0 and d without turning an early job into a tardy job. Schedule π^2 is the schedule where the last tardy job in machine k is removed from its current position to the end of the sequence of machine l . After removing that job, there is a tardy job in machine l and there are $z - 1$ tardy jobs. If the completion time of the recently assigned tardy job is less than the completion time on its previous position, then the tardy workload of machine k decreases while the differences among tardy workloads of machines decrease. If the decrease in the differences among tardy workloads of machines makes the schedule better then $f^{\pi^1} > f^{\pi^2}$.

$$f^{\pi^1,l} = xd \tag{41}$$

$$f^{\pi^1,k} = yd + \sum_{jt=1}^{jt=z} 2C_{jt,k} - d \tag{42}$$

$$f^{\pi^1} = (x + y - z)d + 2 \sum_{j=1}^{j_t=z} C_{j_t,k} \quad (43)$$

where $f^{\pi^1,l}$ and $f^{\pi^1,k}$ are cost effects of machines k and l on the objective function value f^{π^1} of schedule π^1 .

$$f^{\pi^2,l} = xd + 2C_{1,l} - d \quad (44)$$

$$f^{\pi^2,k} = yd + \sum_{j=1}^{j_t=z-1} 2C_{j_t,k} - d \quad (45)$$

$$f^{\pi^2} = (x + y - z)d + 2C_{1,l} + 2 \sum_{j=1}^{j_t=z-1} C_{j_t,k} \quad (46)$$

where $f^{\pi^2,l}$ and $f^{\pi^2,k}$ are cost effects of machines k and l on the objective function value f^{π^2} of schedule π^2 .

$$f^{\pi^1} - f^{\pi^2} = \left[(x + y - z)d + 2 \sum_{j=1}^{j_t=z} C_{j_t,k} \right] - \left[(x + y - z)d + 2C_{1,l} + 2 \sum_{j=1}^{j_t=z-1} C_{j_t,k} \right] \quad (47)$$

$$f^{\pi^1} - f^{\pi^2} = 2(C_{z,k} - C_{1,l}) \quad (48)$$

If the completion time $C_{1,l}$ of the recently assigned tardy job is less than the completion time $C_{z,k}$ on its previous position, then the tardy workload of machine k decreases while the differences among tardy workloads of machines decrease thus $f^{\pi^1} - f^{\pi^2} > 0$. Thus Theorem 4 holds for Case 2 of its proof and it is the end of the proof for Theorem 4. If we find and remove tardy jobs from their current positions to make them tardy jobs again with fewer completion times and fewer differences among tardy workloads of machines, the objective function value of the schedule is decreased.

2.3 Solution construction algorithm

Some optimal policies for $R|d_j = d| \sum_j (C_j + E_j + T_j)$ are discovered. These are as follows:

- Increasing the number of early jobs while decreasing the number of tardy jobs,
- Sequencing tardy jobs in increasing order of their processing times in machines,
- Distributing tardy jobs on machines with equal tardy workloads without increasing the cost.

By using our main observations, a heuristic method for $R|d_j = d|\sum_j(C_j + E_j + T_j)$ is proposed. Since it is needed to increase the number of early jobs per machine, an iterative solution construction algorithm is proposed to assign a job to a machine at each iteration. There are three phases of our proposed algorithm. In the first phase, our construction algorithm selects the job-machine pair that gives the smallest completion time for $R|d_j = d|\sum_j(C_j + E_j + T_j)d$, then it assigns that job to the machine with the smallest completion time. This goes on until there is no unassigned job left. Since our optimal policies are to increase (decrease) the number of early (tardy) jobs and to distribute tardy jobs on machines with equal workloads, it is considered that the machine start times are 0 to assign more early jobs in the time interval between 0 and. Thus completion times of jobs are determined with zero start times of machines. Some of these assigned jobs will be early and tardy according to their completion times. After assigning these jobs, early and tardy jobs of each machine are already sequenced with the SPT dispatching rule. Calculating completion times of jobs and the cost of the schedule are in the second phase of the algorithm. Since Biskup and Cheng [1] proved that there is no idle time between two consecutive jobs in the optimal schedule, the completion times of jobs according to this property of the problem's optimum schedule are calculated. In the third phase of the algorithm, the workloads of tardy jobs by swapping and moving them among machines are balanced. After all swapping and moving operations are completed for selected jobs, tardy jobs of each machine are reordered in order of their increasing processing times according to the SPT dispatching rule. The pseudo-code of our proposed construction algorithm having $O(nm)$ complexity is given in Algorithm 1. Also the flowchart of the algorithm is given in Fig. 1.

The flowchart of the algorithm in Fig. 1 shows phases of the algorithm. Details of these phases can be found in Algorithm 1.

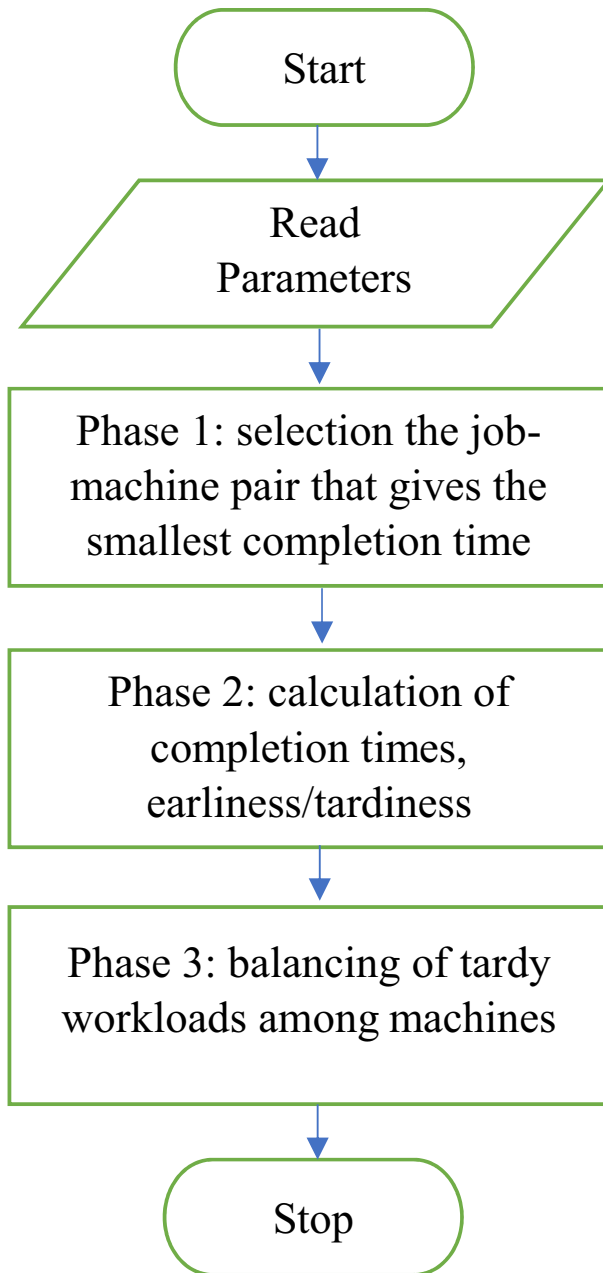


Fig. 1 The flowchart of the algorithm

Algorithm 1 The proposed heuristic for $R|d_j = d|\sum_j (C_j + E_j + T_j)$

Read n, m, d, P_{ij}
Set $S \in \{1, 2, 3, \dots, n\}$ % the set of unassigned jobs
Set $Cost = 0$ % the cost of the solution
Declare a two-dimensional array $\pi[m, n]$ % solution representation with $m * n$ cells
Declare a two-dimensional array $C[m, n]$ % completion times
Declare a two-dimensional array $T[m, n]$ % tardy jobs
Declare an one-dimensional array $NE[m]$ % the last non-empty cell's index in machine i
Declare an one-dimensional array $TR[m]$ % the first tardy job's position index in machine i
% the first phase
For $j = 1$ to n
 $Min = INT_MAXVALUE$
 $Jl = 0$
 $Ml = 0$
 For $i = 1$ to m
 $k =$ Find the index of job that has smallest processing time for machine i
 $y = NE[i] + 1$
 $\pi[i, y] = k$
 Calculate $C[i, y]$
 If $C[i, y] < Min$ **then** $Min = C[i, y]$ **and** $Jl = k$ **and** $Ml = i$
 $\pi[i, y] = 0$
 Next i
 Remove Jl from S
 $z = NE[Ml] + 1$
 $\pi[i, z] = Jl$
 Calculate $C[i, z]$
 If $C[i, z] > d$ **then** $T[i, z] = 1$
 If Jl is the first tardy job in machine i **then** $TR[i] = z$
 $NE[Ml]++$
Next j
% the second phase
 $Cost =$ Calculate the cost of π
% the third phase
For $i1 = 1$ to m
 $r1 = 0$
 If $TR[i1] = 0$ **then** $r1 = NE[i1] + 1$
 If $TR[i1] > 0$ **then** $r1 = TR[i1]$
 For $j1 = r1$ to $NE[i1] + 1$
 π' = the best schedule after swapping $\pi[i1, j1]$ and another tardy job from any machine
 except for machine i or moving $[i1, j1]$ to the last positions of other machines
 Reorder of tardy jobs in π' using SPT dispatching rule
 $Cost' =$ Calculate the cost of π'
 If $Cost' < Cost$ **then** $Cost = Cost'$ **and** $\pi = \pi'$
 Next $j1$
Next $i1$
Display $Cost$ and $\pi[m, n]$

3 Results

For comparison of our proposed heuristic, this section generates some test problems where there are 16 test groups and each group has 10 problems. The numbers of jobs (machines) for test instances are 50, 70, 100, and 150 (2, 4, 7, and 10). The processing times of jobs for unrelated parallel machines are generated randomly by using a uniform distribution like $P_{ij} = U[1, 10] \forall i, j$. For common due date, the formula of Bank and Werner [15] is used as follows:

$$d = \frac{1}{h} \frac{n}{m} \bar{P} \quad (49)$$

where

$$\bar{P} = \frac{\sum_{i=1}^n \sum_{j=1}^m P_{ij}}{nm} \quad (50)$$

and h is the restrictive factor for the problem and Bank and Werner [15] used $h = 3$ in their experiments. Three different restrictive factors ($h \in \{3, 4, 5\}$) are used for our experiments. To compare the performance of our proposed heuristic, two existing heuristic algorithms are used. These are proposed by Biskup and Cheng [1] and Arık and Toksarı [2] for identical parallel machine problems. Biskup and Cheng [1] investigated our problem in an identical parallel machine environment and they proposed a heuristic by using their main observation as “in an optimal schedule, the numbers of jobs on the machines differ by as few as possible”. For an identical parallel machine environment, Arık and Toksarı [2] proposed a serpentine algorithm to distribute jobs evenly to machines in parallel where the objectives are to minimize total tardiness penalty cost, to minimize earliness penalty cost, and to minimize the cost of setting due dates. Up to the nature of the problem in an identical parallel machine environment, if jobs are distributed machines more evenly, the completion times of jobs may decrease and other costs such as earliness and tardiness may be evenly shared among all jobs. Since the proposed problem is in an unrelated parallel machine environment, minimizing differences among workloads of machines instead of minimizing differences among numbers of jobs in machines. Our proposed algorithm is to distribute jobs on machines to have balanced workloads among machines as much as possible considering cost factors. Both algorithms of Biskup and Cheng [1] and Arık and Toksarı [2] are based on list-scheduling that generates a permutation list (SPT or LPT) of jobs and assigns these jobs one by one to positions of machines by following a certain assignment path. Since our problem is in an unrelated parallel machine environment, permutation lists of compared algorithms are modified.

Arık and Toksarı’s [2] serpentine algorithm is designed for identical parallel machine problems and it used an SPT list of jobs considering their processing times to assign them to machines since the processing time of a job is dependent on machines, their algorithm is modified. In each job-machine assignment in the modified serpentine algorithm’s assignment path, the algorithm assigns the job with the smallest processing time to the position of the machine. Biskup and

Table 1 Overall average percentage derivations of compared algorithms

| The restrictive factor | Our proposed algorithm | Arık and Toksarı [2] | Biskup and Cheng [1] |
|------------------------|------------------------|----------------------|----------------------|
| 3 | 0.1233 | 0.2095 | 0.2473 |
| 4 | 0.2125 | 0.3976 | 0.5704 |
| 5 | 0.3743 | 0.6399 | 0.9743 |

Cheng [1] also used an LPT list of jobs to assign them to identical machines. The permutation list proposed by Adamopoulos and Pappis [29] is used within the algorithm of Biskup and Cheng [1]. Adamopoulos and Pappis's [29] permutation list of jobs for unrelated parallel machines is a non-increasing order of jobs' two smallest processing times' differences. Only that list in the algorithm of Biskup and Cheng [1] is integrated.

Three algorithms are coded in C++ programming language with a standard desktop having an Intel i5-10210U CPU and 8 GB RAM. All test instances are solved with our proposed heuristic and both algorithms of Biskup and Cheng [1] and Arık and Toksarı [2]. Lower bounds of test instances are calculated with a simple formula that is $(n - m)d$. It is known that each early job's cost is d and less than any tardy job's cost. If we have a theoretical schedule in where each machine has one job that is completed on the common due date and has no tardy job, then there are $n - m$ early jobs. Thus, the lower bound of the problem is equal to $(n - m)d$. Furthermore, we use the average percentage derivations of proposed algorithms from the lower bound of the problem to compare their solution quality. The formula for this derivation is calculated as $(f - LB)/LB$ where LB is the lower bound of the instance and f is the objective function value obtained from any proposed algorithm. Table 1 shows the overall average percentage deviation of proposed algorithms for all test instances and restrictive factors. As seen from Table 1, the proposed heuristic outperforms other compared algorithms considering its solution quality. In Table 1, the best value for each restrictive factor is marked with bold font.

While the restrictive factor is increasing for the same test instances, the common due date increases, and the solution qualities of proposed algorithms decrease because the objective function of the problem is highly dependent on the common due date. The revealed optimal policies of the problem are integrated within our proposed heuristic and these optimal policies increase the solution quality of the proposed algorithm and make it outperform the other two existing algorithms considering solution quality. The serpentine algorithm of Arık and Toksarı [2] is better than the algorithm of Biskup and Cheng [1] because it uses an assignment method that selects the job with the smallest processing time for the current position of the machine while the algorithm of Biskup and Cheng [1] uses an LPT permutation list for jobs. Our algorithm assigns jobs considering the workloads of machines by selecting the machine-job pair with the smallest workload. Therefore, we may state that considering balanced workloads among machines decreases the objective function more than considering balanced numbers of jobs for machines.

4 Discussion

This section briefly discusses what the study has done in this study for the investigated problem. The objective of the investigated problem is to minimize the sum of completion times, earliness, and tardiness durations of jobs in an unrelated parallel machine environment with the common due date. The problem of minimizing both the sums of E/T durations and the sum of completion times simultaneously has not so much attention from the literature. The common due date in scheduling problems is classified as restrictive and unrestrictive. For the problem with either restrictive or unrestrictive common due dates, the optimal schedule has some characteristics that can be expressed with optimal policies of the problem. In Sect. 2, we reveal the optimal policies of the problem which are to increase the number of early jobs, to use the shortest processing time dispatching rule for tardy jobs, and to distribute tardy jobs on machines with tardy workloads differ from each other as much as possible. We adapt these findings to our proposed heuristic that constructs a solution by using SPT-based list scheduling and improves the solution by swapping jobs between machines to distribute tardy job durations evenly among machines. After each swap operation, resequencing of tardy jobs in each machine is done by using the SPT dispatching rule to decrease the objective function value of the solution. Our proposed heuristic outperforms the existing solution approaches of Biskup and Cheng [1] and Arık and Toksarı [2] by using our main findings in Sect. 2, these are as follows:

- *Solution construction phase* Increasing the number of early jobs while decreasing the number of tardy jobs,
- *Job swapping phase* Distributing tardy jobs on machines with equal tardy workloads without increasing the cost.
- *Resequencing phase* Sequencing tardy jobs in increasing order of their processing times in machines.

In this section, we analyze our solution approach with and without these components that serve for optimal policies of the problem. The variants of our solution approach are as follows:

Var#0: the original solution approach with its full features as designed in Sect. 4,

Var#1: it uses a random initial solution instead of the first phase of Algorithm 1, the rest of its components are the same,

Var#2: it does not reorder tardy jobs in each machine, the rest of its components are the same,

Var#3: it uses a random initial solution instead of the first phase of Algorithm 1 and it does not reorder tardy jobs in each machine.

Var#4: it just reorders tardy jobs in each machine for the best solution in the memory at the end, the rest of its components are the same,

The same test problems in the experimental study of this paper are used to determine which components are effective in solution quality by comparing the

Table 2 Overall average percentage derivations of all variants of the proposed solution approach

| The restrictive factor | Var#0 | Var#1 | Var#2 | Var#3 | Var#4 |
|------------------------|--------|--------|--------|--------|--------|
| 3 | 0.1233 | 0.4054 | 0.1233 | 0.3929 | 0.1233 |
| 4 | 0.2125 | 0.7008 | 0.2125 | 0.7216 | 0.2125 |
| 5 | 0.3743 | 1.0597 | 0.3743 | 1.0220 | 0.3743 |

Factor Information

| Factor | Type | Levels | Values |
|--------------------|-------|--------|-----------------------------------|
| n | Fixed | 4 | 50; 70; 100; 150 |
| m | Fixed | 4 | 2; 4; 7; 10 |
| restrictive factor | Fixed | 3 | 3; 4; 5 |
| Method | Fixed | 5 | Var#0; Var#1; Var#2; Var#3; Var#4 |

Analysis of Variance

| Source | DF | Adj SS | Adj MS | F-Value | P-Value |
|--------------------|------|---------|---------|---------|---------|
| n | 3 | 5,177 | 1,7257 | 40,76 | 0,000 |
| m | 3 | 83,540 | 27,8467 | 657,66 | 0,000 |
| restrictive factor | 2 | 66,697 | 33,3485 | 787,60 | 0,000 |
| Method | 4 | 132,926 | 33,2315 | 784,84 | 0,000 |
| Error | 2387 | 101,070 | 0,0423 | | |
| Lack-of-Fit | 227 | 67,573 | 0,2977 | 19,20 | 0,000 |
| Pure Error | 2160 | 33,497 | 0,0155 | | |
| Total | 2399 | 389,411 | | | |

Fig. 2 ANOVA test for measuring the effects of variants on the solution quality

variants of the proposed solution approach in an equal computation environment. The average relative percentage derivations of solution approach variants are given in Table 2. Table 1 shows the overall average percentage deviation of the variants of the proposed solution approach for all test instances and restrictive factors. As seen from Table 2, Var#0, Var#2 and Var#4 outperform other compared variants in view of solution quality. Var#0 and Var#4 include **Resequencing phase** and **Solution construction phase**. Var#2 includes only **Solution construction phase** and it does not reorder tardy jobs in each machine. The common feature of effective variants of the solution approach is **Solution construction phase** and the common feature of ineffective variants (Var#1 and Var#3) is the random initial solution generation at the beginning of the algorithm. In Table 2, the best value for each restrictive factor is marked with bold font.

As seen from Table 2, the most effective component of the solution approach is **Solution construction phase** that increases/decreases the number of early/tardy jobs before the solution improvement phase of the algorithm. We do an ANOVA test with a 95% confidence level for measuring the effects of variants on the solution quality of the solution approach. The factors and details of the ANOVA test are given in Fig. 2. As seen from our ANOVA test, all factors have a significant difference with a 95% confidence level and all factors' P-values are less than 0.05 and equal to 0. Thus, we may say that selecting the right variant of the solution approach has an impact on the solution quality of the algorithm.

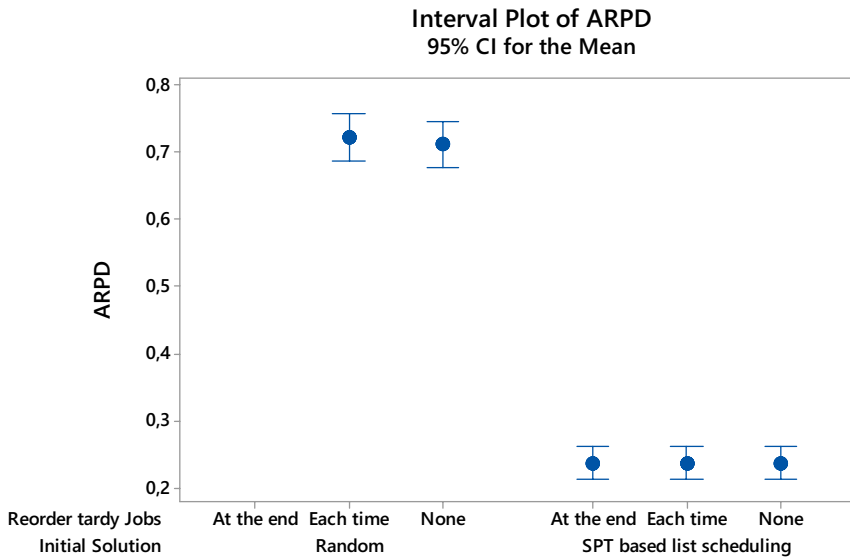


Fig. 3 The interval plot of ARPD values of the variants with different components

It seems that **Resequencing phase** in the proposed solution approach doesn't work as expected. Even though we prove that ordering only tardy jobs according to the SPT rule is an optimal policy for the investigated problem in Sect. 3, the experiment in this section does not validate that ordering tardy jobs in increasing order of their processing time assures better solutions. This can be also deduced from Fig. 3 which shows the interval plot of ARPD values of the variants with different components. As the other component of the proposed solution approach, **Job swapping phase** within a nested loop searches almost all possible sequences per machine and it does the same thing that is intended within **Resequencing phase**. If the algorithm executes fewer iterations without a nested loop that gives almost all swap alternatives, the impact of **Resequencing phase** on the solution quality will be seen more obviously.

If the investigated problem was to minimize only earliness/tardiness cost, then the optimal sequence per machine would have a V-shaped property (please see Toksari and Guner [30]). V-shaped property indicates that the optimal sequence of early (tardy) jobs must be ordered using the weighted longest processing times (weighted shortest processing time) dispatching rule. The general well-known properties of an optimal solution for the single machine weighted earliness/tardiness scheduling problem with a common due date were presented by

Panwalkar et al. [31]. Recent studies (Arık [19, 32], Arık et al. [18]) use this property to generate efficient algorithms. Since our investigated problem is to minimize the total cost including completion times and earliness/tardiness of jobs, the V-shaped property and strategies dependent on it do not fit well with the investigated problem. Therefore, new policies must be developed for the problem. The discovered policies referred to Sect. 2 are as follows:

- Increasing the number of early jobs while decreasing the number of tardy jobs,
- Sequencing tardy jobs in increasing order of their processing times in machines,
- Distributing tardy jobs on machines with equal tardy workloads without increasing the cost.

Using these strategies, a heuristic algorithm is developed and its efficiency is tested against two other existing algorithms (Biskup and Cheng [1] and Arık and Toksarı [2]) in the previous section. In the comparison in the previous section, two different strategies are compared. There are (1) disturbing numbers of jobs evenly to machines and (2) balancing the workload of machines. The compared algorithms of Biskup and Cheng [1] and Arık and Toksarı [2] use the first strategy and the experimental study shows that the proposed heuristic with the second strategy and optimal policies outperforms its rivals.

Lots of variants of parallel machine scheduling problems with common due dates, earliness/ tardiness penalties, and other characteristics have been investigated in the literature, but the investigated problem has not attention from researchers. According to reported literature, Biskup and Cheng [1] only studied the problem. In this study, an efficient heuristic algorithm using optimal policies of the problem is proposed and outperforms previous algorithms. Since the problem can be assumed new, there are lot of potential to extend it considering other scheduling characteristics like distinct weights, fuzzy/grey processing times, learning/deterioration effects, and batch processing. Also, the problem can be investigated by considering other machine environments such as flow-shop. There are so many extension possibilities because the problem considering both completion times and earliness/tardiness cost in the objective function can be associated with so many real-life examples. For instance, let us have a manufacturing company that rents machines/equipment in parallel to produce goods/services as jobs for these machines/equipment and this company has to follow a Just-In-Time management for its manufacturing since there are penalty costs associated with earliness and tardiness of jobs. In that sense, the company has to consider both completion times and the earliness/tardiness of jobs. The company has to consider the completion times because it pays for rental machines in its facilities. So many examples can be associated with the investigated problem.

Despite the existence of extensions such as fuzzy and intuitive fuzzy decision-making research problems, it is of great importance to examine operational research problems in a crisp environment. While fuzzy and intuitive approaches address uncertainties and non-precise data, emphasizing clear environments allows for a strong foundational understanding. This necessity arises from the need to establish a solid framework and foundational understanding before delving into more complex, nuanced extensions. Operational research problems in a precise environment serve the fundamental task of providing clear solutions and establishing foundational methodologies that pave the way for further research in fuzzy and intuitive fuzzy areas. The motivation behind investigating operational research problems in a crisp environment is multifaceted. Firstly, a comprehensive understanding of clear models helps to establish a strong conceptual foundation and enables researchers to grasp the underlying principles and techniques. This foundational understanding not only facilitates the development of more advanced models but also sheds light

on the advantages and limitations of each approach by enabling a comprehensive comparison between clear and non-clear methodologies. Furthermore, examining problems in a clear environment serves as an important step in improving methodologies, increasing computational efficiency, and identifying scenarios where clear models may be more suitable or effective than their fuzzy counterparts. By elucidating the importance of vibrant environments in operational research, studies (Kumar [33–37]) have paved the way for a more nuanced, comparative analysis that enriches the depth and applicability of the field.

The study can be moved in uncertain environments like fuzzy, grey, and/or stochastic. For fuzzification of scheduling parameters like processing time, due dates, and weights, the papers (Toksarı and Arık [38, 39], Arık and Toksarı [2, 40–42], Rostami et al. [43], Arık [44, 45]) can be referred. With fuzzy parameters and decision variables, the arithmetic operations (See Zadeh [46]) must be also done within fuzzy set theory. The solution methodology using these arithmetic operations can be mathematical modeling or approximation methods. For fuzzy mathematical modeling studies of Allahviranloo et al. [47], Kumar et al. [48, 49], Jayalakshmi et al. [50], Lai and Hwang [51], Fullér [52] and Liu and Iwamura [53] can be good guides for readers.

5 Conclusion

5.1 Research conclusion

In this study, we investigate unrelated parallel machine scheduling problems with the restrictive common due date where the objective is to minimize the sum of completion times, earliness durations, and tardiness durations of jobs. We show some properties of the problem. Furthermore, we showed three optimal policies for the problem. The first optimal policy is to increase the number of early jobs in the schedule as much as possible. The second optimal policy is to use the SPT dispatching rule for assigning tardy jobs. The third optimal policy is to distribute tardy jobs on machines with equal tardy workloads without increasing the cost. By using these policies of the problem, we proposed a heuristic algorithm and compared it with two existing algorithms. We generated some test instances for an experimental study. Our experimental study revealed that our proposed heuristic that uses optimal policies of the problem outperforms other existing algorithms. Furthermore, we discuss the performance of our proposed heuristic by separating it into its components. Our discussion and analysis reveal that the most effective component of the proposed heuristic is the solution construction phase that increases/decreases the number of early/tardy jobs at the beginning of the algorithm.

5.2 Theoretical implications

Since the investigated problem considers both earliness/tardiness and completion times in the objective function, the well-known properties and policies such

as V-shaped cannot be implemented. Therefore properties and policies have been drawn for this problem and also these can be implemented and improved for similar scheduling problems with both crisp and uncertain parameters like fuzzy processing times and weights.

5.3 Managerial contribution

The identification of three optimal policies provides a strategic framework for managers to make informed decisions regarding job scheduling. Managers can leverage these policies to devise scheduling strategies that prioritize early jobs, optimize tardy job assignments, and balance workload distribution across machines to minimize completion times and associated costs. Recognizing the potential impact of additional constraints like release dates and setup times prompts managers to prepare for evolving operational scenarios. Managers can anticipate and strategize for the integration of these constraints, allowing for more agile and adaptive scheduling practices. Encourages a culture of innovation within organizations by leveraging cutting-edge scheduling methodologies to stay ahead in a competitive market landscape.

5.4 Limitations and future research perspectives

The proposed properties and policies have been drawn for this problem and also these can be implemented and improved for similar scheduling problems. For future research, optimal policies and the heuristic introduced in this paper may be used in metaheuristic methods to increase solution quality. Furthermore, additional constraints such as release dates and sequence-dependent setup times can be considered within the problem and new optimal policies can be searched.

Author contributions OAA designed the investigated problem and he contributed to the design of the proposed algorithm and experimental study in the paper. OAA edited and wrote the paper. Also, OAA read and approved the final manuscript.

Funding Not applicable.

Data availability The datasets generated during and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Declarations

Conflict of interest The author declares that they have no competing interests.

References

1. Biskup, D., Cheng, T.C.E.: Multiple-machine scheduling with earliness, tardiness and completion time penalties. *Comput. Oper. Res.* **26**, 45–57 (1999). [https://doi.org/10.1016/S0305-0548\(98\)00044-6](https://doi.org/10.1016/S0305-0548(98)00044-6)

2. Arık, O.A., Toksarı, M.D.: Multi-objective fuzzy parallel machine scheduling problems under fuzzy job deterioration and learning effects. *Int. J. Prod. Res.* **56**, 2488–2505 (2018). <https://doi.org/10.1080/00207543.2017.1388932>
3. Su, L.-H.: Minimizing earliness and tardiness subject to total completion time in an identical parallel machine system. *Comput. Oper. Res.* **36**, 461–471 (2009). <https://doi.org/10.1016/j.cor.2007.09.013>
4. Sivrikaya-Şrifoğlu, F., Ulusoy, G.: Parallel machine scheduling with earliness and tardiness penalties. *Comput. Oper. Res.* **26**, 773–787 (1999). [https://doi.org/10.1016/S0305-0548\(98\)00090-2](https://doi.org/10.1016/S0305-0548(98)00090-2)
5. Weng, M.X., Lu, J., Ren, H.: Unrelated parallel machine scheduling with setup consideration and a total weighted completion time objective. *Int. J. Prod. Econ.* **70**, 215–226 (2001). [https://doi.org/10.1016/S0925-5273\(00\)00066-9](https://doi.org/10.1016/S0925-5273(00)00066-9)
6. Bilge, Ü., Kiraç, F., Kurtulan, M., Pekgün, P.: A tabu search algorithm for parallel machine total tardiness problem. *Comput. Oper. Res.* **31**, 397–414 (2004). [https://doi.org/10.1016/S0305-0548\(02\)00198-3](https://doi.org/10.1016/S0305-0548(02)00198-3)
7. Liaw, C.-F., Lin, Y.-K., Cheng, C.-Y., Chen, M.: Scheduling unrelated parallel machines to minimize total weighted tardiness. *Comput. Oper. Res.* **30**, 1777–1789 (2003). [https://doi.org/10.1016/S0305-0548\(02\)00105-3](https://doi.org/10.1016/S0305-0548(02)00105-3)
8. Radhakrishnan, S., Ventura, J.A.: Simulated annealing for parallel machine scheduling with earliness-tardiness penalties and sequence-dependent set-up times. *Int. J. Prod. Res.* **38**, 2233–2252 (2000). <https://doi.org/10.1080/00207540050028070>
9. Cheng, R., Gen, M., Tozawa, T.: Minmax earliness/tardiness scheduling in identical parallel machine system using genetic algorithms. *Comput. Ind. Eng.* **29**, 513–517 (1995). [https://doi.org/10.1016/0360-8352\(95\)00126-L](https://doi.org/10.1016/0360-8352(95)00126-L)
10. Balakrishnan, N., Kanet, J.J., Sridharan, S.V.: Early/tardy scheduling with sequence dependent setups on uniform parallel machines. *Comput. Oper. Res.* **26**, 127–141 (1999). [https://doi.org/10.1016/S0305-0548\(98\)00051-3](https://doi.org/10.1016/S0305-0548(98)00051-3)
11. Cheng, T.C., Chen, Z.-L.: Parallel-machine scheduling problems with earliness and tardiness penalties. *J. Oper. Res. Soc.* **45**, 685–695 (1994). <https://doi.org/10.1057/jors.1994.106>
12. Toksarı, M.D., Güner, E.: Minimizing the earliness/tardiness costs on parallel machine with learning effects and deteriorating jobs: a mixed nonlinear integer programming approach. *Int. J. Adv. Manuf. Technol.* **38**, 801–808 (2008). <https://doi.org/10.1007/s00170-007-1128-3>
13. Biskup, D., Herrmann, J., Gupta, J.N.D.: Scheduling identical parallel machines to minimize total tardiness. *Int. J. Prod. Econ.* **115**, 134–142 (2008). <https://doi.org/10.1016/j.ijpe.2008.04.011>
14. Shim, S.-O., Kim, Y.-D.: Scheduling on parallel identical machines to minimize total tardiness. *Eur. J. Oper. Res.* **177**, 135–146 (2007). <https://doi.org/10.1016/j.ejor.2005.09.038>
15. Bank, J., Werner, F.: Heuristic algorithms for unrelated parallel machine scheduling with a common due date, release dates, and linear earliness and tardiness penalties. *Math. Comput. Model.* **33**, 363–383 (2001). [https://doi.org/10.1016/S0895-7177\(00\)00250-8](https://doi.org/10.1016/S0895-7177(00)00250-8)
16. Yin, Y., Chen, Y., Qin, K., Wang, D.: Two-agent scheduling on unrelated parallel machines with total completion time and weighted number of tardy jobs criteria. *J. Sched.* **22**, 315–333 (2019). <https://doi.org/10.1007/s10951-018-0583-z>
17. Arık, O.A.: Comparisons of metaheuristic algorithms for unrelated parallel machine weighted earliness/tardiness scheduling problems. *Evol. Intell.* **13**, 415–425 (2020). <https://doi.org/10.1007/s12065-019-00305-7>
18. Arık, O.A., Schutten, M., Topan, E.: Weighted earliness/tardiness parallel machine scheduling problem with a common due date. *Expert Syst. Appl.* **187**, 115916 (2022). <https://doi.org/10.1016/j.eswa.2021.115916>
19. Arık, O.A.: A heuristic for single machine common due date assignment problem with different earliness/tardiness weights. *Opsearch* **60**, 1561–1574 (2023). <https://doi.org/10.1007/s12597-023-00652-1>
20. Kordmostafapour, A., Rezaeian, J., Mahdavi, I., Farjad, M.Y.: Scheduling unrelated parallel machine problem with multi-mode processing times and batch delivery cost. *Opsearch* **59**, 1438–1470 (2022). <https://doi.org/10.1007/s12597-022-00587-z>
21. Chen, Y., Xu, Y., Zhang, G., Cheng, Y.: A single machine group scheduling problem with due date assignment and position-dependent costs. *Asia-Pac. J. Oper. Res.* (2023). <https://doi.org/10.1142/S0217595923400079>

22. Geng, X.-N., Sun, X., Wang, J., Pan, L.: Scheduling on proportionate flow shop with job rejection and common due date assignment. *Comput. Ind. Eng.* **181**, 109317 (2023). <https://doi.org/10.1016/j.cie.2023.109317>
23. Pan, L., Sun, X., Wang, J.-B., Zhang, L.-H., Lv, D.-Y.: Due date assignment single-machine scheduling with delivery times, position-dependent weights and deteriorating jobs. *J. Comb. Optim.* **45**, 100 (2023). <https://doi.org/10.1007/s10878-023-01025-6>
24. Mosheiov, G., Sarig, A.: A common due-date assignment problem with job rejection on parallel uniform machines. *Int. J. Prod. Res.* (2023). <https://doi.org/10.1080/00207543.2023.2217277>
25. Wang, W.: Single-machine due-date assignment scheduling with generalized earliness-tardiness penalties including proportional setup times. *J. Appl. Math. Comput.* **68**, 1013–1031 (2022). <https://doi.org/10.1007/s12190-021-01555-4>
26. Nasrollahi, V., Moslehi, G., Reisi-Nafchi, M.: Minimizing the weighted sum of maximum earliness and maximum tardiness in a single-agent and two-agent form of a two-machine flow shop scheduling problem. *Oper. Res.* **22**, 1403–1442 (2022). <https://doi.org/10.1007/s12351-020-00577-3>
27. Falq, A.-E., Fouilhoux, P., Kedad-Sidhoum, S.: Dominance inequalities for scheduling around an unrestrictive common due date. *Eur. J. Oper. Res.* **296**, 453–464 (2022). <https://doi.org/10.1016/j.ejor.2021.03.048>
28. Qian, J., Han, H.: The due date assignment scheduling problem with the deteriorating jobs and delivery time. *J. Appl. Math. Comput.* (2021). <https://doi.org/10.1007/s12190-021-01607-9>
29. Adamopoulos, G.I., Pappis, C.P.: Scheduling under a common due-date on parallel unrelated machines. *Eur. J. Oper. Res.* **105**, 494–501 (1998). [https://doi.org/10.1016/S0377-2217\(97\)00057-X](https://doi.org/10.1016/S0377-2217(97)00057-X)
30. Toksarı, M.D., Güner, E.: The common due-date early/tardy scheduling problem on a parallel-machine under the effects of time-dependent learning and linear and nonlinear deterioration. *Expert Syst. Appl.* **37**, 92–112 (2010). <https://doi.org/10.1016/j.eswa.2009.05.014>
31. Panwalkar, S.S., Smith, M.L., Seidmann, A.: Common due date assignment to minimize total penalty for the one machine scheduling problem. *Oper. Res.* **30**, 391–399 (1982). <https://doi.org/10.1287/opre.30.2.391>
32. Arik, O.A.: Single machine earliness/tardiness scheduling problem with grey processing times and the grey common due date. *Grey Syst.* **11**, 95–109 (2021). <https://doi.org/10.1108/GS-01-2020-0010>
33. Kumar, S.P.: Computationally simple and efficient method for solving real-life mixed intuitionistic fuzzy 3D assignment problems. *Int. J. Softw. Sci. Comput. Intell. (IJSSCI)* **14**, 1–42 (2022). <https://doi.org/10.4018/IJSSCI.291715>
34. Senthil Kumar, P.: Developing a new approach to solve solid assignment problems under intuitionistic fuzzy environment. *Int. J. Fuzzy Syst. Appl.* **9**, 1–34 (2020). <https://doi.org/10.4018/IJFSA.2020010101>
35. Kumar, P.S. (2019) The PSK Method for Solving Fully Intuitionistic Fuzzy Assignment Problems With Some Software Tools. 149–202. <https://doi.org/10.4018/978-1-5225-8458-2.CH009>
36. Senthil Kumar, P.: Intuitionistic fuzzy solid assignment problems: a software-based approach. *Int. J. Syst. Assur. Eng. Manag.* **10**, 661–675 (2019). <https://doi.org/10.1007/S13198-019-00794-W/FIGURES/6>
37. Senthil Kumar, P.: Algorithms for solving the optimization problems using fuzzy and intuitionistic fuzzy set. *Int. J. Syst. Assur. Eng. Manag.* **11**, 189–222 (2020). <https://doi.org/10.1007/S13198-019-00941-3/FIGURES/9>
38. Toksarı, M.D., Arik, O.A.: Single machine scheduling problems under position-dependent fuzzy learning effect with fuzzy processing times. *J. Manuf. Syst.* (2017). <https://doi.org/10.1016/j.jmsy.2017.08.006>
39. Toksarı, M.D., Arık, O.A.: Genetic algorithm applied to the flow shop scheduling problem under effects of fuzzy learning and deterioration with a common fuzzy due date. *New Trends Issues Proc. Hum. Soc. Sci.* **4**, 306–316 (2017)
40. Arik, O.A., Toksarı, M.D.: Fuzzy chance constrained programming technique for single machine earliness/tardiness scheduling problem under effects of fuzzy learning and deterioration. *Sak. Univ. J. Sci.* **22**, 1–1 (2018). <https://doi.org/10.16984/saufenbilder.299354>
41. Arik, O.A., Toksarı, M.D.: Fuzzy parallel machine scheduling problem under fuzzy job deterioration and learning effects with fuzzy processing times. In: Ram, M. (ed.) *Advanced Fuzzy Logic Approaches in Engineering Science*. pp. 49–67. IGI Global (2019)
42. Arik, O.A., Toksarı, M.D.: Minimizing makespan with fuzzy processing times under job deterioration and learning effect. *J. Mech. Eng.* **31**, 1–17 (2020). <https://doi.org/10.4018/978-1-5225-5709-8.CH003ARINDAM>

43. Rostami, M., Pilerood, A.E., Mazdeh, M.M.: Multi-objective parallel machine scheduling problem with job deterioration and learning effect under fuzzy environment. *Comput. Ind. Eng.* **85**, 206–215 (2015). <https://doi.org/10.1016/j.cie.2015.03.022>
44. Arik, O.A.: Credibility based chance constrained programming for project scheduling with fuzzy activity durations. *Int. J. Optim. Control Theor. Appl.* **9**, 208–215 (2019). <https://doi.org/10.11121/ijocta.01.2019.00631>
45. Arik, O.A.: Dissatisfaction levels of earliness and tardiness durations by relaxing common due date on single machine scheduling problems. *J. Multidiscip. Model. Optim.* **2**, 1–15 (2019)
46. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning-I. *Inf. Sci. (Ny)* **8**, 199–249 (1975)
47. Allahviranloo, T., Lotfi, F.H., Kiasary, M.K.: Solving fully fuzzy linear programming problem by the ranking function. *Appl. Math. Sci.* **2**, 19–32 (2008)
48. Kumar, A., Kaur, J., Singh, P.: A new method for solving fully fuzzy linear programming problems. *Appl. Math. Model.* **35**, 817–823 (2011). <https://doi.org/10.1016/j.apm.2010.07.037>
49. Kumar, A., Kaur, J., Singh, P.: Fuzzy optimal solution of fully fuzzy linear programming problems with inequality constraints. *Int. J. Math. Comput. Sci.* **6**, 37–41 (2010)
50. Jayalakshmi, M., Pandian, P.: A new method for finding an optimal fuzzy solution for fully fuzzy linear programming problems. *Ijera. Com.* **2**, 247–254 (2012)
51. Lai, Y.-J., Hwang, C.-L.: A new approach to some possibilistic linear programming problems. *Fuzzy Sets Syst.* **49**, 121–133 (1992)
52. Fullér, R.: On a special type of fuzzy linear programming. *Colloq. Math. Soc. Janos Bolyai.* **50**, 511–519 (1986)
53. Liu, B., Iwamura, K.: Chance constrained programming with fuzzy parameters. *Fuzzy Sets Syst.* **94**, 227–237 (1998). [https://doi.org/10.1016/S0165-0114\(96\)00236-9](https://doi.org/10.1016/S0165-0114(96)00236-9)

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.