OPSEARCH (2023) 60:1561–1574
https://doi.org/10.1007/s12597-023-00652-1

THEORETICAL ARTICLE

A heuristic for single machine common due date assignment problem with different earliness/tardiness weights

Oğuzhan Ahmet Arik[1]

Accepted: 16 April 2023 / Published online: 28 April 2023

Abstract
This paper considers the common due date assignment for single machine weighted earliness/tardiness scheduling problem with different earliness and tardiness weights for jobs where the objective is to minimize the cost of the sum of weighted earliness/tardiness and assignment common due date. The single machine common due date assignment problem where all jobs have the same earliness/tardiness weight has a polynomial-time algorithm to solve it optimally. Furthermore, some properties for the problem where the common due date is an input have been revealed by researchers in the literature. This paper proposes a heuristic algorithm for the problem using the revealed properties of similar problems' optimal solutions such as the V-shaped property and zero-start time of the machine. The experimental study of this paper shows that the proposed heuristic finds better solutions for the problems in a reasonable time than a commercial solver has when the problem size is increased.

Keywords Common due date assignment · Single machine · Earliness · Tardiness · Scheduling

JEL Classification C61 · C44

1 Introduction

Due dates are generally considered as inputs in scheduling problems. If all jobs' due dates are the same, then this is named a scheduling problem with the common due date. Even a common due date can be an input, the decision-maker would desire to optimize the common due date to satisfy the performance criterion minimizing the cost combination of earliness/tardiness and setting a common due date. Such an

✉ Oğuzhan Ahmet Arik
oaarik@nny.edu.tr

[1] Industrial Engineering Department, Nuh Naci Yazgan University, Kayseri 38170, Turkey

optimization for the accurate common due date for the scheduling problem is named the common due date assignment problem. The effort for responding to customer orders on time by utilizing manufacturing facilities, time, and minimizing costs is the main goal of each manufacturing company. Negotiation between customers and the company might be about determining the due dates of jobs and there might be a common due date that all customers agree on. If both the customer's satisfaction and cost minimization of the company lies under setting the due dates of all jobs to a common due date, then the best way to assign that common due date is to optimize it according to each customer's objective and the company's own objective. The common due date in a scheduling problem makes some jobs early and others tardy. When a job is completed before the promised due date, such a job is called an early job. For instance, an early job is not desired because it has to be stored in a warehouse, resulting in costs for inventory and insurance. When a job is completed after the promised due date, such a job is called a tardy job. A tardy job is not desired because the manufacturer loses some of its earnings due to the contract penalties as well as some goodwill and reputation [1]. Since there is a common due date for all jobs of the customers, some of these jobs will be completed before/after the assigned common due date and they will be early/tardy. Earliness/tardiness has some costs for the company. Since these jobs might belong to different customers and their importance might be different as well. Earliness/tardiness unit cost for each job might be different than others. Thus earliness/tardiness weights/costs can be different within a scheduling problem with common due date consideration. The common due date assignment has been investigated by researchers for more than 30 years. Panwanker et al. [2] consider a single machine weighted earliness/tardiness scheduling problem where the common due date is a decision variable and each early/tardy job's weight is the same. In a classical scheduling triple notation, the problem with n jobs can be expressed as $1\left|d_j = d\right|\sum \alpha E_j + \sum \beta T_j + \sum \gamma d$ where j is the job index ($j \in \{1,2,3,.,n\}$), d is the common due date, γ is the penalty cost of assigning common due date, $\alpha(\beta)$ is earliness (tardiness) penalty cost for all jobs, and $E_j\left(T_j\right)$ is earliness/tardiness of job j. E_j is a positive decision variable for the problem and $E_j = \max\left(0, d - C_j\right)$ where C_j is the completion time of job j. Similarly, T_j is also a positive decision variable and $T_j = \max\left(0, C_j - d\right)$. They propose a polynomial time algorithm to solve their investigated problem and prove that their algorithm solve the problem optimally. The problem with given common due date $1\left|d_j = d\right|\sum \alpha E_j + \sum \beta T_j$ has also polynomial time algorithm [2] to solve it optimally if d is an unrestrictive common due date. In a scheduling problem with a given common due date, there are two options: (1) unrestrictive common due date where d is big enough to make all jobs completed before or at the common due date and (2) restrictive common due date where d^{res} is not big enough to make all jobs early or on-time. The same problem with restrictive common due date $1\left|d_j = d^{res}\right|\sum \alpha E_j + \sum \beta T_j$ is a NP-Hard [3] problem and an optimal solution algorithm has not been revealed by the researchers so far. Furthermore, $1\left|d_j = d^{res}\right|\sum a_j E_j + \sum \beta_j T_j$ where $a_j(\beta_j)$ is the distinct earliness/tardiness cost/weight for job j is also NP-Hard and the optimal sequence of this problem has the

V-shaped property [4–8] showing that early jobs are sequenced in decreasing order of their ratios of processing times/earliness weights and tardy jobs are sequenced in increasing order of their ratios of processing times/tardiness weights. Kanet [4] shows that there is a polynomial time algorithm to solve $1\left|d_j = d\right|\sum E_j + \sum T_j$ optimally. As understood from this short literature, if jobs in the problem have different earliness/tardiness weights, the problem becomes a NP-Hard problem.

Even most single-machine earliness/tardiness scheduling problems are still NP-Hard, the researchers have investigated polynomial-time optimal or approximation algorithms for earliness/tardiness scheduling problems with due date determination decisions. Cheng and Gupta [9] presented a survey paper for scheduling problems with due date determination decisions. They investigated the problem from the perspectives "static" and "dynamic". This classification was made by Elion [10]. Nowadays some sort of dynamic is also referred as online. Gordon et al. [11] drew an extensive framework for common due date assignment problems in their survey paper. They investigated common due assignment problems in single and parallel machine environments. They summarized the results of the algorithms and the complexity of the common due date assignment and scheduling problems. Seidmann et al. [12] investigated $1\left|d_j\right|\sum \alpha E_j + \sum \beta T_j + \sum \gamma A_j$ where $A_j = \max(0, d_j - A)$, d_j is assigned due date for job j and A is the lead time that customers consider to be reasonable and expected. In their problem, the objective function is to minimize the earliness/tardiness costs of jobs and the total time that the assigned due dates for all jobs exceeds A. They proved that there is a polynomial time algorithm to solve their problem optimally. For $1\left|d_j = d\right|\sum \alpha E_j + \sum \beta T_j + \sum \gamma d$, the optimal solution has a property that shows the assigned common due date is equal to one of the completion times of jobs. This was showed by Panwalkar et al. [2]. Cheng [13] offered an alternative proof for this property of the optimal solution by using Kuhn-Tucker's optimality conditions. Sung et al. [14] considered the problem of $1\left|d_j = d\right|\sum w_j\left(d - C_j\right)/w_j$ to minimize the total weighted mean absolute deviations of job completion times from the common due date. They determined several dominant solution properties to organize two efficient heuristic solution algorithms for their problem. Koulamas [15] proposed a polynomial time (O(n^3)) dynamic programming to solve the problem of $1\left|d_j = d\right|\sum \alpha E_j + \sum \beta T_j + \sum a_j V_j + \sum b_j U_j + \sum \gamma d$ where $V_j(U_j)$ is a binary decision variable that is 1 if job j is early(tardy) and $a_j(b_j)$ is the penalty cost of job j if it is early(tardy). Furthermore, Koulamas [15] showed that there is a linear time algorithm to solve $1\left|d_j = d\right|\sum a_j V_j + \sum b_j U_j + \sum \gamma d$ optimally. De et al. [16] showed that there is a linear-time algorithm for the problem of $1\left|d_j = d\right|\sum b_j U_j + \sum \gamma d$. Koulamas [17] proposed a linear time dynamic programming algorithm to solve of $1\left|d_j = d\right|\sum \alpha_j E_j + \sum b_j U_j + \sum \gamma d$. For the same problem's optimal solution, Kahlbacker and Cheng [18] proposed an O(n^2) polynomial-time algorithm for the problem in parallel machine environment. Li and Chen [19] proposed a polynomial-time

O(n^3) algorithm to solve the problem of $1\left|d_j = d, p_j = p\right| \sum \alpha_j E_j + \sum \beta_j T_j + \sum a_j V_j + \sum b_j U_j + \sum \gamma d$ where all jobs have the same processing time. Shabtay et al. [20] investigated the problems of $1\left|d_j = d\right| \sum \alpha_j Q_j + \sum \beta_j Y_j + \sum \gamma d$ and $1\left|d \le d_j \le d + \delta\right| \sum \alpha_j Q_j + \sum \beta_j Y_j + \sum \gamma d + \sum \phi \delta$ where $Q_j = \min\left(p_j, max\left(0, d - C_j + p_j\right)\right)\forall j$, $Y_j = \min\left(p_j, T_j\right)\forall j$, δ is the time interval for the common due date window and ϕ is the penalty of the common due date window. Their problems' objective functions are to minimize the penalty cost of early and tardy work, the common due date assignment cost, and the cost of the common due date window. They showed polynomially solvable cases of their problem when the common due date and the common due date window are unbounded. They also stated that their problem becomes NP-Hard when it is bounded. Yeung et al. [21] proposed an O(nlogn) polynomial-time algorithm to find the optimal solution of the problem of $1\left|d \le d_j \le d + \delta\right| \sum \alpha E_j + \sum \beta T_j + \sum a V_j + \sum b U_j + \sum \gamma d$ where $a(b)$ is the penalty cost of any job if it is early(tardy). We summarize the literature review for due date assignment problems in Table 1 by adding other important studies in multi machine environment. As seen in Table 1, $\sum_{j=1}^{n} \left(\alpha_j E_j + \beta_j T_j + \gamma d\right)$ problem for all machine environments has not been studied by researchers. As long as we know from the reported literature, Mosheiov and Yovel [27] only investigated $\sum_{j=1}^{n} \left(\alpha_j E_j + \beta_j T_j + \gamma d\right)$ problem with unit processing times ($P_j = 1\forall j$) by proposing a heuristic approach. The rest of studies considered $\sum \alpha E_j + \sum \beta T_j + \sum \gamma d$ problem or its variants by proosing heuristics and polynomial time algorithms. Since $\sum_{j=1}^{n} \left(\alpha_j E_j + \beta_j T_j + \gamma d\right)$ problem is NP-Hard, it needs still an approximate solution approach for finding a near-optimal solution. As seen from the literature given in Table 1, there are so few studies considering metaheuristic for the problem when the common due date is a decision variable, most of the studies in Table 1 have proposed heuristics for NP hard problem with thecommon due date that is a decision variable.

In this study, we consider $1\left|d_j = d\right| \sum \alpha_j E_j + \sum \beta_j T_j + \sum \gamma d$ where each job's weights for earliness and tardiness are different from others. By using some properties of optimal solutions of similar problems such as $1\left|d_j = d^{res}\right| \sum a_j E_j + \sum \beta_j T_j$, $1\left|d_j = d\right| \sum E_j + \sum T_j$, $1\left|d_j = d\right| \sum \alpha E_j + \sum \beta T_j$ and $1\left|d_j = d\right| \sum \alpha E_j + \sum \beta T_j + \sum \gamma d$, we propose a heuristic for single machine weighted earliness/tardiness scheduling and common due date assignment problem. The remainder of the paper is as follows: Sect. 2 presents a mathematical model for the problem. Section 3 introduces our proposed heuristic for the problem. Section 4 presents an experimental study to compare our proposed heuristic with an exact solution approach. Section 5 draws a conclusion and future projection for the reader.

Table 1 Literature review

Authors	Objective func.	Machine environment	Solution approach
Panwanker et al. [2]	$\sum \alpha E_j + \sum \beta T_j + \sum \gamma d$	Single machine	Polynomial time algorithm
Hall and Posner [3]	$\sum \alpha E_j + \sum \beta T_j$	Single machine	Heuristic
Kanet [4]	$\sum E_j + \sum T_j$	Single machine	Polynomial time algorithm
Seidmann et al. [12]	$\sum \alpha E_j + \sum \beta T_j + \sum \gamma A_j$	Single machine	Polynomial time algorithm
Sung et al. [14]	$\sum w_j \left(d - C_j \right) / w_j$	Single machine	Heuristic
Koulamas [15]	$\sum \alpha E_j + \sum \beta T_j + \sum a_j V_j + \sum b_j U_j + \sum \gamma d$	Single machine	Polynomial time algorithm
Koulamas [15]	$\sum a_j V_j + \sum b_j U_j + \sum \gamma d$	Single machine	Polynomial time algorithm
De et al. [16]	$\sum b_j U_j + \sum \gamma d$	Single machine	Polynomial time algorithm
Kahlbacker and Cheng [18]	$\sum \alpha_j E_j + \sum b_j U_j + \sum \gamma d$	Parallel machine	Polynomial time algorithm
Li and Chen [19]	$\sum \alpha_j E_j + \sum \beta_j T_j + \sum a_j V_j + \sum b_j U_j + \sum \gamma d$	Single machine	Polynomial time algorithm
Shabtay et al. [20]	$\sum \alpha_j Q_j + \sum \beta_j Y_j + \sum \gamma d$	Single machine	Heuristic
Yeung et al. [21]	$\sum \alpha E_j + \sum \beta T_j + \sum a V_j + \sum b U_j + \sum \gamma d$	Single machine	Polynomial time algorithm
De et al. [22]	$\sum \alpha E_j + \sum \beta T_j + \sum \gamma d$	Identical parallel machine	Dynamic Programming
Cheng and Chen [23]	$\sum \alpha E_j + \sum \beta T_j + \sum \gamma d$	Identical parallel machine	Heuristic
Diamond and Cheng [24]	$\sum \alpha E_j + \sum \beta T_j + \sum \gamma d$	Identical parallel machine	Heuristic
Mosheiov [25]	$\sum_{j=1}^{n} \left(\alpha \max \left(E_j \right) + \beta \max \left(T_j \right) + \gamma d \right)$	Identical parallel machine	Heuristic
Xiao and Li [26]	$\sum \alpha E_j + \sum \beta T_j + \sum \gamma d$	Identical parallel machine	Heuristic
Mosheiov and Yovel [27]	$\sum_{j=1}^{n} \left(\alpha_j E_j + \beta_j T_j + \gamma d \right)$	Identical parallel machine where $p_j = 1 \forall j$	Heuristic
Min and Cheng [28]	$\sum \alpha E_j + \sum \beta T_j + \sum \gamma d$	Identical parallel machine	Genetic Algorithm
Drobouchevitch and Sidney [29]	$\sum \alpha E_j + \sum \beta T_j + \sum \gamma d$	Unrelated parallel machine	Heuristic
Kim et al. [30]	$\sum \alpha E_j + \sum \beta T_j + \sum \gamma d$	Identical parallel machine	Simulated Annealing, Tabu Search
Xiong et al. [31]	$\sum \alpha E_j + \sum \beta T_j + \sum \gamma d$	Single machine	Pseudo-polynomial algorithm
Ng Daniel et al. [32]	$\sum \alpha E_j + \sum \beta T_j + \sum \gamma d + \mathrm{g(x)}$	Single machine	Polynomial time algorithm

2 Mathematical model

In this section, we present a mixed-integer linear programming model for our investigated problem. The model was originally proposed by Keha et al. [33]. We just add common due date assignment consideration into the model. There are n jobs to be processed in a single machine. Each job has its own earliness and tardiness weights in the objective function. All jobs have the same common due date which is a decision variable that is a part of the performance criterion of the problem. Determination of this common due date has a constant cost for all jobs in the objective function. We have some assumptions for the investigated problem as follows: the machine can only process at most one job at the same time, the machine is ready to process jobs at zero start time, processing times of jobs, and costs for earliness and tardiness are non-negative and preemption is not allowed. Indices, parameters, decision variables, and equations/inequations of the model are as follows:

Indices:

j, k: indices of jobs ($j \in \{1,2,\dots,n\}$ and $k \in \{1,2,\dots,n\}$)

Parameters:

p_j : processing time of job j

α_j : earliness unit cost/weight of job j

β_j : tardiness unit cost/weight of job j

γ : common due date assignment cost for each job

M: arbitrary large number ($M = \sum_{j=1}^{n} p_j$)

Decision variables:

C_j : completion time of job j

E_j : earliness duration of job j

T_j : tardiness duration of job j

d : common due date

y_{jk} : binary decision variable for determining which job is processed before another job. If job j is processed before job k in the machine, then it is 1. Otherwise, it is 0.

Objective function:

$$\text{Min} \sum_j \alpha_j E_j + \sum_j \beta_j T_j + n\gamma d \tag{1}$$

S.t.:

$$C_j \geq p_j \quad \forall j \tag{2}$$

$$C_j + p_k \geq C_k + M\left(1 - y_{jk}\right) \quad \forall j, k \textit{ and } j < k \tag{3}$$

$$C_k + p_j \geq C_j + My_{jk} \quad \forall j, k \textit{ and } j < k \tag{4}$$

$$C_j + E_j - T_j = d \quad \forall j \tag{5}$$

$$y_{jk} \in \{0{,}1\} \quad \forall j, k \tag{6}$$

$$C_j, E_j, T_j \geq 0 \quad \forall j \tag{7}$$

The objective function (1) is to minimize the cost combination of the weighted earliness/tardiness costs and setting common due date costs. Constraint (2) assures that the completion time of job j is greater than or equal to its processing time. Constraints (3–4) are disjunctive constraints assuring that either job j is processed before job k or job j is processed after job k. These constraints are for all job pairs. For any jobs j and k, there must be only two potential job sequences: (1) the sequence where the job j is processed before job k or (2) the sequence where job j is processed after job k. Therefore if $y_{jk} = 1$, then y_{kj}must be 0 or the other way around. Constraint (5) shows that job j can be completed as early, on time, or tardy. If $C_j = d$ then job j is an on-time job and $E_j(T_j)$ is 0. Else if $C_j < d$ $(C_j > d)$, then job j is early (tardy) and $E_j = d - C_j$ $(T_j = C_j - d)$. For any job j, there is no possibility of both E_j and T_j are greater than zero since C_j is either equal to d or less (greater) than d.

3 The proposed heuristic

In this section, we propose a heuristic for $1\left|d_j = d\right|\sum \alpha_j E_j + \sum \beta_j T_j + \sum \gamma d$ by using some properties of optimal solutions of similar problems such as $1\left|d_j = d^{res}\right|\sum a_j E_j + \sum \beta_j T_j$, $1\left|d_j = d\right|\sum E_j + \sum T_j$, $1\left|d_j = d\right|\sum \alpha E_j + \sum \beta T_j$ and $1\left|d_j = d\right|\sum \alpha E_j + \sum \beta T_j + \sum \gamma d$. These properties are as follows:

- For the problem of $1\left|d_j = d\right|\sum \alpha E_j + \sum \beta T_j + \sum \gamma d$, the optimal common due date must be less than or equal to the sum of processing times $d^* \leq \sum_j P_j$ where $d^*$is the optimal common due date. Therefore, the optimal common due date is a restrictive common due date. This property (See Panwanker et al. [2]) can be applied to our investigated problem.
- For the problem of $1\left|d_j = d\right|\sum \alpha E_j + \sum \beta T_j + \sum \gamma d$, if $\gamma \geq \beta$ then $d^* = 0$ and the job sequence arranged with the shortest processing time (SPT) rule is the optimal sequence. This property (See Panwanker et al. [2]) can be applied to our investigated problem.
- For any job sequence of the problem of $1\left|d_j = d\right|\sum \alpha E_j + \sum \beta T_j + \sum \gamma d$, there exists an optimal common due date that is equal to one of the completion times of jobs. This property (See Panwanker et al. [2]) can be applied to our investigated problem.
- For any job sequence of the problem of $1\left|d_j = d\right|\sum \alpha E_j + \sum \beta T_j + \sum \gamma d$, there exists an optimal common due date equal to $C_{[K]}$, where K is the smallest integral value greater than or equal to $n(\beta - \gamma)/(\alpha + \beta)$. This property (See Panwanker et al. [2]) cannot be applied to our investigated problem since $\alpha_j(\beta_j)$ is distinct for each job.

- For problems of $1\left|d_j = d^{res}\right|\sum a_jE_j + \sum \beta_jT_j$, $1\left|d_j = d\right|\sum E_j + \sum T_j$, and $1\left|d_j = d\right|\sum \alpha E_j + \sum \beta T_j$, the common due date is an input. The optimal job sequences for these problems have the V-Shaped property. This property (See Kanet [4] and Arık [8]) of the optimal solution shows that the early jobs in the optimal sequence are sequenced in decreased order of P_j/a_j values and the tardy jobs are sequences in increased order of P_j/β_j values. Thus the optimal sequence can be arranged with the weighted shortest processing time (WSPT) and the weighted longest processing time (WLPT) rules. This property can be also applied to our investigated problem.
- For problems of $1\left|d_j = d^{res}\right|\sum a_jE_j + \sum \beta_jT_j$, $1\left|d_j = d\right|\sum E_j + \sum T_j$, and $1\left|d_j = d\right|\sum \alpha E_j + \sum \beta T_j$, the start time of the first job in the sequence can be greater than zero and none of the completion times of jobs can be equal to the common due date. This property (See Kanet [4] and Arık [1]) cannot be applied to our problem.

The problem of $1\left|d_j = d\right|\sum \alpha_jE_j + \sum \beta_jT_j + \sum \gamma d$ have structural similarities with other problems of $1\left|d_j = d^{res}\right|\sum a_jE_j + \sum \beta_jT_j$, $1\left|d_j = d\right|\sum E_j + \sum T_j$, $1\left|d_j = d\right|\sum \alpha E_j + \sum \beta T_j$ and $1\left|d_j = d\right|\sum \alpha E_j + \sum \beta T_j + \sum \gamma d$. We use some of above properties in our proposed heuristic as follows:

1. The zero start time of the machine,
2. The common due date that is less than or equal to the sum of processing times,
3. The V-shaped property of the optimal solution, and.
4. (4)The common due date is equal to one of the completion times of jobs.

We apply these properties in our heuristics as follows:

1. We set always the start time of the first job in the sequence zero in our proposed heuristics,
2. We search the common due date between zero and the sum of processing times,
3. By using the V-shaped property, we order early jobs with the WLPT rule and tardy jobs with the WSPT rule around the determined common due date,
4. We set the common due date to the completion time of the job in position r where ($r \in \{0,1,2,3,\ldots,n\}$) and we increase r by one at each iteration.

In our proposed heuristic, we search the common due date from zero. When$d = 0$, the best sequence of the problem can be obtained with the WSPT rule. Since $d = 0$, all jobs are tardy and their best sequence can be obtained by using the WSPT rule. In case there are some early jobs because $d > 0$ and the start time of the machine is non-zero, then early jobs must be ordered with the WLPT rule and the remaining tardy jobs must be ordered with the WSPT rule. This is our initial solution (π) and we record it as the best solution (π^*) for now. Within a for loop, we test each job at the first position of solution π by increasing the position number ($r \in \{0,1,2,3 \ldots, n\}$) of the completion time of the job by one. We set $d = C_{[r]}$ for each r (note that $r = 0$implies that $d = 0$). For each increment of r, we order early(tardy) jobs with the WLPT(WSPT). When we find a better candidate solution π'' than π^*, we replace π^* with π''. Then, the remaining search goes on with new π^* by using the V-shaped property and the zero-start time of the machine until the stopping criterion is satisfied. For this algorithm, we use a predetermined allowed elapsed time limitation as a stopping criterion. The pseudo-code of the proposed heuristic is given in Fig. 1. As seen in Fig. 1, the complexity of our proposed heuristic is O(n^3).

4 Experimental study

In this section, we test our proposed heuristic's performance in view of solution quality by comparing it with solutions obtained from a commercial solver. We use a stopping criterion that allows running the algorithm until n seconds or the algorithm ends its all steps to solve the problem. n seconds means that the algorithm will use the number of jobs in the problem as a stopping criterion of the solution method in elapsed seconds. As seen from the literature review, there is no heuristic, metaheuristic or pseudo-polynomial algorithm for $1\left|d_j = d\right| \sum \alpha_j E_j + \sum \beta_j T_j + \sum \gamma d$ problem. Even there are some approximation algorithms for $\sum \alpha E_j + \sum \beta T_j + \sum \gamma d$ problems and their variants, these algorithms are not applicable because of the same earliness/tardiness weight consideration in those problems. Therefore, we use the mathematical modelling approach and an commercial solver to evaluate our proposed heuristic's performance. We use the CPLEX solver in GAMS 28.2 commercial software to code and solve the mathematical model introduced in this study. The commercial solver's solution time is also restricted until n seconds. We use a benchmark data set of common due date scheduling problems. This data set was originally generated by Biskup and Feldmann [34] for the problem where the common due date is a parameter. In this study, we use this data set for our problem where the common due date is a decision variable. This data set includes seven groups of test problems. These groups have 10, 20, 50, 100, 200, 500, and 1000 jobs. Each group has 10 test instances. The data set can be retrieved from http://people.brunel.ac.uk/~mastjjb/jeb/orlib/schinfo.html. This data set doesn't have γ values for test instances so we use 6 γ values where $\gamma \in \{0,4,8,12,16,20\}$ for each instance. We use γ values between 0 and 20 since other parameters in test instances such as P_j, α_j and β_j are generated randomly between 1 and 20 by Biskup and Feldmann [34]. There are 7 n values and 6 γ values for our test groups and each group has 10 test instances so there are 420 test problems in our experimental study. Both our proposed heuristic and the commercial solver solve each instance once. We code our proposed heuristic in Visual Studio 2019 by using C# programming language. Furthermore, we use C# programing language to generate GAMS file for all test instances. We call GAMS library by using C# programming language in Visual

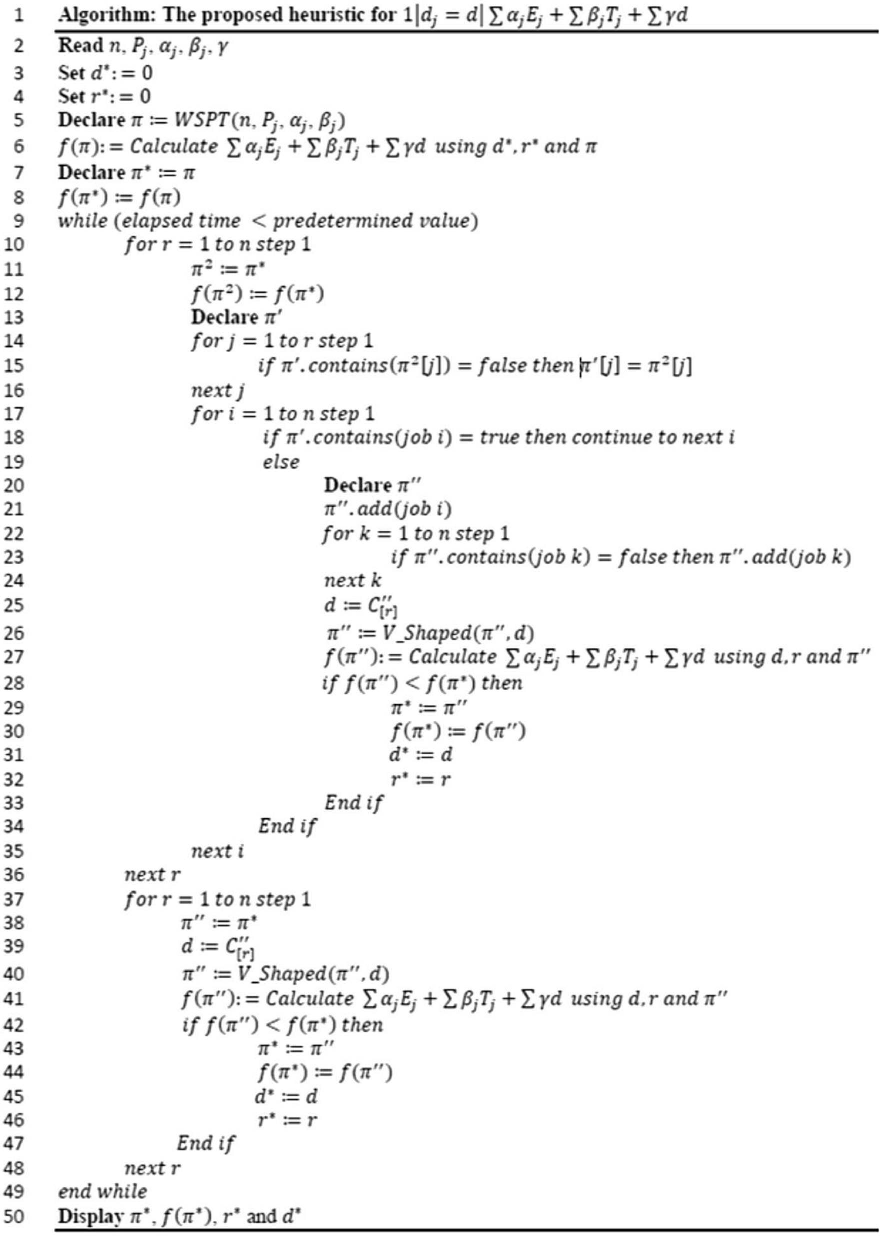
Table 2 How many times our proposed heuristic finds the better solutions

n / γ	0	4	8	12	16	20
10	7	1	1	1	1	1
20	1	0	0	0	0	0
50	7	7	4	5	5	3
100	10	10	9	10	10	10
200	10	10	10	10	10	10
500	10	10	10	10	10	10
1000	10	10	10	10	10	10

Studio to solve test instances. Each solution's objective function, solution times in milliseconds, and the common due date value are recorded. Our experimental study has been done on a workstation desktop having an Intel Xeon E-2136 CPU @ 3.30 GHz and 16 GB RAM. Our proposed heuristic solves test problems in a shorter time by closing more the optimal solution than the commercial solver does. Even the commercial solver could not find a solution for 99 test instances where $n \geq 200$ until its solution time reaches n seconds, the proposed heuristic solves all test instances. For small-size instances where $n \leq 50$, the commercial solver may find better solutions than our proposed heuristic finds. Table 2 shows how many times our proposed heuristic finds better solutions than the commercial solver within time limitations. For all test problems, the proposed heuristic's solution quality is 30.74% better than solutions obtained with GAMS. When $n = 10$ and $\gamma = 0$, the proposed heuristic finds 7 out of 10 best solutions for test instances. When $n = 20$ and $\gamma = 8$, the proposed heuristic finds none of 10 best

Table 3 Average solution times of the proposed heuristic and the commercial solver for test instance group

Method	n/γ	0	4	8	12	16	20
Heuristic	10	0.0073	0.0012	0.0014	0.0013	0.0013	0.0015
	20	0.0036	0.0025	0.0027	0.0036	0.0024	0.0026
	50	0.0238	0.0186	0.0194	0.0184	0.0192	0.0182
	100	0.1523	0.1449	0.1435	0.1440	0.1447	0.1448
	200	1.4010	1.4058	1.4025	1.3985	1.4051	1.4036
	500	35.8555	35.5052	35.4952	35.4644	35.5365	35.6059
	1000	459.9797	464.5944	464.1628	464.6156	464.8264	464.7775
Gams	10	1.3448	2.2815	1.8015	0.7861	0.7845	0.7943
	20	20.1625	20.1578	20.1630	20.1574	20.1629	20.1654
	50	50.4269	50.2991	50.2972	50.2900	50.2825	50.2964
	100	100.3657	100.4070	100.4001	100.4019	100.3658	100.3933
	200	201.0243	200.8899	200.9466	200.8416	200.9014	200.9393
	500	502.3755	502.3962	503.4606	502.5705	502.5700	502.7197
	1000	1008.1621	1006.6984	1013.9403	1008.0730	1007.4889	1007.4585



```
Algorithm: The proposed heuristic for $1|d_j = d|\sum\alpha_j E_j + \sum\beta_j T_j + \sum\gamma d$
Read $n, P_j, \alpha_j, \beta_j, \gamma$
Set $d^*: = 0$
Set $r^*: = 0$
Declare $\pi := WSPT(n, P_j, \alpha_j, \beta_j)$
$f(\pi): = Calculate\ \sum\alpha_j E_j + \sum\beta_j T_j + \sum\gamma d\ using\ d^*, r^*\ and\ \pi$
Declare $\pi^* := \pi$
$f(\pi^*) := f(\pi)$
while (elapsed time < predetermined value)
        for $r = 1$ to $n$ step 1
                $\pi^2 := \pi^*$
                $f(\pi^2) := f(\pi^*)$
                Declare $\pi'$
                for $j = 1$ to $r$ step 1
                        if $\pi'.contains(\pi^2[j]) = false$ then $|\pi'[j] = \pi^2[j]$
                next $j$
                for $i = 1$ to $n$ step 1
                        if $\pi'.contains(job\ i) = true$ then continue to next $i$
                        else
                                Declare $\pi''$
                                $\pi''.add(job\ i)$
                                for $k = 1$ to $n$ step 1
                                        if $\pi''.contains(job\ k) = false$ then $\pi''.add(job\ k)$
                                next $k$
                                $d := C''_{[r]}$
                                $\pi'' := V\_Shaped(\pi'', d)$
                                $f(\pi''): = Calculate\ \sum\alpha_j E_j + \sum\beta_j T_j + \sum\gamma d\ using\ d, r\ and\ \pi''$
                                if $f(\pi'') < f(\pi^*)$ then
                                        $\pi^* := \pi''$
                                        $f(\pi^*) := f(\pi'')$
                                        $d^* := d$
                                        $r^* := r$
                                End if
                        End if
                next $i$
        next $r$
        for $r = 1$ to $n$ step 1
                $\pi'' := \pi^*$
                $d := C''_{[r]}$
                $\pi'' := V\_Shaped(\pi'', d)$
                $f(\pi''): = Calculate\ \sum\alpha_j E_j + \sum\beta_j T_j + \sum\gamma d\ using\ d, r\ and\ \pi''$
                if $f(\pi'') < f(\pi^*)$ then
                        $\pi^* := \pi''$
                        $f(\pi^*) := f(\pi'')$
                        $d^* := d$
                        $r^* := r$
                End if
        next $r$
end while
Display $\pi^*, f(\pi^*), r^*$ and $d^*$
```

Fig. 1 The pseudo-code for the proposed heuristic for $1\left|d_j = d\right|\sum\alpha_j E_j + \sum\beta_j T_j + \sum\gamma d$

solutions for test instances. When $n \leq 50$, the proposed heuristic does not outperform the commercial solver and it just finds 44 of 180 best solutions. When $n \geq 100$, the proposed heuristic obtains better solutions for test instances except for one instance. Table 3 shows the average solution times of compared solution approaches for each test group. As seen in Table 3, the proposed heuristic consumes less time than the commercial solver to obtain a solution. If $n \geq 100$ then these solutions are better than the commercial solver finds and they are obtained in a shorter time with the proposed heuristic. We use the same stopping criterion for both methods to be fair in comparison. Someone may wonder what the solution quality of GAMS will be if there is no time limitation. We test our mathematical model using GAMS with its default settings to solve problems optimally. Most of the problems where $n \geq 200$ could not be solved optimally and the solver did not report solutions for these problems. The solution quality of GAMS for these problems when it works with its default settings is still worse than the solution quality of the proposed heuristic. Since our proposed heuristic is an $O(n^3)$ algorithm, we use *n* as a parameter in our stopping criterion.

5 Conclusion

This study investigates a heuristic algorithm for single machine weighted earliness/tardiness scheduling and common due date assignment problem where earliness/tardiness weights of jobs are different. The objective function of the problem is to minimize the cost combination of weighted earliness/tardiness and common due date assignment cost. By using several properties such as the V-shaped property, the zero-start time, and the common due date that equals to a completion time of a job in the sequence of similar problems, we develop a heuristic solution approach. Since there is no similar heuristic or solution method in the literature, we test our proposed heuristic against the mathematical model coded in a commercial solver. For a fair comparison, we use the same computer environment and solution time restriction in our experimental study. When the number of jobs increases, the efficiency of our proposed heuristic increases and it finds better solutions in a shorter time than the commercial solver finds. For future research, this heuristic can be extended into a parallel machine environment with additional objective functions such as the total tardy work and the total number of tardy jobs. Moreover, solution of this heuristic can be used as an initial solution for other solution improvement heuristics or metaheuristics such as genetic algorithm, simulated annealing and tabu search. Furthermore, additional constraints such as sequence-dependent setup times and release times can be considered in a future extension for the proposed heuristic.

Acknowledgements Not applicable.

Author contributions OAA designed the investigated problem and he contributed to the design of the proposed algorithm and experimental study in the paper. OAA also edited and wrote the paper. 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 on reasonable request.

Declarations

Conflict of interest The author declares that they have no conflict of interest.

Ethical approval and consent to participate Not applicable.

Consent for publication Not applicable.

References

1. 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
2. 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
3. Hall, N.G., Posner, M.E.: Earliness-tardiness scheduling problems, I: weighted deviation of completion times about a common due date. Oper. Res. **39**, 836–846 (1991). https://doi.org/10.1287/opre.39.5.836
4. Kanet, J.J.: Minimizing the average deviation of job completion times about a common, due date. Nav. Res. Logist. Q. **28**, 643–651 (1981). https://doi.org/10.1002/nav.3800280411
5. Raghavachari, M., Zammouri, M.: Single machine scheduling with coefficient of variation minimization. Eur. J. Oper. Res. **62**, 302–310 (1992). https://doi.org/10.1016/0377-2217(92)90120-X
6. Bector, C.R., Gupta, Y.P., Gupta, M.C.: V-shape property of optimal sequence of jobs about a common due date on a single machine. Comput. Oper. Res. **16**, 583–588 (1989). https://doi.org/10.1016/0305-0548(89)90043-9
7. Raghavachari, M.: A V-shape property of optimal schedule of jobs about a common due date. Eur. J. Oper. Res. **23**, 401–402 (1986). https://doi.org/10.1016/0377-2217(86)90306-1
8. Arık, O.A.: Single machine earliness/tardiness scheduling problem with grey processing times and the grey common due date. Grey Syst. Theory Appl. **11**, 95–109 (2021). https://doi.org/10.1108/gs-01-2020-0010
9. Cheng, T.C.E., Gupta, M.C.: Survey of scheduling research involving due date determination decisions. Eur. J. Oper. Res. **38**, 156–166 (1989). https://doi.org/10.1016/0377-2217(89)90100-8
10. Elion, S.: Production scheduling. In: Haley, K.B. (ed.) Operations Research '78, pp. 237–266. North-Holland Publishing Company, Amsterdam (1979)
11. Gordon, V., Proth, J.-M., Chu, C.: A survey of the state-of-the-art of common due date assignment and scheduling research. Eur. J. Oper. Res. **139**, 1–25 (2002). https://doi.org/10.1016/S0377-2217(01)00181-3
12. Seidmann, A., Panwalkar, S.S., Smith, M.L.: Optimal assignment of due-dates for a single processor scheduling problem. Int. J. Prod. Res. **19**, 393–399 (1981). https://doi.org/10.1080/00207548108956667
13. Cheng, T.C.E.: An alternative proof of optimality for the common due-date assignment problem. Eur. J. Oper. Res. **37**, 250–253 (1988). https://doi.org/10.1016/0377-2217(88)90334-7
14. Sung, C.S., Min, J.I., Park, C.K.: Heuristic algorithms for a single-machine common due date assignment under earliness/tardiness measure. J. Oper. Res. Soc. Jpn. **36**, 121–133 (1993)
15. Koulamas, C.: Common due date assignment with generalized earliness/tardiness penalties. Comput. Ind. Eng. **109**, 79–83 (2017). https://doi.org/10.1016/j.cie.2017.04.040
16. De, P., Ghosh, J.B., Wells, C.E.: Optimal delivery time quotation and order sequencing. Decis. Sci. **22**, 379–390 (1991). https://doi.org/10.1111/J.1540-5915.1991.TB00353.X
17. Koulamas, C.: A unified solution approach for the due date assignment problem with tardy jobs. Int. J. Prod. Econ. **132**, 292–295 (2011). https://doi.org/10.1016/J.IJPE.2011.04.023

18. Kahlbacher, H.G., Cheng, T.C.E.: Parallel machine scheduling to minimize costs for earliness and number of tardy jobs. Discrete Appl. Math. **47**, 139–164 (1993). https://doi.org/10.1016/0166-218X(93)90088-6
19. Li, S.-S., Chen, R.-X.: Scheduling with common due date assignment to minimize generalized weighted earliness–tardiness penalties. Optim. Lett. **14**, 1681–1699 (2020). https://doi.org/10.1007/s11590-019-01462-5
20. Shabtay, D., Mosheiov, G., Oron, D.: Single machine scheduling with common assignable due date/due window to minimize total weighted early and late work. Eur. J. Oper. Res. (2022). https://doi.org/10.1016/J.EJOR.2022.02.017
21. Yeung, W.K., Oguz, C., Cheng, T.C.E.: Single-machine scheduling with a common due window. Comput. Oper. Res. **28**, 157–175 (2001). https://doi.org/10.1016/S0305-0548(99)00097-0
22. De, P., Ghosh, J.B., Wells, C.E.: Due-date assignmentand early/tardy scheduling on identical parallel machines. Nav. Res. Logist. **41**, 17–32 (1994). https://doi.org/10.1002/1520-6750(199402)41:1<17::AID-NAV3220410103>3.0.CO;2-X
23. 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
24. Diamond, J.E., Cheng, T.C.E.: Error bound for common due date assignment and job scheduling on parallel machines. IIE Trans. Insti. Ind. Eng. **32**, 445–448 (2000). https://doi.org/10.1023/A:1007644826949
25. Mosheiov, G.: A common due-date assignment problem on parallel identical machines. Comput. Oper. Res. **28**, 719–732 (2001). https://doi.org/10.1016/S0305-0548(99)00127-6
26. Xiao, W.-Q., Li, C.-L.: Approximation algorithms for common due date assignment and job scheduling on parallel machines. IIE Trans. Inst. Ind. Eng. **34**, 466–477 (2002). https://doi.org/10.1080/07408170208928883
27. Mosheiov, G., Yovel, U.: Minimizing weighted earliness-tardiness and due-date cost with unit processing-time jobs. Eur. J. Oper. Res. **172**, 528–544 (2006). https://doi.org/10.1016/j.ejor.2004.10.021
28. Min, L., Cheng, W.: Genetic algorithms for the optimal common due date assignment and the optimal scheduling policy in parallel machine earliness/tardiness scheduling problems. Robot Comput. Integr. Manuf. **22**, 279–287 (2006). https://doi.org/10.1016/j.rcim.2004.12.005
29. Drobouchevitch, I.G., Sidney, J.B.: Minimization of earliness, tardiness and due date penalties on uniform parallel machines with identical jobs. Comput. Oper. Res. **39**, 1919–1926 (2012). https://doi.org/10.1016/j.cor.2011.05.012
30. Kim, J.-G., Kim, J.-S., Lee, D.-H.: Fast and meta-heuristics for common due-date assignment and scheduling on parallel machines. Int. J. Prod. Res. **50**, 6040–6057 (2012). https://doi.org/10.1080/00207543.2011.644591
31. Xiong, X., Wang, D., Cheng, E., Wu, T.C., Yin, C.-C.: Single-machine scheduling and common due date assignment with potential machine disruption. Int. J. Prod. Res. **56**, 1345–1360 (2018). https://doi.org/10.1080/00207543.2017.1346317
32. Ng Daniel, C.T.D., Cheng, T.C.E., Kovalyov, M.Y., Lam, S.S.: Single machine scheduling with a variable common due date and resource-dependent processing times. Comput. Oper. Res. **30**, 1173–1185 (2003). https://doi.org/10.1016/S0305-0548(02)00066-7
33. Keha, A.B., Khowala, K., Fowler, J.W.: Mixed integer programming formulations for single machine scheduling problems. Comput. Ind. Eng. **56**, 357–367 (2009). https://doi.org/10.1016/J.CIE.2008.06.008
34. Biskup, D., Feldmann, M.: Benchmarks for scheduling on a single machine against restrictive and unrestrictive common due dates. Comput. Oper. Res. **28**, 787–801 (2001). https://doi.org/10.1016/S0305-0548(00)00008-3

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