mathematics

MDPI

Article

Two-Agent Slack Due-Date Assignment Scheduling with Resource Allocations and Deteriorating Jobs

Li-Han Zhang †, Dan-Yang Lv † and Ji-Bo Wang *

School of Computer Science, Shenyang Aerospace University, Shenyang 110136, China; zhanglihan@stu.sau.edu.cn (L.-H.Z.); lvdanyang@stu.sau.edu.cn (D.-Y.L.)

* Correspondence: wangjibo@sau.edu.cn

† These authors contributed equally to this work.

Abstract: In enterprise management, there are often multiple agents competing for the same products to reduce production cost. On this basis, this paper investigates a two-agent slack due-date single-machine scheduling problem with deteriorating jobs, where the processing time of a job is extended as a function of position-dependent workload, resource allocation and a common deterioration rate. The goal is to find the optimal sequence and resource allocation that minimizes the maximal value of earliness, tardiness, and decision variables of one agent subject to an upper bound on cost value of the second agent. Through theoretical analysis, a polynomial time algorithm with $O(N^3)$ time is proposed for the problem, where N is the maximum number of jobs between the two agents.

Keywords: scheduling; two-agent; slack due-date; deteriorating job; resource allocation

MSC: 90B35

check for updates

Citation: Zhang, L.-H.; Lv, D.-Y.; Wang, J.-B. Two-Agent Slack Due-Date Assignment Scheduling with Resource Allocations and Deteriorating Jobs. *Mathematics* **2023**, *11*, 2737. https://doi.org/10.3390/math11122737

Academic Editor: Andrea Scozzari

Received: 1 June 2023
Revised: 12 June 2023
Accepted: 15 June 2023
Published: 16 June 2023

1. Introduction

In practical fields such as enterprise management and production processing, there are often multiple agents competing with each other to provide the same products to consumers, with the corresponding multiple comparisons so that can choose the agent in their favor. Such problems are multi-agent scheduling problems (Agnetis et al. [1]; Tuong et al. [2]). Gu et al. [3] proposed an algorithm to minimize the makespan on the basis of given lower bound for the multi-agent scheduling problem of m parallel machine. He et al. [4] elicited pareto-optimal schedule to simultaneously minimize the maximum cost of agent A and makespan of agent B under a two-agent scheduling problem with parallel batch processing. Wang et al. [5] presented a numerical simulation of multi-agent competing for multiple jobs in a cloud manufacturing platform to minimize the total completion time as well as the weighted amount of tardy jobs to provide theoretical support for subsequent investors. Wan et al. [6] also constructed a polynomial time algorithm and a dual FPTAS (fully polynomial time approximation scheme) algorithm to minimize the weighted number of tardy jobs for the single-machine two-agent scheduling problem with unit processing time.

One cause of the tardiness in job production involves another type of scheduling problem, namely, the just-in-time scheduling, which specifies delivery date for a job that incurs excess costs if completed earlier or later. This delivery date is also known as the due date, and familiar due dates include (a) common due date: i.e., the due date of each job is the same constant (Shabtay et al. [7], Falq et al. [8], Wu et al. [9] and Liu and Wang [10]); (b) different due date: the opposite of common due date (Mosheiov et al. [11] and Hidayat et al. [12]); (c) slack due date: the due date of each job, although different, but with a common decision variable (Liu et al. [13] and Liu and Jiang [14]).

For a large due date, it is beneficial for plant production but not for competition, that is to say, the study of the multi-agent scheduling problem with due date is of great importance for practical research. Yin et al. [15] designed a pseudo-polynomial dynamic programming algorithms for a two-agent scheduling problem with minimalist common and slack due date assignment. Wang et al. [16], on the other hand, considered the two-agent single-machine scheduling problem with common, slack, and different due date simultaneously and proposed polynomial-time dynamic programming algorithms to solve them.

To improve the competition rate, the producer can allocate a certain amount of resources to each job to reduce the processing time. The multi-agent scheduling problem with resource allocation has been studied by Wang et al. [17] for a two-agent scheduling with linear resource, for which a FPTAS is proposed for the NP-hard problem. Luo [18] studied the slack due date problem with convex resource, and presented an optimal solution algorithm with time complexity of $O(N^3)$ for the two-agent minmax earliness, tardiness, and common decision variable, where N is the maximum number of jobs between the two agents.

In addition, the increasing processing time due to deteriorating jobs (see Wu et al. [19], Gawiejnowicz [20], Zhang et al. [21], Sun et al. [22]) is unavoidable in practical machining. Individual agents can minimize the common objective function based on rational arrangement of job sequences. Yin et al. [23], Wang et al. [24] and Li et al. [25] explored the two-agent scheduling problem with linear deterioration.

In summary, in this paper, based on Luo [18], the processing time of the job extended as a function of position-dependent workload, resource allocation and a common deterioration rate and minimized maximum value of earliness, tardiness, and decision variable under slack due date of one agent subject to an upper bound on cost value of the second agent is investigated. The goal is to find the minimum cost and the corresponding optimal resource allocation for processing a batch of jobs simultaneously through competition between two agents. The paper is structured as follows: Section 2 describes the problem under study; Section 3 gives the specific algorithm; and Section 4 provides conclusions regarding the problem studied.

The similar literature mentioned above and the specific problem studied in this paper are detailed in Table 1.

Table 1. Literature contents and achievement of this paper.

Research Contents	Algorithm Complexity	Research Conte References
$1\|condd\|r^{\mathrm{A}}q^{\mathrm{A}} + \sum_{k=1}^{N_{\mathrm{A}}} \varpi_k^{\mathrm{A}} U_k^{\mathrm{A}} : \sum_{h=1}^{N_{\mathrm{B}}} C_h^{\mathrm{B}} \leq \widetilde{O}$	$O(N_{\mathrm{A}}^2 N_{\mathrm{B}}/\varepsilon + N_{\mathrm{A}}^2 N_{\mathrm{B}} \log N_{\mathrm{A}})$	Yin et al. [15]
$1\|slkdd\|r^{\mathrm{A}}q^{\mathrm{A}} + \sum_{k=1}^{N_{\mathrm{A}}} \varpi_k^{\mathrm{A}} U_k^{\mathrm{A}} : \sum_{h=1}^{N_{\mathrm{B}}} C_h^{\mathrm{B}} \leq \widetilde{O}$	$O(N_{\mathrm{A}}^3 N_{\mathrm{B}}/\varepsilon + N_{\mathrm{A}}^3 N_{\mathrm{B}} \log N_{\mathrm{A}})$	Yin et al. [15]
$1\|condd\|r^{\mathrm{A}}d^{\mathrm{A}} + \sum_{k=1}^{\mathrm{A}} (a^{\mathrm{A}} E_k^{\mathrm{A}} + \varpi^{\mathrm{A}} U_k^{\mathrm{A}}) :$ $r^{\mathrm{B}}d^{\mathrm{B}} + \sum_{h=1}^{\mathrm{B}} (a^{\mathrm{B}} E_h^{\mathrm{B}} + \varpi^{\mathrm{B}} U_h^{\mathrm{B}}) \leq \widetilde{O}$	$O(N_{\mathrm{A}}^2 N_{\mathrm{B}}^2 \widetilde{O})$	Wang et al. [16]
$1\|slkdd\|r^{\mathrm{A}}q^{\mathrm{A}} + \sum_{k=1}^{\mathrm{A}} (a^{\mathrm{A}} E_k^{\mathrm{A}} + \varpi^{\mathrm{A}} U_k^{\mathrm{A}}) :$ $r^{\mathrm{B}}q^{\mathrm{B}} + \sum_{h=1}^{\mathrm{B}} (a^{\mathrm{B}} E_h^{\mathrm{B}} + \varpi^{\mathrm{B}} U_h^{\mathrm{B}}) \leq \widetilde{O}$	$O(N_{\mathrm{A}}^2 N_{\mathrm{B}}^2 \widetilde{O})$	Wang et al. [16]
$1\|difdd\| \sum_{k=1}^{\mathrm{A}} (r^{\mathrm{A}} d_k^{\mathrm{A}} + a^{\mathrm{A}} E_k^{\mathrm{A}} + \varpi^{\mathrm{A}} U_k^{\mathrm{A}}) :$ $\sum_{h=1}^{\mathrm{B}} (r^{\mathrm{B}} d_h^{\mathrm{B}} + a^{\mathrm{B}} E_h^{\mathrm{B}} + \varpi^{\mathrm{B}} U_h^{\mathrm{B}}) \leq \widetilde{O}$	$O(N_{\mathrm{A}}^2 N_{\mathrm{B}}^2 \widetilde{O})$	Wang et al. [16]
$1\|t_k^L = p_k^L(c+vs), d_k^{\mathrm{A}} = d^{\mathrm{A}}\| \sum_{k=1}^{N_{\mathrm{A}}} a_k^{\mathrm{A}} E_k^{\mathrm{A}} : F_{\max}^{\mathrm{B}}(E^{\mathrm{B}}) \leq \widetilde{O}$	$O(N_{\mathrm{A}} \log N_{\mathrm{A}} + N_{\mathrm{B}} \log N_{\mathrm{B}})$	Yin et al. [23]
$1\|t_k^L = \Upsilon_k(c+vs)\| \sum_{k=1}^{\mathrm{A}} \omega_k C_k^{\mathrm{A}} + zL_{\max}^{\mathrm{B}}$	NP-hard	Wang et al. [24]
$1\|slkdd, conv, \sum_{k}^{N_{\mathrm{A}}} w_k \leq W, \sum_{h=1}^{N_{\mathrm{B}}} u_h \leq U\|$ $\max_{1\leq k\leq N_{\mathrm{A}}} \{\max\{a^{\mathrm{A}} E_k^{\mathrm{A}} + r^{\mathrm{A}}q^{\mathrm{A}}, b^{\mathrm{A}} T_k^{\mathrm{A}} + r^{\mathrm{A}}q^{\mathrm{A}}\}\} :$ $\max_{1\leq h\leq N_{\mathrm{B}}} \{\max\{a^{\mathrm{B}} E_h^{\mathrm{B}} + r^{\mathrm{B}}q^{\mathrm{B}}, b^{\mathrm{B}} T_h^{\mathrm{B}} + r^{\mathrm{B}}q^{\mathrm{B}}\}\} \leq \widetilde{O}$	$O(N^3)$ $(N = \max\{N_{\mathrm{A}}, N_{\mathrm{B}}\})$	Luo [18]
$1\|slkdd, de, \sum_{k}^{N_{\mathrm{A}}} w_k \leq W, \sum_{h=1}^{N_{\mathrm{B}}} u_h \leq U\|$ $\max_{1\leq k\leq N_{\mathrm{A}}} \{\max\{a^{\mathrm{A}} E_k^{\mathrm{A}} + r^{\mathrm{A}}q^{\mathrm{A}}, b^{\mathrm{A}} T_k^{\mathrm{A}} + r^{\mathrm{A}}q^{\mathrm{A}}\}\} :$ $\max_{1\leq h\leq N_{\mathrm{B}}} \{\max\{a^{\mathrm{B}} E_h^{\mathrm{B}} + r^{\mathrm{B}}q^{\mathrm{B}}, b^{\mathrm{B}} T_h^{\mathrm{B}} + r^{\mathrm{B}}q^{\mathrm{B}}\}\} \leq \widetilde{O}$	$O(N^3)$ $(N = \max\{N_{\mathrm{A}}, N_{\mathrm{B}}\})$	This paper

2. Notation Description

The symbols involved in this research are detailed in Table 2:

Table 2. Notation table.

Notation	Significance
$slkdd$	slack due date
de	deteriorating jobs
A (resp. B)	agent A (agent B)
L	set of agents
N_{A} (resp. N_{B})	number of jobs of A (number of jobs of B)
$\mathbb{J}^L$	set of jobs of two agents
J_k^{A} (resp. J_h^{B})	job J_k of agent A, $k = 1, \ldots, N_{\mathrm{A}}$ (job J_h of agent B, $h = 1, \ldots, N_{\mathrm{A}}$)
$J_{k,x}$	J_k^{A} at position x ($k = 1, \ldots, N_{\mathrm{A}}$, $x = 1, \ldots, N_{\mathrm{A}}$)
$J_{h,y}$	J_h^{B} at position y ($h = 1, \ldots, N_{\mathrm{B}}$, $y = 1, \ldots, N_{\mathrm{B}}$)
$t_{k,x}$ ($t_{h,y}$)	processing time of $J_{k,x}$ (processing time of $J_{h,y}$)
$\rho_{k,x}$ ($\eta_{h,y}$)	workload of $J_{k,x}$ (workload of $J_{h,y}$)
w_k (u_h)	resource of J_k^{A} (resource of J_h^{B})
$\sum_{k}^{N_{\mathrm{A}}} w_k \leq W$	upper bound of resource allocation of A
$\sum_{h=1}^{N_{\mathrm{B}}} u_h \leq U$	upper bound of resource allocation of B
Υ	deterioration rate
s_k (s_h)	starting time of J_k^{A} (starting time of J_h^{B})
q^L	common decision variable
d_k^L	due date of J_k^L
C_k^L	completion time J_k^L
E_k^L	amount of earliness
T_k^L	amount of tardiness
σ	job scheduling

3. Problem Description

Consider two agents A and B, each with N_L independent jobs, i.e., $\mathbb{J}^L = \{J_1^L, J_2^L, \ldots, J_{N_L}^L\}$, where $L \in \{\mathrm{A}, \mathrm{B}\}$. These jobs need to be processed on a machine without interruption by competition and one job can be processed at a time. For agent A (B), the processing time $t_{k,x}$ ($t_{h,y}$) of job J_k^{A} (J_h^{B}) at position x (y) can be expressed as

$$t_{k,x}(w_k, s_k) = \left(\frac{\rho_{k,x}}{w_k}\right)^{\kappa} + \Upsilon s_k \tag{1}$$

$$t_{h,y}(u_h, s_h) = \left(\frac{\eta_{h,y}}{u_h}\right)^{\kappa} + \Upsilon s_h \tag{2}$$

in which $\rho_{k,x}$ (reps. $\eta_{h,y}$) is the workload of J_k^{A} (resp. J_h^{B}) at position x (y); w_k (resp. r_h) is the amount of resources allocated to J_k^{A} (resp. J_h^{B}); Υ and s_k (resp. s_h) are the deterioration rate and starting time of J_k^{A} (resp. J_h^{B}), respectively.

As in the slack due date assignment mentioned in the introduction, each job J_k^L ($k = 1, \ldots, N_L$) enjoys a common decision variable, which can be written as q^L (≥ 0). The corresponding due date can be given as

$$d_k^L = t_k^L + q^L, k = 1, \ldots, N_L, L \in \{\mathrm{A}, \mathrm{B}\} \tag{3}$$

where t_k^L is given by (1) and (2) corresponding to the two agents. For the job completed earlier/later than d_k^L, the amount of earliness/tardiness can be indicated as $E_k^L = \max\{0, t_k^L + q^L - C_k^L\} / T_k^L = \max\{0, C_k^L - t_k^L - q^L\}$.

As in Luo [18], this paper minimizes the maximum value of earliness, tardiness, and decision variable q^L; i.e., the objective function is

$$F^L = \max_{1 \le k \le N_L} \left\{ \max\left\{ a^L E_k^L + r^L q^L, b^L T_k^L + r^L q^L \right\} \right\}, L \in \{\mathrm{A}, \mathrm{B}\} \tag{4}$$

and a^L, b^L, and r^L are the costs of earliness, tardiness, and the decision variable of J_k^L.

Under the total amount of resources constraints of agents A and B (i.e., $\sum_k^{N_\mathrm{A}} w_k \le W$, $\sum_{h=1}^{N_\mathrm{B}} u_h \le U$), the goal is to find the optimal sequence and the optimal quantity of resource allocations $\mathbf{w} = \{w_1, \ldots, w_{N_\mathrm{A}}\}$ and $\mathbf{u} = \{u_1, \ldots, u_{N_\mathrm{B}}\}$ such that the cost of agent A is minimized subject to the cost of B satisfies $F^B \le \widetilde{O}$. This problem can be stated in a three-field notation as

$$1 \left| slkdd, de, \sum_k^{N_\mathrm{A}} w_k \le W, \sum_{h=1}^{N_\mathrm{B}} u_h \le U \right| F^\mathrm{A} : F^\mathrm{B} \le \widetilde{O},$$

where *slkdd* is slack due date and *de* is the job processing time expressions (1) and (2).

4. Problem Solving

Let $[k]$ be the job at the k-th position in the sequence. According to the Equations (1) and (2), the completion time of $J_{[k]}$ for agent A (B) can be organized as follows,

$$C_{[k]}^\mathrm{A} = \sum_{l=1}^{k} \left(\frac{\rho_{[l],l}}{w_{[l]}} \right)^\kappa (1+\Upsilon)^{k-l}$$

$$C_{[h]}^\mathrm{B} = \sum_{l=1}^{h} \left(\frac{\eta_{[l],l}}{u_{[l]}} \right)^\kappa (1+\Upsilon)^{h-l}$$

and the corresponding processing time can be collated as

$$t_{[k]}^\mathrm{A} = \left(\frac{\rho_{[k],k}}{w_{[k]}} \right)^\kappa + \Upsilon \left(\sum_{l=1}^{k-1} \left(\frac{\rho_{[l],l}}{w_{[l]}} \right)^\kappa (1+\Upsilon)^{k-l-1} \right)$$

$$t_{[h]}^\mathrm{B} = \left(\frac{\rho_{[h],h}}{w_{[h]}} \right)^\kappa + \Upsilon \left(\sum_{l=1}^{h-1} \left(\frac{\eta_{[l],l}}{u_{[l]}} \right)^\kappa (1+\Upsilon)^{h-l-1} \right)$$

Lemma 1. *For the given sequence σ_A of agent* A*, the sum of processing time can be written as*

$$\sum_{k=1}^{N_\mathrm{A}} t_{[k]}^\mathrm{A} = \sum_{k=1}^{N_\mathrm{A}} \chi_k \left(\frac{\rho_{[k],k}}{w_{[k]}} \right)^\kappa$$

where χ_k is the position-dependent coefficient, and

$$\chi_k = 1 + \Upsilon \sum_{m=1}^{N_\mathrm{A}-k} (1+\Upsilon)^{N_\mathrm{A}-k-m} \tag{5}$$

in which $k = 1, \ldots, N_\mathrm{A}$. The expression for the coefficient of agent B *is similar, and can be expressed in term of ς while replacing N_A with N_B.*

Proof of Lemma 1.

$$\begin{aligned}
\sum_{k=1}^{N_A} t_{[k]}^{A} &= \sum_{k=1}^{N_A}\left[\left(\frac{\rho_{[k],k}}{w_{[k]}}\right)^{\kappa} + \Upsilon\left(\sum_{l=1}^{k-1}\left(\frac{\rho_{[l],l}}{w_{[l]}}\right)^{\kappa}(1+\Upsilon)^{k-l-1}\right)\right] \\
&= \left(\frac{\rho_{[1],1}}{w_{[1]}}\right)^{\kappa} + \left(\frac{\rho_{[2],2}}{w_{[2]}}\right)^{\kappa} + \Upsilon\left(\frac{\rho_{[1],1}}{w_{[1]}}\right)^{\kappa} + \left(\frac{\rho_{[3],3}}{w_{[3]}}\right)^{\kappa} + \Upsilon\left(\frac{\rho_{[2],2}}{w_{[2]}}\right)^{\kappa} \\
&\quad + \Upsilon(1+\Upsilon)\left(\frac{\rho_{[1],1}}{w_{[1]}}\right)^{\kappa} + \ldots + \left(\frac{\rho_{[N_A],N_A}}{w_{[N_A]}}\right)^{\kappa} + \Upsilon\left(\frac{\rho_{[N_A-1],N_A-1}}{w_{[N_A-1]}}\right)^{\kappa} \\
&\quad + \Upsilon(1+\Upsilon)\left(\frac{\rho_{[N_A-2],N_A-2}}{w_{[N_A-2]}}\right)^{\kappa} + \ldots \\
&\quad + \Upsilon(1+\Upsilon)^{N_A-2}\left(\frac{\rho_{[1],1}}{w_{[1]}}\right)^{\kappa} \\
&= \sum_{k=1}^{N_A} \chi_k \left(\frac{\rho_{[k],k}}{w_{[k]}}\right)^{\kappa}
\end{aligned}$$

□

To solve the problem $1\left|slkdd, de, \sum_{k}^{N_A} w_k \le W, \sum_{h=1}^{N_B} u_h \le U\right| F^A : F^B \le \widetilde{O}$, the following properties can be given.

Lemma 2 (Mor and Mosheiov [26])**.** *The jobs in agents* A *and* B *are processed sequentially according to the block structure; i.e., there are two possible feasible sequences:* $\sigma_1 = \tau\{\sigma_A, \sigma_B\}$*, or* $\sigma_2 = \tau\{\sigma_B, \sigma_A\}$*.*

Lemma 3 (Mor and Mosheiov [26])**.** *For a given sequence* σ *and resource allocation, the optimal decision variable* q^L *for agent* L *and the corresponding objective function value are, respectively, determined by the following equations:*

$$q^L(\sigma) = \begin{cases} 0, & r^L > b^L \\ \frac{a^L s_1^L + b^L s_{N_L}^L}{a^L + b^L}, & r^L \le b^L \end{cases} \tag{6}$$

$$F^L(\sigma) = \begin{cases} b^L s_{N_L}^L, & r^L > b^L \\ \frac{b^L(a^L+r^L)s_{N_L}^L - a^L(b^L-r^L)s_1^L}{a^L+b^L}, & r^L \le b^L \end{cases} \tag{7}$$

in which $L \in \{A, B\}$*, and* s_1^L *and* $s_{N_L}^L$ *are the starting time of* J_1^L *and* $J_{N_L}^L$*, and related to the job processing sequence.*

According to (7), when selecting the job processing sequence to minimize $s_{N_L}^L$ ($r^L > b^L$), or to minimize $s_{N_L}^L$ while maximizing s_1^L ($r^L \le b^L$), $F^L(\sigma)$ can be minimized. Because the position of the job is not determined, the specific values of s_1^L and $s_{N_L}^L$ cannot be obtained. For this, the $0-1$ variables need to be introduced, i.e., $\{\Theta_{k,x}|k, x = 1, \ldots, N_A\}$ and $\{\Omega_{h,y}|h, y = 1, \ldots, N_B\}$, in which $\Theta_{k,x} = 1$; that is, the job J_k^A is assigned at the x-th position. Otherwise, $\Theta_{k,x} = 0$; $\Omega_{h,y} = 1$, the job J_h^B is assigned at the y-th position, otherwise $\Omega_{h,y} = 0$. For the introduction of $0-1$ variables, the job sequence is also determined, then the expressions for s_1^L and $s_{N_L}^L$ under the sequence σ_1 can be written according to Lemma 2 as

$$s_1^A(\sigma_1) = 0, s_{N_A}^A(\sigma_1) = \sum_{k=1}^{N_A}\sum_{x=1}^{N_A-1} t_{k,x}(w_k, s_k)\Theta_{k,x} = \sum_{k=1}^{N_A}\sum_{x=1}^{N_A-1} \chi_x \left(\frac{\rho_{k,x}}{w_k}\right)^{\kappa}\Theta_{k,x} \tag{8}$$

It can be inferred from (7) that

$$F^{\mathrm{A}}(\sigma_1) = \psi^{\mathrm{A}} \sum_{k=1}^{N_{\mathrm{A}}} \sum_{x=1}^{N_{\mathrm{A}}-1} \chi_x \left(\frac{\rho_{k,x}}{w_k}\right)^{\kappa} \Theta_{k,x} \tag{9}$$

where ψ^{A} can be obtained by bringing (9) to the two conditions in (7) separately as follows:

$$\psi^{\mathrm{A}} = \min\left\{ b^{\mathrm{A}}, \frac{b^{\mathrm{A}}(a^{\mathrm{A}} + r^{\mathrm{A}})}{a^{\mathrm{A}} + b^{\mathrm{A}}} \right\}$$

$$s_1^{\mathrm{B}}(\sigma_1) = C_{N_{\mathrm{A}}}^{\mathrm{A}}(\sigma_1) = \sum_{k=1}^{N_{\mathrm{A}}} \sum_{x=1}^{N_{\mathrm{A}}} \chi_x \left(\frac{\rho_{k,x}}{w_k}\right)^{\kappa} \Theta_{k,x} \tag{10}$$

$$s_{N_{\mathrm{B}}}^{\mathrm{B}}(\sigma_1) = \sum_{k=1}^{N_{\mathrm{A}}} \sum_{x=1}^{N_{\mathrm{A}}} \chi_x \left(\frac{\rho_{k,x}}{w_k}\right)^{\kappa} \Theta_{k,x} + \sum_{h=1}^{N_{\mathrm{B}}} \sum_{y=1}^{N_{\mathrm{B}}-1} \varsigma_y \left(\frac{\eta_{h,y}}{u_h}\right)^{\kappa} \Omega_{h,y} \tag{11}$$

$$F^{\mathrm{B}}(\sigma_1) = \begin{cases} b^{\mathrm{B}} \left[\begin{array}{c} \sum_{k=1}^{N_{\mathrm{A}}} \sum_{x=1}^{N_{\mathrm{A}}} \chi_x \left(\frac{\rho_{k,x}}{w_k}\right)^{\kappa} \Theta_{k,x} \\ + \sum_{h=1}^{N_{\mathrm{B}}} \sum_{y=1}^{N_{\mathrm{B}}-1} \varsigma_y \left(\frac{\eta_{h,y}}{u_h}\right)^{\kappa} \Omega_{h,y} \end{array} \right], & r^{\mathrm{B}} > b^{\mathrm{B}} \\ \begin{array}{c} r^{\mathrm{B}} \sum_{k=1}^{N_{\mathrm{A}}} \sum_{x=1}^{N_{\mathrm{A}}} \chi_x \left(\frac{\rho_{k,x}}{w_k}\right)^{\kappa} \Theta_{k,x} \\ + \frac{b^{\mathrm{B}}(a^{\mathrm{B}}+r^{\mathrm{B}})}{a^{\mathrm{B}}+b^{\mathrm{B}}} \sum_{h=1}^{N_{\mathrm{B}}} \sum_{y=1}^{N_{\mathrm{B}}-1} \varsigma_y \left(\frac{\eta_{h,y}}{u_h}\right)^{\kappa} \Omega_{h,y} \end{array}, & r^{\mathrm{B}} \le b^{\mathrm{B}} \end{cases} \tag{12}$$

Under the sequence σ_2, s_1^L and $s_{N_{\mathrm{L}}}^L$ can be similarly represented as

$$s_1^{\mathrm{B}}(\sigma_2) = 0, s_{N_{\mathrm{B}}}^{\mathrm{B}}(\sigma_2) = \sum_{h=1}^{N_{\mathrm{B}}} \sum_{y=1}^{N_{\mathrm{B}}-1} t_{h,y}(u_h, s_h)\Omega_{h,y} = \sum_{h=1}^{N_{\mathrm{B}}} \sum_{y=1}^{N_{\mathrm{B}}-1} \varsigma_y \left(\frac{\eta_{h,y}}{u_h}\right)^{\kappa} \Omega_{h,y} \tag{13}$$

It also can be known from (7) that

$$F^{\mathrm{B}}(\sigma_2) = \psi^{\mathrm{B}} \sum_{h=1}^{N_{\mathrm{B}}} \sum_{y=1}^{N_{\mathrm{B}}-1} \varsigma_y \left(\frac{\eta_{h,y}}{u_h}\right)^{\kappa} \Omega_{h,y} \tag{14}$$

where ψ^{B} can be obtained in the same way as ψ^{A}:

$$\psi^{\mathrm{B}} = \min\left\{ b^{\mathrm{B}}, \frac{b^{\mathrm{B}}(a^{\mathrm{B}} + r^{\mathrm{B}})}{a^{\mathrm{B}} + b^{\mathrm{B}}} \right\}$$

$$s_1^{\mathrm{A}}(\sigma_2) = \sum_{h=1}^{N_{\mathrm{B}}} \sum_{y=1}^{N_{\mathrm{B}}} \varsigma_y \left(\frac{\eta_{h,y}}{u_h}\right)^{\kappa} \Omega_{h,y} \tag{15}$$

$$s_{N_{\mathrm{A}}}^{\mathrm{A}}(\sigma_2) = \sum_{h=1}^{N_{\mathrm{B}}} \sum_{y=1}^{N_{\mathrm{B}}} \varsigma_y \left(\frac{\eta_{h,y}}{u_h}\right)^{\kappa} \Omega_{h,y} + \sum_{k=1}^{N_{\mathrm{A}}} \sum_{x=1}^{N_{\mathrm{A}}-1} \chi_x \left(\frac{\rho_{k,x}}{w_k}\right)^{\kappa} \Theta_{k,x} \tag{16}$$

$$F^{\mathrm{A}}(\sigma_2) = \begin{cases} b^{\mathrm{A}} \left[\begin{array}{c} \sum_{h=1}^{N_{\mathrm{B}}} \sum_{y=1}^{N_{\mathrm{B}}} \varsigma_y \left(\frac{\eta_{h,y}}{u_h}\right)^{\kappa} \Omega_{h,y} \\ + \sum_{k=1}^{N_{\mathrm{A}}} \sum_{x=1}^{N_{\mathrm{A}}-1} \chi_x \left(\frac{\rho_{k,x}}{w_k}\right)^{\kappa} \Theta_{k,x} \end{array} \right], & r^{\mathrm{A}} > b^{\mathrm{A}} \\ \begin{array}{c} r^{\mathrm{A}} \sum_{h=1}^{N_{\mathrm{B}}} \sum_{y=1}^{N_{\mathrm{B}}} \varsigma_y \left(\frac{\eta_{h,y}}{u_h}\right)^{\kappa} \Omega_{h,y} \\ + \frac{b^{\mathrm{A}}(a^{\mathrm{A}}+r^{\mathrm{A}})}{a^{\mathrm{A}}+b^{\mathrm{A}}} \sum_{k=1}^{N_{\mathrm{A}}} \sum_{x=1}^{N_{\mathrm{A}}-1} \chi_x \left(\frac{\rho_{k,x}}{w_k}\right)^{\kappa} \Theta_{k,x} \end{array}, & r^{\mathrm{A}} \le b^{\mathrm{A}} \end{cases} \tag{17}$$

It is clear that φ^{A} is a constant and does not affect the ordering of the jobs; therefore, the minimization of (9) obtained above can be translated into the following optimization problem:

$$\begin{aligned} \min \ F_1^{\mathrm{A}} &= \sum_{k=1}^{N_{\mathrm{A}}}\sum_{x=1}^{N_{\mathrm{A}}} \chi_x \left(\frac{\rho_{k,x}}{w_k}\right)^{\kappa} \Theta_{k,x} \\ s.t. \quad & \sum_{k=1}^{N_{\mathrm{A}}} \Theta_{k,x} = 1, x = 1,\ldots,N_{\mathrm{A}}-1 \\ & \sum_{x=1}^{N_{\mathrm{A}}} \Theta_{k,x} = 1, k = 1,\ldots,N_{\mathrm{A}} \\ & \sum_{k}^{N_{\mathrm{A}}} w_k \leq \ W \\ & w_k > 0, k = 1,\ldots,N_{\mathrm{A}} \end{aligned} \tag{18}$$

Obviously, the premise of minimizing $F^{\mathrm{A}}(\sigma_1)$ is to find the optimal resource allocation $\mathbf{w} = \{w_1, \ldots, w_{N_{\mathrm{A}}}\}$, for which the solution can be converted to the following nonlinear programming problem:

$$\begin{aligned} \min \ F_1^{\mathrm{A}} &= \sum_{k=1}^{N_{\mathrm{A}}}\sum_{x=1}^{N_{\mathrm{A}}} \chi_x \left(\frac{\rho_{k,x}}{w_k}\right)^{\kappa} \Theta_{k,x} \\ s.t. \quad & \sum_{k}^{N_{\mathrm{A}}} w_k \leq W \\ & w_k > 0, k = 1,\ldots,N_{\mathrm{A}} \end{aligned} \tag{19}$$

and the following property can be given according to (19).

Lemma 4. *For a given sequence of jobs σ_1, the optimal resource vectors $\boldsymbol{w}^* = (w^1, \ldots, w^*_{N_{\mathrm{A}}})$, $\boldsymbol{u}^* = (u_1^*, \ldots, u^*_{N_{\mathrm{B}}})$, and the corresponding objective function value $F^L(\sigma_1)$ can be written specifically as*

$$w_k^* = \frac{\left(\sum_{x=1}^{N_{\mathrm{A}}} \chi_x \rho_{k,x}^{\kappa} \Theta_{k,x}\right)^{\frac{1}{\kappa+1}}}{\sum_{k=1}^{N_{\mathrm{A}}}\left(\sum_{x=1}^{N_{\mathrm{A}}} \chi_x \rho_{k,x}^{\kappa} \Theta_{k,x}\right)^{\frac{1}{\kappa+1}}} \times W \tag{20}$$

$$u_h^* = \frac{\left(\sum_{y=1}^{N_{\mathrm{B}}} \varsigma_y \eta_{h,y}^{\kappa} \Omega_{h,y}\right)^{\frac{1}{\kappa+1}}}{\sum_{h=1}^{N_{\mathrm{B}}}\left(\sum_{y=1}^{N_{\mathrm{B}}} \varsigma_y \eta_{h,y}^{\kappa} \Omega_{h,y}\right)^{\frac{1}{\kappa+1}}} \times U \tag{21}$$

$$F^{\mathrm{A}}(\sigma_1) = \frac{\psi^{\mathrm{A}}}{W^{\kappa}}\left[\sum_{k=1}^{N_{\mathrm{A}}}\left(\sum_{x=1}^{N_{\mathrm{A}}} \chi_x \rho_{k,x}^{\kappa} \Theta_{k,x}\right)^{\frac{1}{\kappa+1}}\right]^{\kappa+1} \tag{22}$$

and

$$F^{\mathrm{B}}(\sigma_1) = \begin{cases} b^{\mathrm{B}}\left[\begin{array}{c} \frac{1}{W^{\kappa}}\left(\sum_{k=1}^{N_{\mathrm{A}}}\left(\sum_{x=1}^{N_{\mathrm{A}}} \chi_x \rho_{k,x}^{\kappa} \Theta_{k,x}\right)^{\frac{1}{\kappa+1}}\right)^{\kappa+1} \\ +\frac{1}{U^{\kappa}}\left(\sum_{h=1}^{N_{\mathrm{B}}}\left(\sum_{y=1}^{N_{\mathrm{B}}} \varsigma_y \eta_{h,y}^{\kappa} \Omega_{h,y}\right)^{\frac{1}{\kappa+1}}\right)^{\kappa+1} \end{array}\right], & r^{\mathrm{B}} > b^{\mathrm{B}} \\ \frac{r^{\mathrm{B}}}{W^{\kappa}}\left(\sum_{k=1}^{N_{\mathrm{A}}}\left(\sum_{x=1}^{N_{\mathrm{A}}} \chi_x \rho_{k,x}^{\kappa} \Theta_{k,x}\right)^{\frac{1}{\kappa+1}}\right)^{\kappa+1} & \\ +\frac{b^{\mathrm{B}}(a^{\mathrm{B}}+r^{\mathrm{B}})}{(a^{\mathrm{B}}+b^{\mathrm{B}})U^{\kappa}}\left(\sum_{h=1}^{N_{\mathrm{B}}}\left(\sum_{y=1}^{N_{\mathrm{B}}} \varsigma_y \eta_{h,y}^{\kappa} \Omega_{h,y}\right)^{\frac{1}{\kappa+1}}\right)^{\kappa+1}, & r^{\mathrm{B}} \leq b^{\mathrm{B}} \end{cases} \tag{23}$$

Proof of Lemma 4. The proof of agent B is the same.

The Lagrange multiplier method is used to solve (19), and the specific function can be written as

$$L(\mathbf{w}, \mu) = \sum_{k=1}^{N_{\mathrm{A}}} \sum_{x=1}^{N_{\mathrm{A}}} \chi_x \left(\frac{\rho_{k,x}}{w_k} \right)^{\kappa} \Theta_{k,x} + \mu \left(\sum_{k}^{N_{\mathrm{A}}} w_k - W \right)$$

Take partial derivatives of w_k and μ yield

$$\frac{\partial L(\mathbf{w}, \mu)}{\partial w_k} = -\kappa \sum_{x=1}^{N_{\mathrm{A}}} \frac{\rho_{k,x}^{\kappa}}{w_k^{\kappa+1}} \Theta_{k,x} + \mu = 0 \tag{24}$$

$$\frac{\partial L(\mathbf{w}, \mu)}{\partial \mu} = \sum_{k}^{N_{\mathrm{A}}} w_k - W = 0 \tag{25}$$

Combing (24) and (25), the collation gives

$$w_k^* = \frac{\left(\sum_{x=1}^{N_{\mathrm{A}}} \chi_x \rho_{k,x}^{\kappa} \Theta_{k,x} \right)^{\frac{1}{\kappa+1}}}{\sum_{k=1}^{N_{\mathrm{A}}} \left(\sum_{x=1}^{N_{\mathrm{A}}} \chi_x \rho_{k,x}^{\kappa} \Theta_{k,x} \right)^{\frac{1}{\kappa+1}}} \times W$$

Bringing (20) to the objective function F_1^{A} in (18) yields (22). □

According to (22) obtained in this lemma, (18) can be turned into solving the following assignment problem, and agent B in the same way as follows

$$\begin{aligned} AP(\sigma_{\mathrm{A}}): \ \min \quad & \sum_{k=1}^{N_{\mathrm{A}}} \sum_{x=1}^{N_{\mathrm{A}}} \chi_x^{\frac{1}{\kappa+1}} \rho_{k,x}^{\frac{\kappa}{\kappa+1}} \Theta_{k,x} \\ s.t. \quad & \sum_{k=1}^{N_{\mathrm{A}}} \Theta_{k,x} = 1, x = 1, \ldots, N_{\mathrm{A}} - 1 \\ & \sum_{x=1}^{N_{\mathrm{A}}} \Theta_{k,x} \le 1, k = 1, \ldots, N_{\mathrm{A}} \\ & \Theta_{k,x} = 0/1, k, x = 1, \ldots, N_{\mathrm{A}} \end{aligned} \qquad \begin{aligned} AP(\sigma_{\mathrm{B}}): \ \min \quad & \sum_{h=1}^{N_{\mathrm{B}}} \sum_{y=1}^{N_{\mathrm{B}}} \varsigma_y^{\frac{1}{\kappa+1}} \eta_{h,y}^{\frac{\kappa}{\kappa+1}} \Omega_{h,y} \\ s.t. \quad & \sum_{h=1}^{N_{\mathrm{B}}} \Omega_{h,y} = 1, y = 1, \ldots, N_{\mathrm{B}} - 1 \\ & \sum_{y=1}^{N_{\mathrm{B}}} \Omega_{h,y} \le 1, h = 1, \ldots, N_{\mathrm{B}} \\ & \Omega_{h,y} = 0/1, h, y = 1, \ldots, N_{\mathrm{B}} \end{aligned}$$

Then, Algorithm 1 can be given for the case where the deterioration rates are all the same.

Note: The optimal solution of the problem can be determined only by calculating the objective function values F^B under two sequences, $\sigma_1 = \tau\{\sigma_{\mathrm{A}}, \sigma_{\mathrm{B}}\}$ and $\sigma_2 = \tau\{\sigma_{\mathrm{B}}, \sigma_{\mathrm{A}}\}$, separately according to the steps of Algorithm 1. That is, if the objective function value $F^{\mathrm{B}} < \widetilde{O}$ is calculated for either of the two sequences, there is an optimal sequence; otherwise, there is no feasible solution.

Theorem 1. *For the problem*

$$1 \left| slkdd, de, \sum_{k}^{N_{\mathrm{A}}} w_k \le W, \sum_{h=1}^{N_{\mathrm{B}}} u_h \le U \right| F^{\mathrm{A}} : F^{\mathrm{B}} \le \widetilde{O},$$

it can be solved in $O(N^3)$ time by Algorithm 1, where $N = \max\{N_{\mathrm{A}}, N_{\mathrm{B}}\}$.

Proof of Theorem 1. Step 1, the solution of $AP(\sigma_{\mathrm{A}})$ and $AP(\sigma_{\mathrm{B}})$ takes $O(N_{\mathrm{A}}^3)$ and $O(N_{\mathrm{B}}^3)$; Step 2 takes $O(N_{\mathrm{A}})$ or $O(N_{\mathrm{B}})$ time; Steps 3–6 are constant time, then the total time complexity does not exceed $O(N^3)$ ($N = \max\{N_{\mathrm{A}}, N_{\mathrm{B}}\}$). □

Algorithm 1: The algorithm of $1\left|slkdd, de, \sum_{k}^{N_{\mathrm{A}}} w_k \leq W, \sum_{h=1}^{N_{\mathrm{B}}} u_h \leq U\right|F^{\mathrm{A}} : F^{\mathrm{B}} \leq \widetilde{O}$

Input: N_L, a^L, b^L, r^L, κ, Υ, W, U, $\widetilde{O}$, where $L \in \{\mathrm{A}, \mathrm{B}\}$, and workload $(\rho_{k,x})_{N_{\mathrm{A}} \times \mathrm{A}}$ and $(\eta_{h,y})_{N_{\mathrm{B}} \times \mathrm{B}}$.
Output: The optimal sequence σ_1 or σ_2, or no optimal sequence, resource allocation $\mathbf{w}$ and $\mathbf{u}$, and common decision variable q^L and corresponding due date d_k^L.
Step 1. According to the assignment problem, $AP(\sigma_{\mathrm{A}})$ and $AP(\sigma_{\mathrm{B}})$, and bring them into (22) and (23) to calculate the value of objective function. If $F_{\mathrm{B}}(\sigma_1) \leq \widetilde{O}$, then output the optimal sequence $\sigma_1 = \tau\{\sigma_{\mathrm{A}}, \sigma_{\mathrm{B}}\}$; otherwise turn to Step 4.
Step 2. Calculate the optimal amount of resource allocation according to (20) and (21).
Step 3. Calculate the processing time $t_{k,x}$ ($t_{h,y}$) of the job by using (1) and (2), $q^{\mathrm{A}}(\sigma_1)$ ($q^{\mathrm{B}}(\sigma_1)$) according to Lemma 3 and $d_k^{\mathrm{A}}(\sigma_1)$ ($d_h^{\mathrm{B}}(\sigma_1)$) according to (3).
Step 4. Consider the feasibility of the sequence $\sigma_2 = \tau\{\sigma_{\mathrm{B}}, \sigma_{\mathrm{A}}\}$ at this point. Calculate $F_{\mathrm{B}}(\sigma_2)$ after making appropriate adjustments from (22), if $F_{\mathrm{B}}(\sigma_2) \leq \widetilde{O}$, turn to S_5; otherwise turn to Step 6.
Step 5. Calculate $F_{\mathrm{A}}(\sigma_2)$ after adjusting according to (23), and output the optimal sequence $\sigma_2 = \tau\{\sigma_{\mathrm{B}}, \sigma_{\mathrm{A}}\}$. The following steps are the same as Step 2 and Step 3.
Step 6. Output conclusion: the problem has no optimal solution. End of the algorithm.

5. Examples

Example 1. *Consider* $N_{\mathrm{A}} = 7$, $N_{\mathrm{B}} = 6$ *as an example,* $\Upsilon = \kappa = 2$, $W = 100$, $U = 80$, $\widetilde{O} = 2000$, $(a^{\mathrm{A}}, a^{\mathrm{B}}) = (15, 7)$, $(b^{\mathrm{A}}, b^{\mathrm{B}}) = (10, 8)$, $(r^{\mathrm{A}}, r^{\mathrm{B}}) = (6, 8)$, *the workload of agent* A *can be randomly generated by MATLAB with a* 7×7 *matrix, as detailed in the following table (Table 3), and the workload of* B *can be obtained by removing the last row and column from the table.*

Table 3. The workload of agent A.

k/x (h/y)	1	2	3	4	5	6	7
1	82	55	81	4	66	83	77
2	91	96	15	85	18	70	80
3	13	97	43	94	71	32	19
4	92	16	92	68	4	96	49
5	64	98	80	76	28	4	45
6	10	96	96	75	5	44	65
7	28	49	66	40	1	39	71

Because $b^{\mathrm{A}} = 10 > 6 = r^{\mathrm{A}}$ and $b^{\mathrm{B}} = 8 = r^{\mathrm{B}}$, both agents A and B belong to the second case. The coefficient matrices of $AG(\sigma_{\mathrm{A}})$ and $AG(\sigma_{\mathrm{B}})$ can be seen in the following two tables (Tables 4 and 5).

Table 4. The coefficient matrices of $AG(\sigma_{\mathrm{A}})$.

k/x	1	2	3	4	5	6	7
1	169.8707	90.2493	81.0000	**7.5595**	33.9711	27.4426	18.0992
2	182.0833	130.8327	**26.3162**	57.9963	14.2866	24.4966	18.5664
3	49.7590	131.7397	53.1056	62.0212	35.6659	14.5370	7.1204
4	183.4148	**39.6231**	88.1766	49.9797	5.2415	30.2381	13.3905
5	144.0000	132.6436	80.3320	53.8266	19.1802	**3.6342**	12.6515
6	**41.7743**	130.8327	90.7143	53.3534	6.0822	17.9753	16.1662
7	82.9879	83.5602	70.6628	35.0882	**2.0801**	16.5863	17.1464

The bold values are the optimal solution.

Table 5. The coefficient matrices of $AG(\sigma_{\mathrm{B}})$.

k*/*x	**1**	**2**	**3**	**4**	**5**	**6**
1	117.7817	62.5754	56.1623	**5.2415**	23.5543	19.0277
2	126.2495	90.7143	**18.2466**	40.2124	9.9058	16.9850
3	**34.5009**	91.3432	36.8214	43.0031	24.7293	10.0794
4	127.1727	**27.4731**	61.1383	34.6540	3.6342	20.9659
5	99.8440	91.9699	55.6991	37.3213	13.2988	2.5198
6	28.9647	90.7143	62.8978	36.9932	**4.2172**	12.4634

The bold values are the optimal solution.

Where the assignment result $\Theta_{1,4} = \Theta_{2,3} = \Theta_{4,2} = \Theta_{5,6} = \Theta_{6,1} = \Theta_{7,5} = 1$ of agent A can be calculated by Lingo, and place J_3^{A} in the last position of the sequence because of $\Theta_{3,x} = 0$, in which $x = 1, \ldots, 7$. Then, the optimal sequence of agent $\sigma_{\mathrm{A}} = \{J_6^{\mathrm{A}}, J_4^{\mathrm{A}}, J_2^{\mathrm{A}}, J_1^{\mathrm{A}}, J_7^{\mathrm{A}}, J_5^{\mathrm{A}}, J_3^{\mathrm{A}}\}$. The result of agent B is $\Omega_{1,4} = \Omega_{2,3} = \Omega_{3,1} = \Omega_{4,2} = \Omega_{6,5} = 1$ and $\Omega_{5,y} = 0$ $(y = 1, \ldots, 6)$; therefore, the optimal sequence is $\sigma_{\mathrm{B}} = \{J_3^{\mathrm{B}}, J_4^{\mathrm{B}}, J_2^{\mathrm{B}}, J_1^{\mathrm{B}}, J_6^{\mathrm{B}}, J_5^{\mathrm{B}}\}$.

Now determine whether $\sigma_1 = \tau\{\sigma_{\mathrm{A}}, \sigma_{\mathrm{B}}\}$ is feasible. Calculate the optimal resource allocation for the two agents based on (20) and (21), as follows:

$$\mathbf{w}^* = (w_1, w_2, w_3, w_4, w_5, w_6, w_7) = (32.6087, 30.9295, 20.5422, 5.9009, 1.6237, 2.8368, 5.5581)$$

$$\mathbf{u}^* = (u_1, u_2, u_3, u_4, u_5, u_6) = (29.9360, 23.8381, 15.8323, 4.5480, 3.6592, 2.1864)$$

The corresponding processing time is

$$(t_1^{\mathrm{A}}, t_2^{\mathrm{A}}, t_3^{\mathrm{A}}, t_4^{\mathrm{A}}, t_5^{\mathrm{A}}, t_6^{\mathrm{A}}, t_7^{\mathrm{A}}) = (0.0940, 0.4556, 1.6324, 4.8235, 14.3903, 44.7798, 144.0368)$$

$$(t_1^{\mathrm{A}}, t_2^{\mathrm{A}}, t_3^{\mathrm{A}}, t_4^{\mathrm{A}}, t_5^{\mathrm{A}}, t_6^{\mathrm{A}}) = (0.1886, 0.8277, 2.9301, 8.6663, 27.0924, 82.7570)$$

Obviously, $s_1^{\mathrm{A}}(\sigma_1) = 0$, $s_7^{\mathrm{A}}(\sigma_1) = 66.1756$, $s_1^{\mathrm{B}}(\sigma_1) = 210.2124$, and $s_6^{\mathrm{B}}(\sigma_1) = 249.9175$. The corresponding can be calculated by (22) and (23) to obtain $F^{\mathrm{A}}(\sigma_1) = 1766.1$ and $F^{\mathrm{B}}(\sigma_1) = 2661.7 > 2000$.

The sequence $\sigma_1 = \tau\{\sigma_{\mathrm{A}}, \sigma_{\mathrm{B}}\}$ is obviously not a feasible solution; now consider $\sigma_2 = \tau\{\sigma_{\mathrm{B}}, \sigma_{\mathrm{A}}\}$. At this point, $s_1^{\mathrm{B}}(\sigma_2) = 0$, $s_6^{\mathrm{B}}(\sigma_2) = 39.7051$, $s_1^{\mathrm{A}}(\sigma_2) = 122.4621$, and $s_7^{\mathrm{A}}(\sigma_2) = 188.6377$. The objective function values are $F^{\mathrm{B}}(\sigma_2) = 979.6931$ and $F^{\mathrm{A}}(\sigma_2) = 2500.9$. It can be seen that $F^{\mathrm{B}}(\sigma_2) = 979.6931 < 2000$; therefore, $\sigma_2 = \tau\{\sigma_{\mathrm{B}}, \sigma_{\mathrm{A}}\}$ is the optimal sequence. The optimal decision variables are $q^{\mathrm{B}}(\sigma_2) = 21.1761$ and $q^{\mathrm{A}}(\sigma_2) = 148.9323$. The optimal due date can be correspondingly written as

$$(d_1^*, d_2^*, d_3^*, d_4^*, d_5^*, d_6^*)(\mathrm{B}) = (21.3647, 22.0038, 24.1062, 29.8424, 48.2685, 103.9331)$$

and

$$(d_1^*, d_2^*, d_3^*, d_4^*, d_5^*, d_6^*, d_7^*)(\mathrm{A}) = (149.0263, 149.3879, 150.5647, 153.7558, 163.3226, 193.7121, 292.9691)$$

Example 2. *The modified date is as follows:* $(a^{\mathrm{A}}, a^{\mathrm{B}}) = (6, 4)$, $(b^{\mathrm{A}}, b^{\mathrm{B}}) = (7, 5)$, $(r^{\mathrm{A}}, r^{\mathrm{B}}) = (6, 8)$, *and other data are the same as Example 1. Because the workload of the job remains unchanged and the assignment is independent of the cost coefficients, the optimal sequences of* A *and* B *are constant, and the optimal resource allocations are also constant.*

Clearly, $r^{\mathrm{A}} < b^{\mathrm{A}}$ *and* $r^{\mathrm{B}} > b^{\mathrm{B}}$, *then agent* A *belongs to case 2 of (23) and agent* B *belongs to case 1 of (23). For the sequence* $\sigma_1 = \tau\{\sigma_{\mathrm{A}}, \sigma_{\mathrm{B}}\}$, $F^{\mathrm{B}}(\sigma_1) = 1633.5 < 2000$, *the the optimal sequence is* $\sigma_1 = \tau\{\sigma_{\mathrm{A}}, \sigma_{\mathrm{B}}\}$. *In addition,* $q^{\mathrm{A}} = 35.6330$, $q^{\mathrm{B}} = 0$, *and the corresponding optimal due date is*

$$(d_1^*, d_2^*, d_3^*, d_4^*, d_5^*, d_6^*, d_7^*)(\mathrm{A}) = (35.7270, 36.0886, 37.2654, 40.4565, 50.0233, 80.4128, 179.6698)$$

and

$$(d_1^*, d_2^*, d_3^*, d_4^*, d_5^*, d_6^*)(\mathrm{B}) = (0.1886, 0.8277, 2.9301, 8.6663, 27.0924, 82.7570)$$

6. Conclusions

This paper solved the two-agent scheduling problem with deteriorating jobs, where the processing time is a function of position-dependent workload, resource allocation, and a common deterioration rate. Under slack due date assignment, the goal was to find the optimal sequence and resource allocation through minimize the maximum value of earliness, tardiness, and decision variables (q) of one agent subject to an upper bound on cost value of the second agent. A polynomial time algorithm with $O(N^3)$ time is proposed for the case where the deterioration rate of each job is the same, where N is the maximum number of jobs between the two agents. Further research should consider the problem with a general linear deterioration or extend our model to the problems with variable processing times (see Sun and Geng [27], Wu et al. [28] and Wu et al. [29]).

Author Contributions: Methology: J.-B.W., L.-H.Z. and D.-Y.L.; Writing—original draft: L.-H.Z. and D.-Y.L.; Investigation: J.-B.W.; Writing—review and editing: J.-B.W., L.-H.Z. and D.-Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by LiaoNing Revitalization Talents Program grant number XLYC2002017.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare there is no conflict of interest.

References

1. Agnetis, A.; Pacciarelli, D.; Pacifici, A. Multi-agent single machine scheduling. *Ann. Oper. Res.* **2007**, *150*, 3–15. [CrossRef]
2. Tuong, N.H.; Soukhal, A.; Billaut, J.-C. Single-machine multi-agent scheduling problems with a global objective function. *J. Sched.* **2012**, *15*, 311–321.
3. Gu, M.; Gu, J.; Lu, X. An algorithm for multi-agent scheduling to minimize the makespan on m parallel machines. *J. Sched.* **2018**, *21*, 483–492. [CrossRef]
4. He, G.; Wu, J.; Lin, H. Two-agent bounded parallel-batching scheduling for minimizing maximum cost and makespan. *Discret. Optim.* **2022**, *45*, 100698. [CrossRef]
5. Wang, D.; Yu, Y.; Yin, Y.; Cheng, T.C.E. Multi-agent scheduling problems under multitasking. *Int. J. Prod. Res.* **2021**, *59*, 3633–3663. [CrossRef]
6. Wan, L.; Mei, J.; Du, J. Two-agent scheduling of unit processing time jobs to minimize total weighted completion time and total weighted number of tardy jobs. *Eur. J. Oper. Res.* **2021**, *290*, 26–35. [CrossRef]
7. 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**, *303*, 66–77. [CrossRef]
8. Falq, A.E.; Fouilhoux, P.; Kedad-Sidhoum, S. Dominance inequalities for scheduling around an unrestrictive common due date. *Eur. J. Oper. Res.* **2022**, *296*, 453–464. [CrossRef]
9. Wu, W.; Lv, D.-Y.; Wang, J.-B. Two due-date assignment scheduling with location-dependent weights and a deteriorating maintenance activity. *Systems* **2023**, *11*, 150. [CrossRef]
10. Liu, W.; Wang, X. Group technology scheduling with due-date assignment and controllable processing times. *Processes* **2023**, *11*, 1271. [CrossRef]
11. Mosheiov, G.; Oron, D.; Shabtay, D. On the tractability of hard scheduling problems with generalized due-dates with respect to the number of different due-dates. *J. Sched.* **2022**, *25*, 577–587. [CrossRef]
12. Hadayat, N.P.A.; Cakravastia, A.; Aribowo, W.; Halim, A.H. A single-stage batch scheduling model with m heterogeneous batch processors producing multiple items parts demanded at different due dates. *Int. J. Ind. Syst. Eng.* **2022**, *41*, 254–275. [CrossRef]
13. Liu, W.; Hu, X.; Wang, X. Single machine scheduling with slack due dates assignment. *Eng. Optim.* **2016**, *49*, 709–717. [CrossRef]
14. Liu, W.W.; Jiang, C. Due-date assignment scheduling involving job-dependent learning effects and convex resource allocation. *Eng. Optim.* **2020**, *52*, 74–89. [CrossRef]
15. Yin, Y.; Wang, D.J.; Wu, C.C.; Cheng, T.C.E. CON/SLK due date assignment and scheduling on a single machine with two agents. *Nav. Res. Logist.* **2016**, *63*, 416–429. [CrossRef]
16. Wang, D.J.; Yin, Y.Q.; Cheng, S.R.; Cheng, T.C.E.; Wu, C.C. Due date assignment and scheduling on a single machine with two competing agents. *Int. J. Prod. Res.* **2016**, *54*, 1152–1169. [CrossRef]
17. Wang, D.; Yu, Y.; Qiu, H.; Yin, Y.; Cheng, T.C.E. Two-agent scheduling with linear resource-dependent processing times. *Nav. Res. Logist.* **2020**, *67*, 573–591. [CrossRef]

18. Luo, C. A two-agent slack due-date assignment single machine scheduling problem with position-dependent workload and resource constraint. *J. Chongqing Norm. Univ. (Nat. Sci.)* **2022**, *39*, 1–8. (In Chinese)
19. Wu, C.-C.; Wu, W.-H.; Wu, W.-H.; Hsu, P.-H.; Yin, Y.; Xu, J. A single-machine scheduling with a truncated linear deterioration and ready times. *Inf. Sci.* **2014**, *256*, 109–125. [CrossRef]
20. Gawiejnowicz, S. *Models and Algorithms of Time-Dependent Scheduling*; Springer: Berlin, Germany, 2020.
21. Zhang, X.; Liu, S.-C.; Lin, W.-C.; Wu, C.-C. Parallel-machine scheduling with linear deteriorating jobs and preventive maintenance activities under a potential machine disruption. *Comput. Ind. Eng.* **2020**, *145*, 106482. [CrossRef]
22. Sun, X.; Liu, T.; Geng, X.-N.; Hu, Y.; Xu, J.-X. Optimization of scheduling problems with deterioration effects and an optional maintenance activity. *J. Sched.* 2023, *26*, 251–266. [CrossRef]
23. Yin, Y.; Cheng, T.C.E.; Wan, L.; Wu, C.C.; Liu, J. Two-agent single-machine scheduling with deteriorating jobs. *Comput. Ind. Eng.* **2015**, *81*, 177–185. [CrossRef]
24. Wang, Z.; Wei, C.M.; Wu, Y.B. Single machine two-agent scheduling with deteriorating jobs. *Asia-Pac. J. Oper. Res.* **2016**, *33*, 1650034. [CrossRef]
25. Li, D.; Li, G. Cheng, F. Two-agent single machine scheduling with deteriorating jobs and rejection. *Math. Probl. Eng.* **2022**, *2022*, 3565133. [CrossRef]
26. Mor, B., Mosheiov, G. A two-agent single machine scheduling problem with due-window assignment and a common flow-allowance. *J. Comb. Optim.* **2017**, *33*, 1454–1468. [CrossRef]
27. Sun, X.; Geng, X.-N. Single-machine scheduling with deteriorating effects and machine maintenance. *Int. J. Prod. Res.* **2019**, *57*, 3186–3199. [CrossRef]
28. Wu, C.-C.; Bai, D.; Zhang, X.; Cheng, S.-R.; Lin, J.-C.; Wu, Z.-L.; Lin, W.-C. A robust customer order scheduling problem along with scenario-dependent component processing times and due dates. *J. Manuf. Syst.* **2021**, *58*, 291–305. [CrossRef]
29. Wu, C.-C.; Bai, D.; Chen, J.-H.; Lin, W.-C.; Xing, L.; Lin, J.-C.; Cheng, S.-R. Several variants of simulated annealing hyper-heuristic for a single-machine scheduling with two-scenario-based dependent processing times. *Swarm Evol. Comput.* **2021**, *60*, 100765. [CrossRef]