RAIRO-Oper. Res. 55 (2021) 3701–3714
https://doi.org/10.1051/ro/2021175

RAIRO Operations Research
www.rairo-ro.org

SIMULTANEOUS OPTIMIZATION SCHEDULING WITH TWO AGENTS ON AN UNBOUNDED SERIAL-BATCHING MACHINE

Cheng He[1,*], Shi-Sheng Li[2] and Jing Wu[1]

Abstract. This paper considers a class of simultaneous optimization scheduling with two competitive agents on an unbounded serial-batching machine. The cost function of each agent depends on the completion times of its jobs only. According to whether the jobs from different agents can be processed in a common batch, compatible model and incompatible model are investigated. For the incompatible model, we consider batch availability and item availability. For each problem, we provide a polynomial-time algorithm that can find all Pareto optimal schedules.

Mathematics Subject Classification. 90C27, 90B35.

Received May 4, 2021. Accepted November 21, 2021.

1. Introduction

There are two agents A and B with job sets $\mathcal{F}^A = \left\{J_1^A, J_2^A, \cdots, J_{n_A}^A\right\}$ and $\mathcal{F}^B = \left\{J_1^B, J_2^B, \cdots, J_{n_B}^B\right\}$, respectively. Agents A and B must schedule their jobs on a common unbounded serial-batching machine and the jobs are processed in batches, where "unbounded" implies that the machine can process any number of jobs in a batch and "a batch" refers to a set of jobs which are processed jointly and contiguously. The processing time of a batch amounts to the sum of processing times of all jobs in the batch. A setup time is inserted whenever a new batch starts. According to whether the jobs from different agents can be processed in the same batch, we investigate compatible model (*i.e.*, the jobs of two agents can be processed in a common batch, short for *co*) and incompatible model (*i.e.*, the jobs of two agents cannot be processed in a common batch, short for *inco*). For compatible model, we suppose that the setup time equals to s. For incompatible model, we suppose that the setup time that is inserted before a batch which belongs to agent X is s_X ($X \in \{A, B\}$). Moreover, in incompatible model, two kinds of batch scheduling cases are presented according to the time when the jobs become available. In the case of *batch availability* (short for *batch-avail*), a job is available only when the batch to which it belongs has been processed. In the case of *item availability* (short for *item-avail*), a job becomes available immediately after it is completed processing (see [19]). In the paper, the objective function of agent A is a lateness-like objective function, such as $C_{\max}^A, L_{\max}^A, T_{\max}^A, WC_{\max}^A$ and that of agent B is a lateness-like objective function or the special total weighted completion time, such as $C_{\max}^B, L_{\max}^B, T_{\max}^B, WC_{\max}^B$, $\sum w_j^B C_j^B$ with $p_j^B = p$ or $w_j^B = w$ for any $1 \le j \le n_B$, where $C_{\max}^X, L_{\max}^X, T_{\max}^X, WC_{\max}^X$ and $\sum w_j^B C_j^B$ are

Keywords. Serial-batching, two-agent scheduling, compatible/incompatible, batch/item availability.

[1] School of Science, Henan University of Technology, Zhengzhou 450001, P.R. China.
[2] Department of Information and Computation Science, Zhongyuan University of Technology, Zhengzhou 450007, P.R. China.
*Corresponding author: hech202@163.com

© The authors. Published by EDP Sciences, ROADEF, SMAI 2021

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

the makespan, the maximum lateness, the maximum tardiness, the maximum weighted completion time (*i.e.*, $WC_{\max}^X = \max\{\sum w_j^X C_j^X : 1 \le j \le n_X\}$), of agent $X (X = A$ or $X = B)$ and the total weighted completion time of agent B respectively. Each agent's objective function depends on the completion times of its jobs only. The aim is to find all Pareto optimal schedules for the two-agents' objective functions under various scheduling environment. Here, two objective functions may on behalf of different profits of two decision-makers. Moreover, we first investigate the two problems with objective vectors $\left(L_{\max}^A, L_{\max}^B\right)$ and $\left(L_{\max}^A, \sum C_j^B\right)$, respectively, where objective vector (γ_1, γ_2) represents minimizing two criteria γ_1 and γ_2 simultaneously. Finally, we generalize the results of each problem to a class of problems.

The problems depicted above can be found in many manufacturing applications and many negotiation procedures. For example, a mill can handle orders from two types of agents. The agents' orders are interpreted as jobs to be processed. Agent A expects the maximum lateness of his jobs to be as small as possible, while agent B expects the maximum lateness or the total completion time of his jobs to be as small as possible. Moreover, the manufacturer is also concerned about minimizing any order delays which cause economic loss. For the purpose of meeting the needs of two agents to the maximum extent, the manufacturer needs to design some strategies to stimulate the agents to cooperate. This circumstance can be modelled as the simultaneous optimization problems under consideration, *i.e.*, objective vector is $\left(L_{\max}^A, L_{\max}^B\right)$ or $\left(L_{\max}^A, \sum C_j^B\right)$.

Serial-batching scheduling problems are urged by grouped jobs' processing environment with conversion times between different groups. For example, when the machine switches from processing one batch to another batch, the machine usually need to be changed a tool or to be cleaned, which shows that the machine needs a setup time before a new batch is processed [4]. Besides, the management and technical constraints (*e.g.*, different processing environment, etc.) lead to the compatibility and incompatibility of jobs from different agents [14,16].

For serial-batching scheduling problems with batch availability, Albers and Brucker [3] show that $1|\text{batch-avail}|\sum w_j C_j$ is strongly $\mathcal{NP}$-hard while $1|\text{batch-avail}|\sum C_j$ can be solved in $O(n \log n)$ time (see [6]). He *et al.* [10,11] present an $O(n^2)$-time algorithm for $1|\text{batch-avail}|(C_{\max}, L_{\max})$ and $1|\text{batch-avail}|(C_{\max}, \sum C_j)$, respectively. Geng *et al.* [8] solve $1|\text{batch-avail}|(C_{\max}, f_{\max})$ in $O\left(n^4\right)$ time. If the number of agents is given, the single-machine scheduling problems, with item availability, to minimize the maximum lateness or the number of tardy jobs or the total weighted completion time are polynomially solvable, while all these problems become very intractable when the number of agents is a variable [17, 20]. Reviews of the research on the topic are provided by Potts and Kovalyov [19] and Allahverdi [4].

The discuss on multi-agent scheduling originated from Baker and Smith [5] and Agnetis *et al.* [1]. Since it has been surveyed by Perez-Gonzalez and Framinan [18] and Agnetis *et al.* [2], we only review the results related to our study.

For two-agent constrained optimization scheduling on an unbounded serial-batching machine, Kovalyov *et al.* [14] investigate a series of batch availability models, in which closely related to our problems are the problems $1|\text{batch-avail, inco}, f_{\max}^A \le Q|\gamma^B$ with $\gamma^B \in \left\{f_{\max}^B, \sum C_j^B\right\}$. Yin *et al.* [21] generalize the work of [14] by adding a delivery cost for each manufacture batch; Yin *et al.* [22,23] study the problems in which there exist batch delivery cost and due date assignment, etc. Li *et al.* [15] study a series of item availability models, in which closely related to our problems are the problems $1|\text{item-avail, inco}, L_{\max}^A \le Q|\gamma^B$ with $\gamma^B \in \left\{L_{\max}^B, \sum C_j^B\right\}$ and $1|\text{item-avail, inco}, p_j^B = p, L_{\max}^A \le Q|\sum w_j^B C_j^B$. Li *et al.* [16] also investigated a series of job compatibility problems, in which closely related to our problems are the problems $1|\text{batch-avail, co}, f_{\max}^A \le Q|\gamma^B$ with $\gamma^B \in \left\{f_{\max}^B, \sum C_j^B\right\}$.

For two-agent simultaneous optimization scheduling on a serial-batching machine, to the best of our knowledge, the results are very few and the solved problems are very classical (see [2]). Feng *et al.* [7] give an $O\left(n_A + n_B^4\right)$-time algorithm for $1|\text{batch-avail, inco}|\left(C_{\max}^A, L_{\max}^B\right)$. He *et al.* [12] solve the problem $1|\text{batch-avail, inco}|\left(C_{\max}^A, f_{\max}^B\right)$ in $O\left(n_A + n_B^5\right)$ time. Agnetis *et al.* [2] show that problem $1|\text{batch-avail, inco}|\left(C_{\max}^A, \sum C_j^B\right)$ can be solved in $O\left(n_A n_B + n_B^4\right)$ time and $1|\text{batch-avail, inco}|\left(\sum U_j^A, \gamma^B\right)$ can be solved in polynomial time, where $\gamma^B \in \left\{\sum U_j^B, f_{\max}^B\right\}$. He and Lin [9] give an improved $O\left(n_B + n_A^2 \log n_A\right)$-time algorithm for $1|\text{batch-avail, inco}|\left(L_{\max}^A, C_{\max}^B\right)$ and $1|\text{batch-avail, inco}|\left(\sum_j C_j^A, C_{\max}^B\right)$, respectively. The paper investigates simultaneous optimization scheduling $1|\beta|\big(F^A_{\max}, F^B_{\max}\big)$ and $1|\beta, p^B_j = p$ or $w^B_j = w|\big(F^A_{\max}, \sum w^B_j C^B_j\big)$ and give polynomial-time algorithm for each problem respectively, where $F^X_{\max} \in \big\{C^X_{\max}, L^X_{\max}, T^X_{\max}, WC^X_{\max}\big\}$ ($X = A$ or B) and $\beta \in \{\{\text{batch-avail, inco}\}, \{\text{item-avail, inco}\}, \{\text{batch-avail, co}\}\}$.

The paper is arranged as follows. In Section 2, we elaborate some preliminaries and list an overview of the results in the paper. Section 3 is dedicated to two-agent problems of minimizing objective vectors $\big(L^A_{\max}, L^B_{\max}\big)$ and $\big(L^A_{\max}, \sum C^B_j\big)$ with batch availability and incompatibility. Section 4 is focused on two-agent problems of minimizing objective vectors $\big(L^A_{\max}, L^B_{\max}\big)$ and $\big(L^A_{\max}, \sum C^B_j\big)$ with item availability and incompatibility. Section 5 concentrates on two-agent problems of minimizing objective vectors $\big(L^A_{\max}, L^B_{\max}\big)$ and $\big(L^A_{\max}, \sum C^B_j\big)$ with batch availability and compatibility. Section 6 expands on the results in Sections 3–5. Section 7 gives a concluding remarks.

2. Preliminaries and overview of the results

Suppose that agents A and B have job sets $\mathcal{F}^A = \big\{J^A_1, J^A_2, \cdots, J^A_{n_A}\big\}$ and $\mathcal{F}^B = \big\{J^B_1, J^B_2, \cdots, J^B_{n_B}\big\}$, respectively. For $X \in \{A, B\}$, the jobs in $\mathcal{F}^X$ are also called X-jobs, job J^X_j ($X \in \{A, B\}$) has a processing time p^X_j, a weight w^X_j and a due date d^X_j. Let $n = n_A + n_B$. Given a feasible schedule σ, the completion time of job J^X_j ($X \in \{A, B\}$, $1 \le j \le n_X$) is denoted by $C^X_j(\sigma)$ in σ, $L^X_j(\sigma) = C^X_j(\sigma) - d^X_j$ and $T^X_j(\sigma) = \max\big\{0, C^X_j(\sigma) - d^X_j\big\}$ are the lateness and the tardiness of job J^X_j in σ, respectively; $L^X_{\max}(\sigma) = \max^n_{j=1} L^X_j(\sigma)$ and $T^X_{\max}(\sigma) = \max^n_{j=1} T^X_j(\sigma)$ are the maximum lateness and the maximum tardiness of agent X in σ, respectively; $WC^X_{\max}(\sigma) = \max\big\{w^X_j C^X_j(\sigma) : 1 \le j \le n_X\big\}$ and $\sum w^X_j C^X_j(\sigma)$ are the maximum weighted completion time and the total weighted completion time of agent X in σ, where the summation notation is taken over all jobs of agent X. When the jobs of two agents cannot be processed in a common batch (the model is called incompatible), an agent-dependent setup time s_X is inserted before each new batch of agent X ($X \in \{A, B\}$) is processed. When the jobs of two agents can be processed in the same batch (the model is called compatible), a setup time s is inserted before each new batch. The processing time of a batch is equal to the sum of processing times of all jobs in the batch. According to the time when the jobs become available, batch availability model and item availability model are investigated. In the case of batch availability, a job is available only when the batch to which it belongs has been processed. In the case of item availability, a job is available immediately after it is processed.

The paper considers the simultaneous optimization scheduling $1|\beta|\big(F^A_{\max}, F^B_{\max}\big)$ and $1|\beta, p^B_j = p$ or $w^B_j = w|\big(F^A_{\max}, \sum w^B_j C^B_j\big)$, where $\beta \in \{\{\text{batch-avail, inco}\}, \{\text{item-avail, inco}\}, \{\text{batch-avail, co}\}\}$ and $F^X_{\max} \in \big\{C^X_{\max}, L^X_{\max}, T^X_{\max}, WC^X_{\max}\big\}$ and $X = A$ or B. Note that "batch-avail" and "item-avail" represent that the considered problems are batch availability and item availability respectively, "inco" and "co" denote that A-jobs and B-jobs are incompatible and compatible respectively. The purpose is to find all Pareto optimal schedules in respect to two criteria in polynomial time for each problem.

Note that each job is available at time zero and each objective involved in the paper is regular (*i.e.*, nondecreasing in the completion times). So there exists an optimal schedule such that all jobs (batches) are processed continuously from time zero onwards. Throughout this paper, we focus our attention on the schedules with the property.

Definition 2.1. A feasible schedule σ is Pareto optimal, or nondominated, with respect to the performance criteria f and g if there is no feasible schedule π such that both $f(\pi) \le f(\sigma)$ and $g(\pi) \le g(\sigma)$, where at least one of the inequalities is strict. Besides, $(f(\sigma), g(\sigma))$ is called a Pareto optimal point corresponding to Pareto optimal schedule σ.

The following theorem provides a general approach, the so-called ε-constraint approach, for finding Pareto optimal schedules.

Theorem 2.2 ([13]). *Let y be the optimal value of constraint problem $\alpha|f \le \widehat{x}|g$ (where $\widehat{x}$ is a upper bound of f), and let x be the optimal value of the problem $\alpha|g \le y|f$. Then (x, y) is a Pareto optimal point for $\alpha||(f, g)$.*

Suppose that problem $\alpha|g \leq y|f$ is efficiently solvable. Then the other method to find Pareto optimal schedules is unilateral ε-constraint approach in the following.

Unilateral ε-constraint approach ([13]). Let (x^i, y^i) be a previously obtained point. Then solve the problem $\alpha|g < y^i|f$ (or $\alpha|g \leq y^i - 1|f$ if g is integral) and obtain a schedule π^{i+1} which give the next point $(x^{i+1}, y^{i+1}) = (f(\pi^{i+1}), g(\pi^{i+1}))$. Repeat the above process until the problem $\alpha|g < y^i|f$ is infeasible.

Theorem 2.3 ([13]). *Let (x^1, y^1) be the first point obtained by Unilateral ε-constraint approach, where y^1 be an upper bound of g. Then all Pareto optimal schedules of problem $\alpha||(f,g)$ can by obtained by Unilateral ε-constraint approach.*

Definition 2.4. A schedule is called EDD-EDD-schedule if A-jobs and B-jobs are sequenced in the Earliest Due Date (EDD) order, respectively.

Definition 2.5. A schedule is called EDD-SPT-schedule if A-jobs are processed in the EDD order and B-jobs are processed in the Shortest Processing Time (SPT) order.

The results in the paper are summarized in the Table 1.

3. Incompatible two-agent problems with batch availability

In this section, we study the two-agent scheduling problems under the batch availability and incompatible assumption.

Lemma 3.1 ([14]). *If problem $1|batch\text{-}avail,\ inco, L^A_{\max} \leq L|L^B_{\max}$ is feasible, then the problem has an optimal EDD-EDD-schedule.*

Lemma 3.2 ([14]). *If problem $1|batch\text{-}avail,\ inco, L^A_{\max} \leq L|\sum C^B_j$ is feasible, then the problem has an optimal EDD-SPT-schedule.*

3.1. 1|batch-avail, inco$|(L^A_{\max}, L^B_{\max})$

The goal of this subsection is to find all Pareto optimal schedules of 1|batch-avail, inco$|(L^A_{\max}, L^B_{\max})$. Note that for any Pareto optimal point of 1|batch-avail, inco$|(L^A_{\max}, L^B_{\max})$, there is an EDD-EDD-schedule corresponding to it by Lemma 3.1 and the definition of Pareto optimal point. Re-index the jobs such that $d^A_1 \leq d^A_2 \leq \cdots \leq d^A_{n_A}$ and $d^B_1 \leq d^B_2 \leq \cdots \leq d^B_{n_B}$.

For the feasible problem 1|batch-avail, inco, $L^A_{\max} \leq L, L^B_{\max} \leq L'|-$, Kovalyov *et al.* [14] presented a dynamic programming algorithm (denoted by **DP1**) to solve it, and obtained the following result.

Lemma 3.3 ([14]). *For problem $1|batch\text{-}avail,\ inco, L^A_{\max} \leq L, L^B_{\max} \leq L'|-$, DP1 can either find a feasible EDD-EDD-schedule or show that it is infeasible in $O(n_A n_B n)$ time.*

Lemma 3.4 ([10]). *The optimal schedule of problem $1|batch\text{-}avail|L_{\max}$ can be obtained in $O(n^2)$ time, where n is the total number of jobs.*

Let σ^* and π^* be the optimal schedules (with the minimum makespan $C_{\max}$) of the problem 1|batch-avail$|L_{\max}$ in respect to A-jobs and B-jobs, respectively. Let $L^* := L^A_{\max}(\sigma^*)$, $\overline{L} := L^* + C^B_{\max}(\pi^*)$, $L'^* := L^B_{\max}(\pi^*)$ and $\overline{L'} := L'^* + C^A_{\max}(\sigma^*)$ in the following. Then we have the following lemma by the definition of Pareto optimal point.

Lemma 3.5. *Let $(L_1, L'_1), (L_2, L'_2), \cdots, (L_k, L'_k)$ be all Pareto optimal points of problem $1|batch\text{-}avail,\ inco|(L^A_{\max}, L^B_{\max})$ and $L_1 < L_2 < \cdots < L_k$. Then $L_1 = L^*$ and $\overline{L} \geq L_k$ and $\overline{L'} \geq L'_1 > L'_2 > \cdots > L'_k = L'^*$.*

TABLE 1. Complexity results for two-agent scheduling.

Problem	Complexity	Reference
$1\|\text{batch-avail, inco}\|(C^A_{\max}, C^B_{\max})$	$O(n)$	Agnetis *et al.* [2]
$1\|\text{batch-avail, inco}\|(C^A_{\max}, L^B_{\max})$	$O(n_A + n_B^2 \log n_B)$	He and Lin [9]
$1\|\text{batch-avail, inco}\|(C^A_{\max}, F^B_{\max})$ where $F^B_{\max} \in \{T^B_{\max}, WC^B_{\max}\}$	$O(n_A + n_B^2 \log n_B)$	Theorem 6.4
$1\|\text{batch-avail, inco}\|(C^A_{\max}, \sum C^B_j)$	$O(n_A + n_B^2 \log n_B)$	He and Lin [9]
$1\|\text{batch-avail, inco}, p^B_j = p\|(C^A_{\max}, \sum w^B_j C^B_j)$	$O(n_A + n_B^2 \log n_B)$	Remark in Section 6
$1\|\text{batch-avail, inco}\|(L^A_{\max}, L^B_{\max})$	$O(n_A^3 n_B^3 n^2)$	Theorem 3.6
$1\|\text{batch-avail, inco}\|(F^A_{\max}, F^B_{\max})$ where $F^X_{\max} \in \{L^X_{\max}, T^X_{\max}, WC^X_{\max}\}$ and $X \in \{A, B\}$	$O(n_A^3 n_B^3 n^2)$	Theorem 6.12
$1\|\text{batch-avail, inco}\|(L^A_{\max}, \sum C^B_j)$	$O(n_A^5 n_B^4 n)$	Theorem 3.8
$1\|\text{batch-avail, inco}\|(F^A_{\max}, \sum C^B_j)$ where $F^A_{\max} \in \{T^A_{\max}, WC^A_{\max}\}$	$O(n_A^5 n_B^4 n)$	Theorem 6.13
$1\|\text{batch-avail, inco}, p^B_j = p\|(F^A_{\max}, \sum w^B_j C^B_j)$ where $F^A_{\max} \in \{L^A_{\max}, T^A_{\max}, WC^A_{\max}\}$	$O(n_A^5 n_B^4 n)$	Remark in Section 6
$1\|\text{item-avail, inco}\|(C^A_{\max}, C^B_{\max})$	$O(n)$	Theorem 6.3
$1\|\text{item-avail, inco}\|(C^A_{\max}, F^B_{\max})$ where $F^B_{\max} \in \{L^B_{\max}, T^B_{\max}, WC^B_{\max}\}$	$O(n_A + n_B^2)$	Theorem 6.5
$1\|\text{item-avail, inco}\|(C^A_{\max}, \sum C^B_j)$	$O(n_A + n_B^2)$	Theorem 6.5
$1\|\text{item-avail, inco}, p^B_j = p\|(C^A_{\max}, \sum w^B_j C^B_j)$	$O(n_A + n_B^2)$	Remark in Section 6
$1\|\text{item-avail, inco}\|(L^A_{\max}, L^B_{\max})$	$O(n_A^2 n_B^2 n)$	Theorem 4.4
$1\|\text{item-avail, inco}\|(F^A_{\max}, F^B_{\max})$ where $F^X_{\max} \in \{L^X_{\max}, T^X_{\max}, WC^X_{\max}\}$ and $X \in \{A, B\}$	$O(n_A^2 n_B^2 n)$	Theorem 6.12
$1\|\text{item-avail, inco}\|(L^A_{\max}, \sum C^B_j)$	$O(n_A^4 n_B^3)$	Theorem 4.6
$1\|\text{item-avail, inco}\|(F^A_{\max}, \sum C^B_j)$ where $F^A_{\max} \in \{T^A_{\max}, WC^A_{\max}\}$	$O(n_A^4 n_B^3)$	Theorem 6.13
$1\|\text{item-avail, inco}, p^B_j = p\|(F^A_{\max}, \sum w^B_j C^B_j)$ where $F^A_{\max} \in \{L^A_{\max}, T^A_{\max}, WC^A_{\max}\}$	$O(n_A^4 n_B^3)$	Remark in Section 6
$1\|\text{batch-avail, co}\|(C^A_{\max}, C^B_{\max})$	$O(n)$	Theorem 6.3
$1\|\text{batch-avail, co}\|(C^A_{\max}, F^B_{\max})$ where $F^B_{\max} \in \{L^B_{\max}, T^B_{\max}, WC^B_{\max}\}$	$O(n_A + n_B^4 \log n_B)$	Theorem 6.8
$1\|\text{batch-avail, co}\|(C^A_{\max}, \sum C^B_j)$	$O(n_A + n_B^5)$	Theorem 6.11
$1\|\text{batch-avail, co}, p^B_j = p\|(C^A_{\max}, \sum w^B_j C^B_j)$	$O(n_A + n_B^5)$	Remark in Section 6
$1\|\text{batch-avail, co}\|(L^A_{\max}, L^B_{\max})$	$O(n_A n_B n^3 \log n)$	Theorem 5.4
$1\|\text{batch-avail, co}\|(F^A_{\max}, F^B_{\max})$ where $F^X_{\max} \in \{L^X_{\max}, T^X_{\max}, WC^X_{\max}\}$ and $X \in \{A, B\}$	$O(n_A n_B n^3)$	Theorem 6.12
$1\|\text{batch-avail, co}\|(L^A_{\max}, \sum C^B_j)$	$O(n_A^4 n_B^3 n^2)$	Theorem 5.6
$1\|\text{batch-avail, co}\|(F^A_{\max}, \sum C^B_j)$ where $F^A_{\max} \in \{T^A_{\max}, WC^A_{\max}\}$	$O(n_A^4 n_B^3 n^2)$	Theorem 6.13
$1\|\text{batch-avail, co}, p^B_j = p\|(F^A_{\max}, \sum w^B_j C^B_j)$ where $F^A_{\max} \in \{L^A_{\max}, T^A_{\max}, WC^A_{\max}\}$	$O(n_A^4 n_B^3 n^2)$	Remark in Section 6

Let $\sigma = (B_1, B_2, \cdots, B_l)$ be any feasible EDD-EDD-schedule for $1|\text{batch-avail, inco}|\big(L^A_{\max}, L^B_{\max}\big)$, where B_i $(1 \le i \le l)$ is the i-th batch of σ. Since σ is an EDD-EDD-schedule, without loss of generality, let $\bigcup_{i=1}^{k} B_i = \big\{J^A_1, J^A_2, \cdots, J^A_{g_k}\big\} \bigcup \big\{J^B_1, J^B_2, \cdots, J^B_{h_k}\big\}$ for any $1 \le k \le l$. Let θ_k and ϑ_k denote the numbers of batches of agent A and agent B in the first k batches $B_1, B_2, \cdots, B_k$, respectively. Let $t(g_k, \theta_k, h_k, \vartheta_k)$ be the completion time of batch B_k. Then $\theta_k + \vartheta_k = k$ and $t(g_k, \theta_k, h_k, \vartheta_k) = \theta_k \cdot s_A + \sum_{j=1}^{g_k} p^A_j + \vartheta_k \cdot s_B + \sum_{j=1}^{h_k} p^B_j$, where $0 \le \theta_k \le g_k \le n_A$ and $0 \le \vartheta_k \le h_k \le n_B$. If the jobs contained in batch B_k belong to agent B, then let $B_k = \big\{J^B_{\lambda_k}, J^B_{\lambda_k+1}, \cdots, J^B_{h_k}\big\}$ because the jobs of agent B are scheduled in EDD order in σ, where $1 \le \vartheta_k \le \lambda_k \le h_k$. Hence $\max_{\lambda_k \le j \le h_k} L^B_j(\sigma) = t(g_k, \theta_k, h_k, \vartheta_k) - d^B_{\lambda_k}$ by $d^B_{\lambda_k} \le d^B_{\lambda_k+1} \le \cdots \le d^B_{h_k}$. Thus by the arbitraries of σ and k, we can derive all possible values of maximum lateness $L^B_{\max}$ belong to the set

$$\mathcal{C}^B := \big\{t(g_k, \theta_k, h_k, \vartheta_k) - d^B_{\lambda_k} | 0 \le \theta_k \le g_k \le n_A,\ 1 \le \vartheta_k \le \lambda_k \le h_k \le n_B\big\}. \tag{3.1}$$

If the jobs contained in batch B_k belong to agent A, then we infer similarly that all possible values of maximum lateness $L^A_{\max}$ belong to the set

$$\mathcal{C}^A := \big\{t(g_k, \theta_k, h_k, \vartheta_k) - d^A_{\lambda_k} | 1 \le \theta_k \le \lambda_k \le g_k \le n_A,\ 0 \le \vartheta_k \le h_k \le n_B\big\}. \tag{3.2}$$

Suppose that (L, L') is any Pareto optimal point for problem $1|\text{batch-avail, inco}|\big(L^A_{\max}, L^B_{\max}\big)$. Then we further get $L' \in \big\{p | p \in \mathcal{C}^B \text{ and } L'^{*} \le p \le \overline{L'}\big\}$ from Lemma 3.5. Let $\mathcal{E}^B := \big\{p | p \in \mathcal{C}^B \text{ and } L'^{*} \le p \le \overline{L'}\big\}$ and $\mathcal{E}^A := \big\{p | p \in \mathcal{C}^A \text{ and } L^{*} \le p \le \overline{L}\big\}$. Hence $L' \in \mathcal{E}^B$. Similar to the above discussion, we have $L \in \mathcal{E}^A$. Moreover, $|\mathcal{E}^A| \le O\big(n^3_A n^2_B\big)$ and $|\mathcal{E}^B| \le O(n^2_A n^3_B)$, where $|\mathcal{E}^X|$ denotes the number of the elements in set $\mathcal{E}^X, X \in \{A, B\}$.

Lemma 3.3 implies that problem $1|\text{batch-avail, inco}, L^A_{\max} \le L | L^B_{\max}$ can be solved by solving a series of feasible problems $1|\text{batch-avail, inco}, L^A_{\max} \le L, L^B_{\max} \le L'|-$ with decreasing L'. From the above discussion, we can only consider the values $L' \in \mathcal{E}^B$. Similarly, $1|\text{batch-avail, inco}, L^B_{\max} \le L' | L^A_{\max}$ can be solved by solving a series of feasible problems $1|\text{batch-avail, inco}, L^A_{\max} \le L, L^B_{\max} \le L'|-$ with decreasing L, where $L \in \mathcal{E}^A$.

Assume that $\mathcal{E}^A = \{l_1, l_2, \cdots, l_r\}$ and $\mathcal{E}^B = \{l'_1, l'_2, \cdots, l'_s\}$ with $l_1 < l_2 < \cdots < l_r$ and $l'_1 < l'_2 < \cdots < l'_s$ in the following, where $r \le O\big(n^3_A n^2_B\big)$ and $s \le O(n^2_A n^3_B)$. Then the following algorithm can solve problem $1|\text{batch-avail, inco}|\big(L^A_{\max}, L^B_{\max}\big)$.

Algorithm PO-$\big(L^A_{\max}, L^B_{\max}\big)$

Step 1. Let l_i $(1 \le i \le r)$ and l'_j $(1 \le j \le s)$ be defined as above, $k := r$, $h := 1$, $L := l_k$, $L' := l'_h$, $f := 0$ and $g := 0$.

Step 2. Solve problem $1|\text{batch-avail, inco}, L^A_{\max} \le L, L^B_{\max} \le L'|-$ by DP1. If the current constrained problem is infeasible, then go to Step 3; otherwise we get a new schedule σ_{f+1} and let $f := f + 1$ and $L := L^A_{\max}(\sigma_f)$ (it is obvious that $L \le l_k$ and $L \in \mathcal{E}^A$) and go to Step 4.

Step 3. If $h = s$, then return all Pareto optimal schedules $\pi_1, \pi_2, \cdots, \pi_g$ and stop. Otherwise $h < s$, let $h := h + 1$, $L' := l'_h$ and go back to Step 2.

Step 4. If $L = l_1$, then let $g := g + 1$ and $\pi_g := \sigma_f$ and return all Pareto optimal schedules $\pi_1, \pi_2, \cdots, \pi_g$ and stop. If $L > l_1$, then assume that $L = l_{i_k}$ (where $2 \le i_k \le k$). Let $k := i_k - 1$, $L := l_k$.

Step 5. Solve problem $1|\text{batch-avail, inco}, L^A_{\max} \le L, L^B_{\max} \le L'|-$ by DP1. If the current constrained problem is infeasible, then let $g := g + 1$ and $\pi_g := \sigma_f$ and go back to Step 3; otherwise we get a new schedule σ_{f+1} and let $f := f + 1$ and $L := L^A_{\max}(\sigma_f)$ and go back to Step 4.

Theorem 3.6. *Algorithm PO-$\big(L^A_{\max}, L^B_{\max}\big)$ can solve problem $1|\text{batch-avail, inco}|\big(L^A_{\max}, L^B_{\max}\big)$ in $O(n^3_A n^3_B n^2)$ time.*

Proof. The validity of Algorithm PO-$\big(L^A_{\max}, L^B_{\max}\big)$ is guaranteed by Theorem 2.2 and Lemmas 3.3 and 3.5 and the above discussion. Next, we discuss the time bound.

Step 1 takes $O\big((n_A^3 n_B^2 + n_A^2 n_B^3)\big) = O\big(n_A^2 n_B^2 n\big)$ time to compute all l_i $(1 \le i \le r)$ and l'_j $(1 \le j \le s)$ and takes $O\big(n_A^3 n_B^2 \log n + n_A^2 n_B^3 \log n\big) = O\big(n_A^2 n_B^2 n \log n\big)$ time to sort the sets $\mathcal{E}^{\mathcal{A}}$ and $\mathcal{E}^{\mathcal{B}}$. On the other hand, Steps 2 and 3 have at most $\left|\varepsilon^B\right| \le O(n_A^2 n_B^3)$ rounds and Steps 4 and 5 have at most $\left|\varepsilon^A\right| \le O(n_A^3 n_B^2)$ rounds. Each round needs $O(n_A n_B n)$ time by Lemma 3.3. Therefore, the overall time complexity is $O\big((n_A^2 n_B^3 + n_A^3 n_B^2) n_A n_B n\big) = O\big(n_A^3 n_B^3 n^2\big)$ time. □

3.2. 1|batch-avail, inco|$\left(L_{\max}^A, \sum C_j^B\right)$

Let (x, y) be a Pareto optimal point of problem 1|batch-avail, inco|$\big(L_{\max}^A, \sum C_j^B\big)$. Then there exists an EDD-SPT-schedule σ corresponding to (x, y) (*i.e.*, $(x, y) = \big(L_{\max}^A(\sigma), \sum C_j^B(\sigma)\big)$) by Lemma 3.2 and the definition of Pareto optimal point. Re-index the jobs such that $d_1^A \le d_2^A \le \cdots \le d_{n_A}^A$ and $p_1^B \le p_2^B \le \cdots \le p_{n_B}^B$.

For problem 1|batch-avail, inco, $L_{\max}^A \le L|\sum C_j^B$, Kovalyov *et al.* [14] presented a dynamic programming algorithm (denoted by **DP2**), and obtained the following result.

Lemma 3.7 ([14]). *Problem* 1|*batch-avail, inco*, $L_{\max}^A \le L|\sum C_j^B$ *can be solved by DP2 in* $O(n_A^2 n_B^2 n)$ *time.*

Similar to the discussion of Section 3.1, all possible values of the maximum lateness $L_{\max}^A$ belong to the set $\mathcal{C}^A$ and $|\mathcal{C}^A| = O\big(n_A^3 n_B^2\big)$, where $\mathcal{C}^A$ is defined as that in (3.2) of Section 3.1.

Next, we give an algorithm to solve problem 1|batch-avail, inco|$\big(L_{\max}^A, \sum C_j^B\big)$.

Algorithm PO-$\left(L_{\max}^A, \sum C_j^B\right)$

Step 1. Let l_i $(1 \le i \le r)$ be defined as that in Section 3.1, $k := r$, $h := 0$ and $L := l_k$.

Step 2. Solve problem 1|batch-avail, inco, $L_{\max}^A \le L|\sum C_j^B$ by DP2. If the current constrained problem is infeasible, then return all feasible schedules $\sigma_1, \sigma_2, \cdots, \sigma_h$ and go to Step 4; otherwise we get a new schedule σ_{h+1} and let $h := h + 1$ and $L := L_{\max}^A(\sigma_h)$ (it is obvious that $L \le l_k$ and $L \in \mathcal{C}^A$).

Step 3. If $L = l_1$, then return all feasible schedules $\sigma_1, \sigma_2, \cdots, \sigma_h$ and go to Step 4. If $L > l_1$, then assume that $L = l_{i_k}$ (where $2 \le i_k \le k$). Let $k := i_k - 1$, $L := l_k$ and go back to Step 2.

Step 4. For each schedule σ_j $(1 \le j \le h)$, compute the corresponding objective values $\big(L_{\max}^A(\sigma_j), \sum C_j^B(\sigma_j)\big)$ and pick out all Pareto optimal points and the corresponding Pareto optimal schedules.

Theorem 3.8. *Algorithm PO-*$\big(L_{\max}^A, \sum C_j^B\big)$ *can solve* 1|*batch-avail, inco*|$\big(L_{\max}^A, \sum C_j^B\big)$ *in* $O\big(n_A^5 n_B^4 n\big)$ *time.*

Proof. The validity of Algorithm PO-$\big(L_{\max}^A, \sum C_j^B\big)$ is guaranteed by Theorem 2.3 and Lemma 3.7 and the above discussion. Next, we discuss the time bound.

Step 1 takes $O\big(n_A^3 n_B^2\big)$ time to compute all l_i $(1 \le i \le r)$ and takes $O\big(n_A^3 n_B^2 \log n\big)$ time to sort the sets $\mathcal{C}^{\mathcal{A}}$. Step 2-Step 3 have at most $|\mathcal{C}^{\mathcal{A}}| = O\big(n_A^3 n_B^2\big)$ rounds (since $L = l_k \in \mathcal{C}^{\mathcal{A}}$). Each round needs $O(n_A^2 n_B^2 n)$ time by Lemma 3.7. Since $h \le O\big(n_A^3 n_B^2\big)$ and computing each pair of objective values needs $O(n)$ time and picking out all Pareto optimal points and the corresponding Pareto optimal schedules need $O(n)$ time (since $L_{\max}^A(\sigma_1) > L_{\max}^A(\sigma_2) > \cdots > L_{\max}^A(\sigma_h)$ and $\sum C_j^B(\sigma_1) \le \sum C_j^B(\sigma_2) \le \cdots \le \sum C_j^B(\sigma_h)$), Step 4 needs $O(n_A^3 n_B^2 n)$ time. Therefore, the overall time complexity is $O\big(n_A^5 n_B^4 n\big)$ time. □

4. Incompatible two-agent problems with item availability

In this section, we study the two-agent scheduling problems under the item availability and incompatible assumption. Notice that under the item availability assumption, if two batches of the same agent are processed consecutively, then these two batches can be merged into a large batch with only one setup time retained (which is impossible under the batch availability assumption), *i.e.*, the batches belonging to different agents appear alternately. We restrict our search to the schedules with this property.

Lemma 4.1 ([15]). *If problem* $1|$*item-avail, inco*$, L^A_{\max} \le L|L^B_{\max}$ *is feasible, then the problem has an optimal EDD-EDD-schedule.*

Lemma 4.2 ([15]). *If problem* $1|$*item-avail, inco*$, L^A_{\max} \le L|\sum C^B_j$ *is feasible, then the problem has an optimal EDD-SPT-schedule.*

4.1. 1|item-avail, inco|$\big(L^A_{\max}, L^B_{\max}\big)$

Let (x, y) be a Pareto optimal point of problem 1|item-avail, inco|$\big(L^A_{\max}, L^B_{\max}\big)$. Then there exists an EDD-EDD-schedule σ corresponding to (x, y) by Lemma 4.1 and the definition of Pareto optimal point. Re-index the jobs such that $d^A_1 \le d^A_2 \le \cdots \le d^A_{n_A}$ and $d^B_1 \le d^B_2 \le \cdots \le d^B_{n_B}$.

Let $\sigma = (B_1, B_2, \cdots, B_l)$ be any feasible EDD-EDD-schedule for 1|item-avail, inco|$\big(L^A_{\max}, L^B_{\max}\big)$. Suppose that job $J^B_\lambda \in \mathcal{F}^B$ and $J^B_\lambda \in B_k$. Then B_k is a batch of agent B. Further, batch B_{k-1} must be a batch of agent A by the rotation of batches from the different agents. Let θ_k and ϑ_k denote the numbers of batches of agent A and agent B in the first k batches $B_1, B_2, \cdots, B_k$, respectively. Then $\theta_k + \vartheta_k = k$ and

$$\vartheta_k = \begin{cases} \frac{k+1}{2} & \text{if } k \text{ is odd number} \\ \frac{k}{2} & \text{if } k \text{ is even number} \end{cases}$$

according to the rotation of batches from the different agents. Hence,

$$\theta_k = \begin{cases} \vartheta_k - 1 & \text{if } k \text{ is odd number} \\ \vartheta_k & \text{if } k \text{ is even number.} \end{cases} \tag{4.1}$$

Since A-jobs are scheduled in EDD order in σ, without loss of generality, let the A-jobs that are scheduled before J^B_λ are $J^A_1, J^A_2, \cdots, J^A_{g_k}$. From the definition of item availability, we know that the completion time of job J^B_λ is only related to the numbers θ_k and ϑ_k of batches and the jobs that is scheduled before job J^B_λ. So let $t(g_k, \theta_k, \lambda, \vartheta_k)$ be the completion time of job J^B_λ. Then $t(g_k, \theta_k, \lambda, \vartheta_k) = \theta_k \cdot s_A + \sum_{j=1}^{g_k} p^A_j + \vartheta_k \cdot s_B + \sum_{j=1}^{\lambda} p^B_j$ because σ is an EDD-EDD-schedule, where $0 \le \theta_k \le g_k \le n_A$ and $1 \le \vartheta_k \le \lambda \le n_B$. Hence $L^B_\lambda(\sigma) = t(g_k, \theta_k, \lambda, \vartheta_k) - d^B_\lambda$. Thus by the arbitraries of σ and λ, we can derive all possible values of maximum lateness $L^B_{\max}$ belong to the set

$$\mathcal{C}^B := \big\{t(g_k, \theta_k, \lambda, \vartheta_k) - d^B_\lambda | 0 \le \theta_k \le g_k \le n_A,\ 1 \le \vartheta_k \le \lambda \le n_B\big\} \text{ and } \theta_k \text{ and } \vartheta_k \text{ satisfy (4.1).}$$

Similarly, all possible values of maximum lateness $L^A_{\max}$ belong to the set

$$\mathcal{C}^A := \big\{t(\lambda, \theta_k, h_k, \vartheta_k) - d^A_\lambda | 1 \le \theta_k \le \lambda \le n_A,\ 0 \le \vartheta_k \le h_k \le n_B\big\}, \text{ where}$$
$$\vartheta_k = \begin{cases} \theta_k - 1 & \text{if } k \text{ is odd number} \\ \theta_k & \text{if } k \text{ is even number.} \end{cases} \tag{4.2}$$

Hence, $|\mathcal{C}^A| = O\big(n^2_A n_B\big)$ and $|\mathcal{C}^B| = O\big(n_A n^2_B\big)$.

Remark. Li *et al.* [15] present an $O(n^2_A n^2_B \log n)$-time algorithm for 1|item-avail, inco$, L^A_{\max} \le L|L^B_{\max}$, in which the time bound is decided by sorting set $\mathcal{C}^B$ with $|\mathcal{C}^B| = O(n^2_A n^2_B)$. Hence we can improve the time bound to $O(n_A n^2_B \log n)$ by $|\mathcal{C}^B| = O\big(n_A n^2_B\big)$.

Lemma 4.3 ([15]). *Problem* $1|$*item-avail, inco*$, L^A_{\max} \le L, L^B_{\max} \le L'|-$ *can be solved in* $O(n_A n_B)$ *time.*

By Lemma 4.3 and the above discussion, problem 1|item-avail, inco$, L^A_{\max} \le L|L^B_{\max}$ can be solved by solving a series of feasibility problems 1|item-avail, inco$, L^A_{\max} \le L, L^B_{\max} \le L'|-$ with the the decreasing values L' and $L' \in \mathcal{C}^B$. Similarly, 1|item-avail, inco$, L^B_{\max} \le L'|L^A_{\max}$ can also be solved by solving a series of feasibility problems 1|item-avail, inco$, L^A_{\max} \le L, L^B_{\max} \le L'|-$ with the decreasing values L and $L \in \mathcal{C}^A$.

By slightly modifying Algorithm PO-$\big(L^A_{\max}, L^B_{\max}\big)$ in Section 3.1 at the corresponding place, we derive the following result.

Theorem 4.4. *Problem* $1|$*item-avail, inco*$|\big(L^A_{\max}, L^B_{\max}\big)$ *can be solved in* $O\big(n_A^2 n_B^2 n\big)$ *time.*

4.2. $1|$item-avail, inco$|\Big(L^A_{\max}, \sum C^B_j\Big)$

Notice that for any Pareto optimal point of $1|$item-avail, inco$|\big(L^A_{\max}, \sum C^B_j\big)$, there is an EDD-SPT-schedule corresponding to it by Lemma 4.2 and the definition of Pareto optimal point. Re-index the jobs such that $d^A_1 \le d^A_2 \le \cdots \le d^A_{n_A}$ and $p^B_1 \le p^B_2 \le \cdots \le p^B_{n_B}$.

Lemma 4.5 ([15])**.** *Problem* $1|$*item-avail, inco*$, L^A_{\max} \le L|\sum C^B_j$ *can be solved in* $O(n_A^2 n_B^2)$ *time.*

Through the similar discussion in Section 4.1, all possible values of the maximum lateness $L^A_{\max}$ belong to the set $\mathcal{C}^A$ and $|\mathcal{C}^A| = O\big(n_A^2 n_B\big)$, where $\mathcal{C}^A$ is defined as that in Section 4.1. By slightly modifying Algorithm PO-$\big(L^A_{\max}, \sum C^B_j\big)$ in Section 3.1 at the corresponding place, we derive the following result.

Theorem 4.6. *Problem* $1|$*item-avail, inco*$|\big(L^A_{\max}, \sum C^B_j\big)$ *can be solved in* $O(n_A^4 n_B^3)$ *time.*

5. Compatible two-agent problems with batch availability

In this section, we study the two-agent scheduling problems under the batch availability and compatible assumption.

Lemma 5.1 ([16])**.** *If problem* $1|$*batch-avail, co*$, L^A_{\max} \le L|L^B_{\max}$ *is feasible, then the problem has an optimal EDD-EDD-schedule.*

Lemma 5.2 ([16])**.** *If problem* $1|$*batch-avail, co*$, L^A_{\max} \le L|\sum C^B_j$ *is feasible, then the problem has an optimal EDD-SPT-schedule.*

5.1. $1|$batch-avail, co$|\big(L^A_{\max}, L^B_{\max}\big)$

For any Pareto optimal point of $1|$batch-avail, co$|\big(L^A_{\max}, L^B_{\max}\big)$, there exists a corresponding EDD-EDD-schedule by Lemma 5.1 and the definition of Pareto optimal point. Re-index the jobs such that $d^A_1 \le d^A_2 \le \cdots \le d^A_{n_A}$ and $d^B_1 \le d^B_2 \le \cdots \le d^B_{n_B}$.

Let $\sigma = (B_1, B_2, \cdots, B_l)$ be any feasible EDD-EDD-schedule for $1|$batch-avail, co$|\big(L^A_{\max}, L^B_{\max}\big)$. Without loss of generality, for any $1 \le k \le l$, let $\bigcup_{i=1}^k B_i = \big\{J^A_1, J^A_2, \cdots, J^A_{g_k}\big\} \bigcup \big\{J^B_1, J^B_2, \cdots, J^B_{h_k}\big\}$ since σ is an EDD-EDD-schedule. Let $t(g_k, h_k, k)$ be the completion time of batch B_k. Then $t(g_k, h_k, k) = k \cdot s + \sum_{j=1}^{g_k} p^A_j + \sum_{j=1}^{h_k} p^B_j$, where $0 \le g_k \le n_A$ and $0 \le h_k \le n_B$ and $1 \le k \le g_k + h_k$. For any job $J^X_j \in B_k$ with $X = A$ or $X = B$ (for A-jobs and B-jobs are compatible), we have $L^X_j(\sigma) = t(g_k, h_k, k) - d^X_j$, where $1 \le j \le g_k$ if $X = A$; otherwise $1 \le j \le h_k$. By the arbitraries of σ and k, we can derive all possible values of maximum lateness $L^A_{\max}$ belong to the set

$$\mathcal{C}^A := \big\{t(g_k, h_k, k) - d^A_j | 1 \le j \le g_k \le n_A,\ 0 \le h_k \le n_B,\ 1 \le k \le g_k + h_k \le n\big\}.$$

Similarly, all possible values of maximum lateness $L^B_{\max}$ belong to the set

$$\mathcal{C}^B := \big\{t(g_k, h_k, k) - d^B_j | 0 \le g_k \le n_A,\ 1 \le j \le h_k \le n_B,\ 1 \le k \le g_k + h_k \le n\big\}.$$

Observe that, $|\mathcal{C}^A| = O(n_A^2 n_B n)$ and $|\mathcal{C}^B| = O(n_A n_B^2 n)$.

Lemma 5.3 ([16])**.** *Problem* $1|$*batch-avail, co*$, L^A_{\max} \le L, L^B_{\max} \le L'|-$ *can be solved in* $O(n \log n)$ *time.*

By Lemma 5.3 and the above discussion, problem $1|$batch-avail, co$, L^A_{\max} \le L|L^B_{\max}$ can be solved by solving a series of feasibility problems $1|$batch-avail, co$, L^A_{\max} \le L, L^B_{\max} \le L'|-$ with the decreasing values L', where $L' \in \mathcal{C}^B$. Similarly, $1|$batch-avail, co$, L^B_{\max} \le L'|L^A_{\max}$ can also be solved by solving a series of feasibility problems $1|$batch-avail, co$, L^A_{\max} \le L, L^B_{\max} \le L'|-$ with the decreasing values L, where $L \in \mathcal{C}^A$.

By slightly modifying Algorithm PO-$\big(L^A_{\max}, L^B_{\max}\big)$ in Section 3.1 at the corresponding place, we derive the following result.

Theorem 5.4. *Problem* $1|batch\text{-}avail,\ co|\big(L_{\max}^A, L_{\max}^B\big)$ *can be solved in* $O\big(n_A n_B n^3 \log n\big)$ *time.*

5.2. $1|$batch-avail, co$|\Big(L_{\max}^A, \sum C_j^B\Big)$

For any Pareto optimal point of problem $1|\text{batch-avail, co}|\big(L_{\max}^A, \sum C_j^B\big)$, there exists a corresponding EDD-SPT-schedule by Lemma 5.2 and the definition of Pareto optimal point. Re-index the jobs such that $d_1^A \le d_2^A \le \cdots \le d_{n_A}^A$ and $p_1^B \le p_2^B \le \cdots \le p_{n_B}^B$.

For problem $1|\text{batch-avail, co}, L_{\max}^A \le L|\sum C_j^B$, Li *et al.* [16] showed that $L_{\max}^A(\sigma) \in \mathcal{C}^A$ (defined in Sect. 5.1) and present a polynomial time algorithm for it.

Lemma 5.5 ([16])**.** *Problem* $1|batch\text{-}avail,\ co, L_{\max}^A \le L|\sum C_j^B$ *can be solved in* $O(n_A^2 n_B^2 n)$ *time.*

By modifying Algorithm PO-$\big(L_{\max}^A, \sum C_j^B\big)$ in Section 3.2 slightly on the corresponding place, we get the following result.

Theorem 5.6. *Problem* $1|batch\text{-}avail,\ co|\big(L_{\max}^A, \sum C_j^B\big)$ *can be solved in* $O(n_A^4 n_B^3 n^2)$ *time.*

6. A class of problems with lateness-like objective function

For each of the above problems studied in Sections 3–5, there is an optimal schedule with a specific structure, such as EDD-EDD or EDD-SPT that depends on the objective functions held by the two agents. We note that the specific structure is a key to solving our problems so that our methods are applicable to a class of problems with the specific structure. We introduce two concepts in [24].

– An objective function $F_{\max}^X$ of agent X that needs to be minimized is named *a lateness-like objective function* of agent X when $F_{\max}^X$ is regular and all X-jobs have a order (without loss of generality, denoted by $(J_1^X, J_2^X, \cdots, J_{n_X}^X)$) so that $F_1^X(t) \ge F_2^X(t) \ge \cdots \ge F_{n_X}^X(t)$ for any time t, where $F_{\max}^X(\sigma) = \max_{1\le j\le n_X}\{F_j^X(C_j^X(\sigma))\}$ and $X \in \{A, B\}$ for a given schedule σ. Moreover, order $\big(J_1^X, J_2^X, \cdots, J_{n_X}^X\big)$ is called *an EDD-like order.*

Observe that a scheduling problem in which agent X has a lateness-like objective function has an optimal schedule so that X-jobs are scheduled in the EDD-like order $\big(J_1^X, J_2^X, \cdots, J_{n_X}^X\big)$. For example, when $F_{\max}^X = L_{\max}^X$, the EDD-like order is $\big(J_1^X, J_2^X, \cdots, J_{n_X}^X\big)$ with $d_1^X \le d_2^X \le \cdots \le d_{n_X}^X$ (see [24]).

$C_{\max}^X$, $L_{\max}^X$, $T_{\max}^X$ and $WC_{\max}^X$ are common lateness-like objective functions. Assuming their EDD-like orders are $\big(J_1^X, J_2^X, \cdots, J_{n_X}^X\big)$. Then the EDD-like order is arbitrary for $C_{\max}^X$, the EDD-like order is $d_1^X \le d_2^X \le \cdots \le d_{n_X}^X$ for $L_{\max}^X$ and $T_{\max}^X$, the EDD-like order is $w_1^X \ge w_2^X \ge \cdots \ge w_{n_X}^X$ for $WC_{\max}^X$.

Lemma 6.1. *Let* (x, y) *be a Pareto optimal point of* $1|\beta|\big(F_{\max}^A, F_{\max}^B\big)$. *Then there is a corresponding Pareto optimal schedule so that all* X*-jobs are scheduled in the EDD-like order, where* $\beta \in \{\{batch\text{-}avail,\ inco\}, \{item\text{-}avail,\ inco\}, \{batch\text{-}avail,\ co\}\}$ *and* $F_{\max}^X \in \big\{C_{\max}^X, L_{\max}^X, T_{\max}^X, WC_{\max}^X\big\}$. *Especially, if* $F_{\max}^X = C_{\max}^X$, *then all* X*-jobs belong to a common batch.*

Proof. We can prove the lemma by job-shifting argument. □

Lemma 6.2. *Let* (x, y) *be a Pareto optimal point of* $1|\beta|\big(F_{\max}^A, \sum C_j^B\big)$. *Then there is a corresponding Pareto optimal schedule so that all* A*-jobs are scheduled in the EDD-like order and all* B*-jobs are scheduled in the SPT order, where* $F_{\max}^A \in \big\{C_{\max}^A, L_{\max}^A, T_{\max}^A, WC_{\max}^A\big\}$ *and* $\beta \in \{\{batch\text{-}avail,\ inco\}, \{item\text{-}avail,\ inco\}, \{batch\text{-}avail,\ co\}\}$. *Especially, if* $F_{\max}^A = C_{\max}^A$, *then all* A*-jobs belong to a common batch.*

Proof. We can prove the lemma by A-job-shifting argument and B-job-exchanging argument. □

Theorem 6.3. *Problem* $1|\beta|(C^A_{\max}, C^B_{\max})$ *with* $\beta \in \{\{$*item-avail, inco*$\}, \{$*batch-avail, co*$\}\}$ *can be solved in* $O(n)$ *time.*

Proof. By Lemma 6.1 and the symmetry of A and B, problem $1|$item-avail, inco$|(C^A_{\max}, C^B_{\max})$ has only two Pareto optimal schedules $(\mathcal{F}^A, \mathcal{F}^B)$ and $(\mathcal{F}^B, \mathcal{F}^A)$, $1|$batch-avail, co$|(C^A_{\max}, C^B_{\max})$ has only three Pareto optimal schedules $(\mathcal{F}^A, \mathcal{F}^B)$ and $(\mathcal{F}^B, \mathcal{F}^A)$ and $(\mathcal{F}^A\mathcal{F}^B)$, where $\mathcal{F}^X$ and $\mathcal{F}^A\mathcal{F}^B$ denote a batch composed of all X-jobs ($X = A$ or B) and a batch composed of all jobs, respectively. Hence, the problems can be solved in $O(n)$ time. □

Theorem 6.4. *Problem* $1|$*batch-avail, inco*$|(C^A_{\max}, F^B_{\max})$ *with* $F^B_{\max} \in \{T^B_{\max}, WC^B_{\max}\}$ *can be solved in* $O(n_A + n_B^2 \log n_B)$ *time.*

Proof. From Lemma 6.1 and EDD-like order, the proof for $1|$batch-avail, inco$|(C^A_{\max}, L^B_{\max})$ in [9] is applicable to $1|$batch-avail, inco$|(C^A_{\max}, F^B_{\max})$ with $F^B_{\max} \in \{T^B_{\max}, WC^B_{\max}\}$. □

Theorem 6.5. *Problems* $1|$*item-avail, inco*$|(C^A_{\max}, F^B_{\max})$ *with* $F^B_{\max} \in \{L^B_{\max}, T^B_{\max}, WC^B_{\max}\}$ *and* $1|$*item-avail, inco*$|(C^A_{\max}, \sum C^B_j)$ *can be solved in* $O(n_A + n_B^2)$ *time.*

Proof. Due to alternate batches from the different agents in any Pareto optimal schedule, each Pareto optimal schedule of each problem contains either two batches or three batches by Lemmas 6.1 and 6.2. Hence there is at most $O(n_B)$ Pareto optimal schedules. It takes $O(n_B)$ time to compute each objective vector when $\sum_j p^A_j$ is computed in advance and all B-jobs is sorted in advance. And sorting B-jobs needs $O(n_B \log n_B)$ time and computing $\sum_j p^A_j$ needs $O(n_A)$ time. So the total time complexity is $O(n_A + n_B \log n_B + n_B^2) = O(n_A + n_B^2)$. □

Lemma 6.6 ([16])**.** *For problem* $1|$*batch-avail, co*$, F^B_{\max} \le L|C^A_{\max}$*,* $C^A_{\max}$ *has at most* $O(n_B^2)$ *possible values, where* $F^B_{\max} \in \{L^B_{\max}, T^B_{\max}, WC^B_{\max}\}$*.*

Lemma 6.7 ([16])**.** *Problem* $1|$*batch-avail, co*$, F^B_{\max} \le L|C^A_{\max}$ *with* $F^B_{\max} \in \{L^B_{\max}, T^B_{\max}, WC^B_{\max}\}$ *can be solved in* $O(n_B \log n_B)$ *time if* $\sum_j p^A_j$ *are computed in advance and all* $O(n_B^2)$ *possible values of* $C^A_{\max}$ *are sorted in advance and all* B*-jobs are sorted in the nondecreasing order of their deadlines in advance.*

Proof. The feasibility problem $1|$batch-avail, co$, f_{\max} \le L|-$ can be solved in $O(n)$ if the deadlines of all jobs are given in advance (see [16]). So problem $1|$batch-avail, co$, F^B_{\max} \le L, C^A_{\max} \le L'|-$ can be solved in $O(n_B + \log n_B) = O(n_B)$ if $\sum_j p^A_j$ is computed in advance and all B-jobs are sorted in the nondecreasing order of their deadlines in advance by Lemma 6.1 (where all A-jobs are looked as a merged large job and $\log n_B$ is the time taken to insert the merged job into B-jobs). Since problem $1|$batch-avail, co$, F^B_{\max} \le L|C^A_{\max}$ can be solved by solving a series of feasibility problems $1|$batch-avail, co$, F^B_{\max} \le L, C^A_{\max} \le L'|-$ with varied L', where the varied L' is determined by binary search on all $O(n_B^2)$ possible values of $C^A_{\max}$, which needs $O(\log n_B)$ time if all $O(n_B^2)$ possible values of $C^A_{\max}$ is sorted in advance, and the order of all B-jobs keeps no change, problem $1|$batch-avail, co$, F^B_{\max} \le L|C^A_{\max}$ with $F^B_{\max} \in \{L^B_{\max}, T^B_{\max}, WC^B_{\max}\}$ can be solved in $O(n_B \log n_B)$ time if $\sum_j p^A_j$ is computed in advance and all $O(n_B^2)$ possible values of $C^A_{\max}$ is sorted in advance and all B-jobs are sorted in the nondecreasing order of their deadlines in advance. □

Theorem 6.8. *Problem* $1|$*batch-avail, co*$|(C^A_{\max}, F^B_{\max})$ *with* $F^B_{\max} \in \{L^B_{\max}, T^B_{\max}, WC^B_{\max}\}$ *can be solved in* $O(n_A + n_B^4 \log n_B)$ *time.*

Proof. Problem $1|\text{batch-avail, co}|\left(C_{\max}^A, F_{\max}^B\right)$ can be solved by solving a series of constraint problem $1|\text{batch-avail, co}, F_{\max}^B \le L|C_{\max}^A$ with the decreasing values L, where L is all possible values of $F_{\max}^B$.

From reference [16], $F_{\max}^B$ has at most $O\left(n_A n_B^2 n\right)$ possible values if $C_{\max}^A$ is replaced by $L_{\max}^A$. From Lemma 6.1, all A-jobs can be treated as a merged large job with processing time $\sum_j p_j^A$. Thus our problem is equivalent to $n_A = 1$ jobs. So $F_{\max}^B$ has at most $O\left(n_B^3\right)$ values in our problem. Note that the order of all B-jobs (provided that they are scheduled in nondecreasing order of their deadlines) has no change in each round. So if we sort all B-jobs in advance, then each time sorting all jobs needs only to insert one merged A-jobs into B-jobs, which takes $O(\log n_B)$ time. And computing $\sum_j p_j^A$ needs $O(n_A)$ time and sorting all B-jobs needs $O(n_B \log n_B)$ time in advance. Sorting all $O\left(n_B^2\right)$ possible values of $C_{\max}^A$ needs $O\left(n_B^2 \log n_B\right)$ time. Hence, the total time complexity is $O\left(n_A + n_B \log n_B + n_B^2 \log n_B + n_B \log n_B \cdot n_B^3\right) = O\left(n_A + n_B^4 \log n_B\right)$ time. □

Lemma 6.9 ([16])**.** *Problem* $1|\textit{batch-avail, co}, F_{\max}^A \le L|\sum C_j^B$ *can be solved in* $O\left(n n_A^2 n_B^2\right)$ *time, where* $F_{\max}^B \in \left\{L_{\max}^B, T_{\max}^B, WC_{\max}^B\right\}$.

Corollary 6.10. *Problem* $1|\textit{batch-avail, co}, C_{\max}^A \le L|\sum C_j^B$ *can be solved in* $O(n_B^3)$ *time if* $\sum_j p_j^A$ *is computed in advance.*

Theorem 6.11. *Problem* $1|\textit{batch-avail, co}|\left(C_{\max}^A, \sum C_j^B\right)$ *can be solved in* $O(n_A + n_B^5)$ *time.*

Proof. Since $C_{\max}^A$ has at most $O(n_B^2)$ possible values (see ([16])), sorting the values needs $O(n_B^2 \log n_B)$ time and computing $\sum_j p_j^A$ needs $O(n_A)$ time in advance. The remaining proof is similar to that of Theorem 6.8. □

Since lateness-like objective functions and the maximum lateness have a similar property (see Lemmas 6.1 and 6.2), the methods used to solve problem$1|\beta|\left(L_{\max}^A, L_{\max}^B\right)$ and problem $1|\beta|\left(L_{\max}^A, \sum C_j^B\right)$ in Sections 3–5 are also applicable to problems $1|\beta|\left(F_{\max}^A, F_{\max}^B\right)$ and $1|\beta|\left(F_{\max}^A, \sum C_j^B\right)$, where $\beta \in \{\{\text{batch-avail, inco}\}, \{\text{item-avail, inco}\}, \{\text{batch-avail, co}\}\}$ and $F_{\max}^X \in \left\{L_{\max}^X, T_{\max}^X, WC_{\max}^X\right\}$ ($X = A$ or B). Hence we have the following theorems.

Theorem 6.12. *Problems* $1|\beta|\left(F_{\max}^A, F_{\max}^B\right)$ *and* $1|\beta|\left(L_{\max}^A, L_{\max}^B\right)$ *have the same time complexity, where* $\beta \in \{\{\textit{batch-avail, inco}\}, \{\textit{item-avail, inco}\}, \{\textit{batch-avail, co}\}\}$ *and* $X \in \{A, B\}$ *and* $F_{\max}^X \in \left\{L_{\max}^X, T_{\max}^X, WC_{\max}^X\right\}$.

Theorem 6.13. *Problems* $1|\beta|\left(F_{\max}^A, \sum C_j^B\right)$ *and* $1|\beta|\left(L_{\max}^A, \sum C_j^B\right)$ *have the same time complexity, where* $\beta \in \{\{\textit{batch-avail, inco}\}, \{\textit{item-avail, inco}\}, \{\textit{batch-avail, co}\}\}$ *and* $F_{\max}^A \in \{T_{\max}^A, WC_{\max}^A\}$.

Remark. For $\beta \in \{\{\text{batch-avail, inco}\}, \{\text{item-avail, inco}\}, \{\text{batch-avail, co}\}\}$ and $F_{\max}^A \in \{C_{\max}^A, L_{\max}^A, T_{\max}^A, WC_{\max}^A\}$, Pareto optimal schedules of problem $1|\beta|\left(F_{\max}^A, \sum C_j^B\right)$ and Pareto optimal schedules of problem $1|\beta, p_j^B = p|\left(F_{\max}^A, \sum w_j^B C_j^B\right)$ have a similar property, *i.e.*, their A-jobs are scheduled in the EDD-like order, and all B-jobs are scheduled in the SPT order for former problem and all B-jobs are scheduled in the SPT-like order (*i.e.*, $w_1^B \ge w_2^B \ge \cdots \ge w_{n_B}^B$) for latter problem. Therefore, problems $1|\beta|\left(F_{\max}^A, \sum C_j^B\right)$ and $1|\beta, p_j^B = p|\left(F_{\max}^A, \sum w_j^B C_j^B\right)$ have the same time complexity.

7. Concluding remarks

In the foregoing discussion, we study two-agent simultaneous optimization scheduling, in which the objective function of agent A is lateness-like objective function, such as $C_{\max}^A, L_{\max}^A, T_{\max}^A, WC_{\max}^A$ and that of agent B is a lateness-like objective function or the special total weighted completion time, on an unbounded serial-batching machine. Moreover, the problems are considered under three cases: batch availability and incompatibility, item availability and incompatibility, and batch availability and compatibility. For all problems studied in the paper, we give a polynomial-time algorithm, respectively. On the one hand, our future work would be to generalize the objective vectors, for example, $(f^A_{\max}, f^B_{\max})$ and $(\sum C^B_j, \sum C^B_j)$ and $(f^A_{\max}, \sum C^B_j)$. On the other hand, we can also consider the corresponding bounded cases for the problems in the paper.

Acknowledgements. He was supported by the Key Research Project of Henan Higher Education Institutions (20A110003) and STAPHNOS (No. 2020-70) and IFPHNUT (No. 2020ZKCJ08). Li was supported by the Key Research Project of Henan Higher Education Institutions (20A110037) and the Young Backbone Teachers training program of Zhongyuan University of Technology (2018XQG15).

References

[1] A. Agnetis, P.B. Mirchani, D. Pacciarelli and A. Pacifici, Scheduling problems with two competing agents. *Oper. Res.* **52** (2004) 229–242.

[2] A. Agnetis, J.-C. Billaut, S. Gawiejnowicz, D. Pacciarelli and A. Soukhal, Multiagent Scheduling: Models and Algorithms. Springer-Verlag (2014).

[3] S. Albers and P. Brucker, The complexity of one-machine batching problems. *Discrete Appl. Math.* **47** (1993) 87–107.

[4] A. Allahverdi, The third comprehensive survey on scheduling problems with setup times/costs. *Eur. J. Oper. Res.* **246** (2015) 345–378.

[5] K.R. Baker and J.C. Smith, A multiple-criterion model for machine scheduling. *J. Scheduling* **6** (2003) 7–16.

[6] E.G. Coffman, M. Yannakakis, M.J. Magazine and C. Santos, Batch sizing and job sequencing on a single machine. *Ann. Oper. Res.* **26** (1990) 135–147.

[7] Q. Feng, Z.Y. Yu and W.P. Shang, Pareto optimization of serial-batching scheduling problems on two agents. In: The 2011 International Conference on Advanced Mechatronic Systems. IEEE (2011).

[8] Z.C. Geng, J.J. Yuan and J.L. Yuan, Scheduling with or without precedence relations on a serial-batch machine to minimize makespan and maximum cost. *Appl. Math. Comput.* **332** (2018) 1–18.

[9] C. He and H. Lin, Improved algorithms for two-agent scheduling on an unbounded serial-batching machine. *Discrete Optim.* **41** (2021) 100655.

[10] C. He, Y.X. Lin and J.J. Yuan, Bicriteria scheduling of minimizing maximum lateness and makespan on a serial-batching machine. *Found. Comput. Decis. Sci.* **33** (2008) 369–376.

[11] C. He, Y.X. Lin and J.J. Yuan, A DP algorighm for minimizing makespan and total completion time on a series-batching machine. *Inf. Process. Lett.* **109** (2009) 603–607.

[12] C. He, C.Q. Xu and H. Lin, Serial-batching scheduling with two agents to minimize makespan and maximum cost. *J. Scheduling* **23** (2020) 609–617.

[13] H. Hoogeveen, Multicriteria scheduling. *Eur. J. Oper. Res.* **167** (2005) 592–623.

[14] M.Y. Kovalyov, A. Oulamara and A. Soukhal, Two-agent scheduling with agent specific batches on an unbounded serial batching machine. *J. Scheduling* **18** (2015) 423–434.

[15] S.S. Li, R.X. Chen and Q. Feng, Scheduling two job families on a single machine with two competitive agents. *J. Comb. Optim.* **32** (2016) 784–799.

[16] S.S. Li, T.C.E. Cheng, C.T. Ng and J.J. Yuan, Two-agent scheduling on a single sequential and compatible batching machine. *Nav. Res. Logistics* **64** (2017) 628–641.

[17] C.L. Monma and C.N. Potts, On the complexity of scheduling with batch setup times. *Oper. Res.* **37** (1989) 798–804.

[18] P. Perez-Gonzalez and J.M. Framinan, A common framework and taxonomy for multicriteria scheduling problems with interfering and competing jobs: multi-agent scheduling problems. *Eur. J. Oper. Res.* **235** (2014) 1–16.

[19] C.N. Potts and M.Y. Kovalyov, Scheduling with batching: a review. *Eur. J. Oper. Res.* **120** (2000) 228–249.

[20] S. Webster and K.R. Baker, Scheduling groups of jobs on a single machine. *Oper. Res.* **43** (1995) 692–703.

[21] Y.Q. Yin, Y. Wang, T.C.E.Cheng, D.J. Wang and C.-C. Wu, Two-agent single-machine scheduling to minimize the batch delivery cost. *Comput. Ind. Eng.* **92** (2016) 16–30.

[22] Y.Q. Yin, Y.J. Yang, D.J. Wang, T.C.E.Cheng and C.-C. Wu, Integrated production, inventory, and batch delivery scheduling with due date assignment and two competing agents. *Nav. Res. Logistics* **65** (2018) 393–409.

[23] Y.Q. Yin, D.D. Li, D.J. Wang and T.C.E. Cheng, Single-machine serial-batch delivery scheduling with two competing agents and due date assignment. *Ann. Oper. Res.* **298** (2021) 497–523.

[24] J.J. Yuan, C.T. Ng and T.C.E. Cheng, Scheduling with release dates and preemption to minimize multiple max-form objective functions. *Eur. J. Oper. Res.* **280** (2020) 860–875.

Subscribe to Open (S2O)

A fair and sustainable open access model

This journal is currently published in open access under a Subscribe-to-Open model (S2O). S2O is a transformative model that aims to move subscription journals to open access. Open access is the free, immediate, online availability of research articles combined with the rights to use these articles fully in the digital environment. We are thankful to our subscribers and sponsors for making it possible to publish this journal in open access, free of charge for authors.

Please help to maintain this journal in open access!

Check that your library subscribes to the journal, or make a personal donation to the S2O programme, by contacting subscribers@edpsciences.org

More information, including a list of sponsors and a financial transparency report, available at: https://www.edpsciences.org/en/maths-s2o-programme