Models, constructive heuristics, and benchmark instances for the flexible job shop scheduling problem with nonlinear routes and position-based learning*

Kennedy A. G. Araujo† Ernesto G. Birgin‡ Débora P. Ronconi§

April 1, 2024¶

Abstract

This paper addresses the flexible job shop scheduling problem with nonlinear routes and position-based learning effect. In this variant of the flexible job shop scheduling problem, precedence constraints of the operations constituting a job are given by an arbitrary directed acyclic graph, generalizing the classical case in which a total order is imposed. Additionally, it is assumed that a worker performing an operation at a machine undergoes a learning process: the more times a worker has performed an operation, the faster he or she will be able to complete it on subsequent repetitions. Mixed integer programming and constraint programming models are presented and compared in the present work. In addition, sets of benchmark instances and constructive heuristics are introduced. The problem we study here corresponds to modern problems of great relevance in the printing industry. The framework we introduce aims to assist authors in the design of novel, effective, and efficient methods. Instances with optimal solutions provide a means to evaluate the quality of solutions found by other methods. Constructive heuristics serve two purposes. First, they provide an initial solution to off-the-shelf exact methods, offering a warm start when addressing instances of the proposed models. They also lay the groundwork for developing algorithms, such as list scheduling, neighborhoods, local searches, and ultimately, metaheuristics.

Keywords: flexible job shop scheduling problem, nonlinear routes, learning effect, models, instances, constructive heuristics.

---

*This work has been partially supported by the Brazilian agencies FAPESP (grants 2013/07375-0, 2022/05803-3, and 2023/08706-1) and CNPq (grants 311536/2020-4 and 302073/2022-1).

†Department of Applied Mathematics, Institute of Mathematics and Statistics, University of São Paulo, Rua do Matão, 1010, Cidade Universitária, 05508-090, São Paulo, SP, Brazil. e-mail: kennedy94@ime.usp.br. ORCID: 0000-0003-2917-025X.

‡Department of Computer Science, Institute of Mathematics and Statistics, University of São Paulo, Rua do Matão, 1010, Cidade Universitária, 05508-090, São Paulo, SP, Brazil. e-mail: egbirgin@ime.usp.br. ORCID: 0000-0002-7466-7663. Corresponding author.

§Department of Production Engineering, Polytechnic School, University of São Paulo, Av. Prof. Luciano Gualberto, 1380, Cidade Universitária, 05508-010 São Paulo, SP, Brazil. e-mail: dronconi@usp.br. ORCID: 0000-0001-7647-2328.

¶Revisions made on March 1, 2026, and June 25, 2026.

1 Introduction

The online print shop (OPS) scheduling problem is a modern complex scheduling problem that falls within the scope of the flexible job shop scheduling problem with nonlinear routes. In the OPS, customers place online orders to purchase products (such as books, greeting cards, wedding invitations, photo albums, among many other possibilities) that require printing. Since the raw material (paper) is expensive, orders that share the same type of paper and printing are grouped together and a cutting stock problem is solved to accommodate their sheets in large pieces of paper that are printed and then cut. On the other hand, the production of a book, for example, includes a variety of operations such as layout, printing, gluing, packaging, etc. Thus, the jobs of the OPS scheduling problem, which are the union of the operations of all the orders whose printing operations have been combined, are diverse and complex. Some operations must satisfy precedence relationships, whereas other operations are unrelated and can even be processed in parallel. Overall, the precedence relationships between the operations in a job can be represented by an acyclic directed graph. This is one of the main features of the OPS scheduling problem. In [22, 23], a variety of additional constraints have been considered. In this work, we focus on the fact that several operations of the OPS scheduling problem require human interaction. Since human workers learn by repeatedly performing an operation, we consider that processing times are subject to learning effects.

The flexible job shop (FJS) scheduling problem with nonlinear routes and position-based learning effects, when the objective is to minimize the makespan, is NP-hard, since it generalizes the classical job shop scheduling problem, which is known to be NP-hard [14]. The nonlinear routes feature refers to the fact that the precedence constraints imposed on the operations of a job are given by an arbitrary directed acyclic graph instead of the usual linear order enforced in the FJS scheduling. Many real-life scheduling problems fit into this scope, such as, for example, problems in the printing industry [4, 22, 23], mold manufacturing industry [12], and glass industry [1]. In [23], the FJS with nonlinear routes and several additional features, such as, for example, resumable operations, periods of unavailability of the machines, sequence-dependent setup times, partial overlapping between operations with precedence constraints, and fixed operations, was addressed. A review of the literature on FJS scheduling, including nonlinear routes, was conducted in [11]. However, learning effects have not yet been considered in connection with such routes. The present work is devoted to this problem and presents only a first step towards its effective and efficient resolution. Mixed integer linear programming (MILP) and constraint programming (CP) models and a relatively large set of instances are introduced for the purpose of producing a benchmark test set. Since, leaving aside very small instances, commercial exact solvers alone are hardly even able to find a feasible solution, constructive heuristics are proposed with the goal of enhancing their performance. Overall, this work takes a first step towards solving the proposed problem and provides a solid and robust basis for the future development of more sophisticated heuristic and metaheuristic methods.

In classical scheduling problems, the processing time of a given operation on a given machine is a fixed input parameter. However, there are real-life situations in which the manufacturing process involves repetitive manual tasks and the worker undergoes a learning process that results in a reduction of the execution time of his/her task. For instance, workers can get proficient at performing assemblies quickly, or more confident and skillful in manipulating hardware, software,

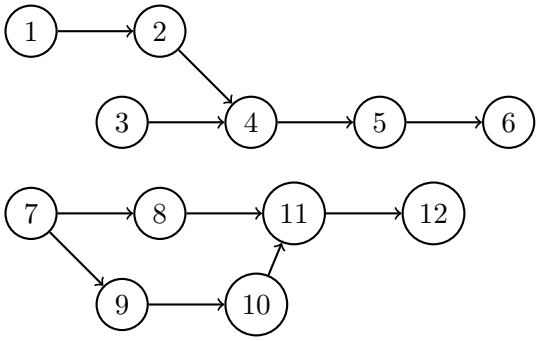
and/or raw materials. If we consider that the reduction in the operation processing time is related to the number of times the worker has already performed the operation, this phenomenon is known as a position-based learning effect. The pioneering works applying the concept of learning effect to scheduling problems are [6, 9, 15]. Surveys on the subject can be found in [3, 7, 16].

A brief literature review in chronological order of the FJS with nonlinear routes follows. In [13], the problem was addressed by considering non-fixed intervals of machine unavailability for preventive maintenance. A multi-objective MILP formulation and a hybrid multi-objective genetic algorithm were proposed. In [12] and [19], process flexibility was also taken into account, which means that the same result can be obtained with different operation sequences. In [12] a branch-and-bound algorithm was proposed, while [19] considered a symbiotic evolutionary algorithm. In [25], where a MILP formulation was presented to minimize the weighted tardiness, a hybrid bacterial foraging optimization algorithm was developed. Furthermore, the algorithm was enhanced by a local search method based on the manipulation of critical operations. A research that addresses the FJS with nonlinear routes and sequence-dependent setup times can be found in [8]. The authors proposed a knowledge-based cuckoo search algorithm associated with a reinforcement learning strategy for self-adjustment. In [18], MILP and CP models and a hybrid evolutionary algorithm with local search mechanisms were introduced. A variation in which the processing of each operation requires multiple resources was considered in [17], in which models are presented and some properties of the problems are analyzed.

In [1], an application in the glass industry was described and a heuristic approach combining local search and priority rules was proposed to minimize the total cost related to final product completion times. Other industrial environments, such as the printing industry, have also been modeled as an FJS with nonlinear routes. Regarding this particular application, [24] suggested a bi-objective genetic algorithm based on NSGA II to solve the problem. In [4], a MILP model and a constructive heuristic were presented, while [5] introduced a list scheduling algorithm and its extension to a beam search method. In [22, 23], formulations using constrained programming and mixed integer linear programming were established, and trajectory and population meta-heuristics were introduced. In [28], it was considered the flight deck scheduling problem, which is an FJS with nonlinear routes with additional constraints that state that some operations must be completed before others. The problem was described through its MILP formulation and instances were solved with a differential evolution type method. In [2], the scheduling of repair orders and worker assignment in an automotive repair shop was analyzed. The main scheduling problem is a two-resource FJS with nonlinear routes. The problem was modeled by extending the formulation introduced in [4] and an iterated greedy heuristic was presented.

Let us consider an illustrative example of the FJS with nonlinear routes and position-based learning effect. The example has 12 operations and 3 machines. Figure 1 shows the precedence relationships between the operations and the standard processing times of each operation on each machine. The small table with the standard processing times shows that in the FJS scheduling problem each operation can be processed by one or more machines (situation known as routing flexibility), generalizing the job shop (JS) scheduling problem in which each operation can be processed by only one machine. The concept of job is implicitly defined by the directed acyclic graph (DAG) and corresponds to a set of operations that have some dependency relationship between them. The figure makes it clear that, in the FJS scheduling problem with nonlinear routes, the dependencies of the operations of a job are given by an arbitrary acyclic directed

graph instead of the total order of the FJS scheduling problem. In this example, there are two jobs. The first job has six operations, numbered from 1 to 6. The second job also has six operations, numbered from 7 to 12.



		Machines		
		1	2	3
Operations	1	10	20	15
	2	20	15	5
	3	10	–	20
	4	–	30	–
	5	30	40	10
	6	20	–	30
	7	–	10	20
	8	40	10	–
	9	–	40	20
	10	10	20	10
	11	20	10	–
	12	–	–	15

Figure 1: On the left, representation of operations' precedence constraints by a directed acyclic graph $D = (\mathcal{O}, \widehat{A})$, where $\mathcal{O} = \{1, 2, \ldots, 12\}$ represents the set of operations and $\widehat{A}$ is the set of arcs that represent the precedence constraints. On the right, standard processing times of the twelve operations on each of the three machines. In the table, cells containing "–" indicate that the machine cannot process the operation.

A feasible solution to the instance of Figure 1 can be illustrated by a DAG in which a source node s and a target node t are added to the DAG that represents the precedence constraints; see Figure 2. Arcs from s to all operations with no predecessors and from all nodes without successor to t must also be added. In addition, dashed arcs represent the sequence (list) in which operations are processed by each machine. Figures at the top or bottom of the operations correspond to their processing times. In this directed graph, that we name $G = (V, A)$ from now on[1], the longest path from s to t corresponds to the makespan. In this example, the longest path, with value 80, corresponds to the path $s, 1, 2, 4, 5, 6, t$. The depicted feasible solution corresponds to an optimal solution for minimizing the makespan. Figure 3 shows the Gantt chart representation of the solution. Note that standard processing times were used, i.e. in this example the learning effect was not considered at all.

In the present work, we consider that when an operation i is assigned to a machine k and it is the rth operation to be processed by the machine, the standard processing time p_{ik} is affected by a learning effect and becomes $\psi_\alpha(p_{ik}, r)$. Following [6], we might consider $\psi_\alpha(p, r) := p/r^\alpha$. In this formulation, r^α is known as the learning rate, while $\alpha > 0$ is called the learning index. The larger the value of α, the faster the learning. The model is based on the premise that the time

[1]Note that $V = \mathcal{O} \cup \{s, t\}$ and A is composed by the given arcs in $\widehat{A}$ that represent the precedence relations among operations, plus the mentioned arcs related to the new nodes s and t, plus the mentioned machine arcs.

needed to complete an operation decreases as the number of repetitions increases. Additionally, it presumes that each machine consistently carries out the same type of operation. As noted in [6, §2], these assumptions are considered reasonable in many practical settings, especially in high-technology manufacturing environments. In the present work, a constraint programming model of the problem will be introduced and instances solved with a commercial solver that requires processing times to assume integer values only. Thus, we consider $\psi_\alpha(p, r) := \lfloor 100\, p/r^\alpha + 1/2 \rfloor$. Multiplying by one hundred, adding 0.5, and taking the floor corresponds to changing the unit of measures (from seconds to milliseconds, for example) and rounding to the closest integer value. The learning function and its learning index may differ from machine to machine, but for the sake of simplicity, we consider them to be the same for all machines. According to the classification in [3], our work falls under "position-based learning, which takes place independently on each machine." Figures 4 and 5 illustrate the graph representation and the Gantt chart of an optimal solution in which the learning function ψ_α with $\alpha = 0.5$ is considered. This value of α has illustrative purposes only. It is worth noting that a different schedule is found, whose makespan, given by the critical path $s, 7, 8, 4, 5, 6, t$, is equal to 50.16 in the original units of time (5016 in the new ones).

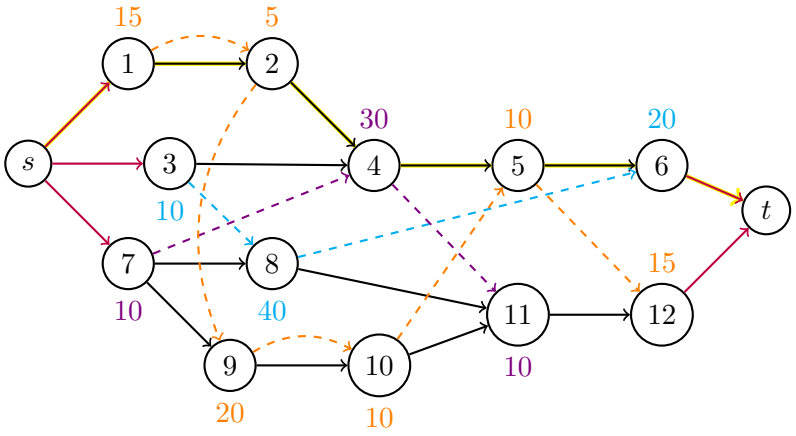


Figure 2: Representation of an optimal solution to the instance in Figure 1. Additional arcs leaving s and arriving at t are shown in purple. Blue dashed arcs correspond to machine 1, violet dashed arcs correspond to machine 2, and orange dashed arcs correspond to machine 3. The longest path, which corresponds to the makespan, is highlighted in yellow.

The rest of this paper is organized as follows. In Section 2, we introduce the mixed integer linear programming and constraint programming models. In Section 3, we outline the proposed constructive heuristics. In Section 4.1, we report the introduced instances. Numerical experiments with the constructive heuristics and exact commercial solvers are reported in Section 4. Conclusions and lines of future research are presented in the concluding section.

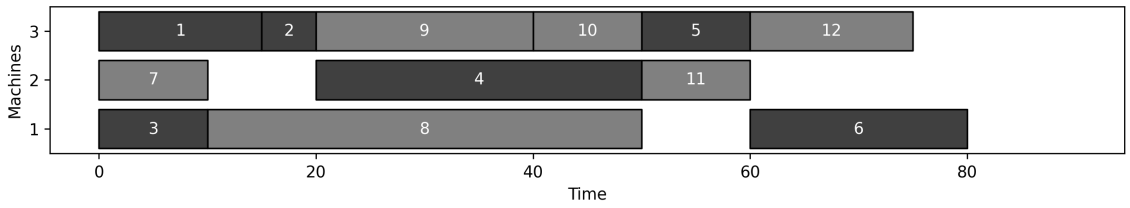


Figure 3: Gantt chart representation of the optimal solution shown in Figure 2 to the instance of Figure 1.

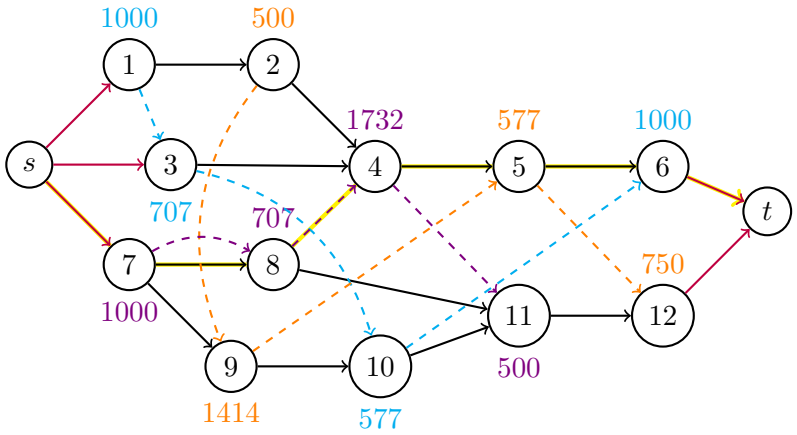


Figure 4: Representation of an optimal solution to the instance in Figure 1 in the presence of learning effect.

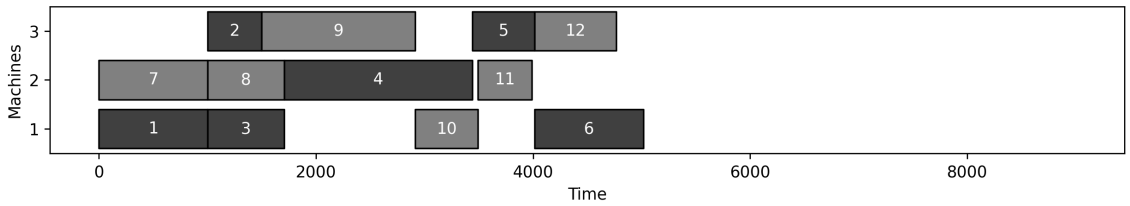


Figure 5: Gantt chart representation of the optimal solution shown in Figure 4 to the instance of Figure 1 in the presence of learning effect.

2 Mixed integer and constraint programming models

In this section, we present mixed-integer linear programming (MILP) and constraint programming (CP) formulations for the FJS scheduling problem with nonlinear routes and position-based learning effect.

2.1 Mixed-integer linear programming model

Position-based decision variables provide a fundamental framework for modeling position-based learning effects, since they allow processing times to depend on an operation's position in the processing sequence on a machine. If a different learning effect were considered, such as one based on the cumulative processing time of the operations already completed on a machine, alternative decision variables might be preferred or required. The proposed MILP model is derived from [4] and is built upon the model introduced in [26] that considers position-based decision variables; see also [27]. The sets, decision variables, and parameters used in the MILP model are described below.

Sets:

$\mathcal{O}$**:** set of operations,

$\mathcal{F}$**:** set of machines,

$\mathcal{O}_k$**:** set of operations that can be processed by machine k,

$\mathcal{F}_i$**:** set of machines that can process operation i,

$\widehat{A}$**:** set of directed arcs that represent operations' precedence constraints.

Parameters:

p_{ik}**:** standard processing time of operation i in machine k,

$\tilde{p}_{ikr}$**:** actual processing time of operation i if processed in the rth position of machine k, taking into account the effect of learning. The value of $\tilde{p}_{ikr}$ is given by $\tilde{p}_{ikr} = \psi_\alpha(p_{ik}, r)$, where, as already mentioned in the introduction, $\psi_\alpha(p, r) := \lfloor 100\, p/r^\alpha + 1/2 \rfloor$.

Decision variables:

x_{ikr}**:** 1 if operation i is the rth operation to be processed in machine k; 0, otherwise,

s_i**:** starting time of operation i,

$C_{\max}$**:** upper bound on the completion time of all operations which, when minimized, is equal to the makespan,

h_{kr}**:** starting time of the rth operation to be processed in machine k,

p'_i**:** actual processing time of operation i (considering the learning effect).

The MILP model follows.

$$\text{Minimize} \quad C_{\max} \tag{1}$$

$$\text{subject to}$$

$$\sum_{k \in \mathcal{F}_i} \sum_{r=1}^{|\mathcal{O}_k|} x_{ikr} = 1, \quad i \in \mathcal{O}, \tag{2}$$

$$\sum_{i \in \mathcal{O}_k} x_{ikr} \leq 1, \quad k \in \mathcal{F}, r = 1, \ldots, |\mathcal{O}_k|, \tag{3}$$

$$\sum_{i \in \mathcal{O}_k} x_{i,k,r+1} \leq \sum_{i \in \mathcal{O}_k} x_{ikr}, \quad k \in \mathcal{F}, r = 1, \ldots, |\mathcal{O}_k| - 1, \tag{4}$$

$$p'_i = \sum_{k \in \mathcal{F}_i} \sum_{r=1}^{|\mathcal{O}_k|} \tilde{p}_{ikr}\, x_{ikr}, \quad i \in \mathcal{O}, \tag{5}$$

$$h_{kr} + \sum_{i \in \mathcal{O}_k} \tilde{p}_{ikr}\, x_{ikr} \leq h_{k,r+1}, \quad k \in \mathcal{F}, r = 1, \ldots, |\mathcal{O}_k| - 1, \tag{6}$$

$$h_{kr} + \sum_{i \in \mathcal{O}_k} \tilde{p}_{ikr}\, x_{ikr} \leq C_{\max}, \quad k \in \mathcal{F}, r = |\mathcal{O}_k|, \tag{7}$$

$$s_i + p'_i \leq s_j, \quad \forall (i,j) \in \widehat{A}, \tag{8}$$

$$s_i + p'_i - \left(2 - x_{ikr} - \sum_{t=r+1}^{|\mathcal{O}_k|} x_{jkt}\right) M \leq s_j, \quad \begin{array}{c} \forall i,j \in \{\mathcal{O} \mid i \neq j\}, \forall k \in \mathcal{F}_i \cap \mathcal{F}_j, \\ r = 1, \ldots, |\mathcal{O}_k| - 1, \end{array} \tag{9}$$

$$h_{kr} - M\,(1 - x_{ikr}) \leq s_i, \quad i \in \mathcal{O}, k \in \mathcal{F}_i, r = 1, \ldots, |\mathcal{O}_k|, \tag{10}$$

$$s_i - M\,(1 - x_{ikr}) \leq h_{kr}, \quad i \in \mathcal{O}, k \in \mathcal{F}_i, r = 1, \ldots, |\mathcal{O}_k|, \tag{11}$$

$$s_i \geq 0, \quad i \in \mathcal{O}, \tag{12}$$

$$h_{kr} \geq 0, \quad k \in \mathcal{F}_i, r = 1, \ldots, |\mathcal{O}_k|, \tag{13}$$

$$x_{ikr} \in \{0,1\}, \quad i \in \mathcal{O}, k \in \mathcal{F}_i, r = 1, \ldots, |\mathcal{O}_k|. \tag{14}$$

Objective function (1) represents the minimization of the makespan. Constraints (2) state that each operation must be processed by exactly one machine and occupy exactly one position on that machine. Constraints (3) state that each possible position within the sequence of operations processed by a machine is occupied by at most one operation. Constraints (4) state that a machine position (other than the first position) can be occupied by an operation only if the previous position is also occupied by another operation. Constraints (5) define the actual processing time of each operation i. This quantity can then be used in constraints (8) and (9) allowing us to make the model formulation simpler and easier to understand. Constraints (6) avoid the overlapping of operations assigned to the same machine. Constraints (7) say that the makespan must be greater than or equal to the completion time of all operations. Combining these constraints with the minimization of the objective function (1) makes the makespan to coincide with the completion time of the last operation to be completed. Constraints (8) models precedence constraints between operations by saying that if an operation i precedes an

operation j then j cannot be started before i is completed. Constraints (9) state that, if two operations i and j were assigned to the same machine k and operation i precedes operation j, then j cannot be started before i is completed. If operation i is assigned to position r of machine k (i.e. if $x_{ikr} = 1$) then constraints (10) and (11) force h_{kr} (the starting time of rth operation to be processed by machine k) to be equal to s_i (the starting time of operation i). Constraints (12), (13), and (14) determine the decision variables' domains. M is a "sufficiently large" number and may assume the value $\sum_{i\in\mathcal{O}}\sum_{k\in\mathcal{F}} p_{ik}$.

2.2 Constraint Programming model

Constraint Programming (CP) is a potent methodology widely employed for solving scheduling problems in academic and industry settings. CP Optimizer [20], being an optimization commercial solver rooted in CP, incorporates specialized concepts to the definitions of constraints and variables. In this section, we introduce a CP model specifically designed for its utilization in connection with CP Optimizer. We will explain the CP Optimizer syntax as we walk through the model formulation later in this subsection. This model uses the same sets and parameters described in Section 2.1. It also uses an interval variable o_i for each operation $i \in \mathcal{O}$ and an optional interval variable a_{ikr} for each $i \in \mathcal{O}$, $k \in \mathcal{F}_i$ and $r \in \{1, \ldots, |\mathcal{O}_k|\}$ whose existence indicates that operation i is assigned to position r of machine k. (More on optional interval variables later.) The model follows below.

$$\text{Minimize} \quad \max_{i\in\mathcal{O}} \text{endOf}(o_i) \tag{15}$$

$$\text{subject to}$$

$$\text{endBeforeStart}(o_i, o_j), \quad (i,j) \in \widehat{A}, \tag{16}$$

$$\text{alternative}\left(o_i, [a_{ikr}]_{k\in\mathcal{F}_i, r=1,\ldots,|\mathcal{O}_k|}\right), \quad i \in \mathcal{O}, \tag{17}$$

$$\text{noOverlap}\left([a_{ikr}]_{i\in\mathcal{O}_k, r=1,\ldots,|\mathcal{O}_k|}\right), \quad k \in \mathcal{F}, \tag{18}$$

$$\text{endBeforeStart}(a_{ikr}, a_{jk,r+1}), \quad \begin{array}{c} i,j \in \mathcal{O},\ k \in \mathcal{F}_i \cap \mathcal{F}_j, \\ r = 1, \ldots, |\mathcal{O}_k| - 1, \end{array} \tag{19}$$

$$\sum_{i\in\mathcal{O}_k} \text{presenceOf}(a_{i,k,r+1}) \leq \sum_{i\in\mathcal{O}_k} \text{presenceOf}(a_{i,k,r}), \quad k \in \mathcal{F}, r = 1, \ldots, |\mathcal{O}_k| - 1, \tag{20}$$

$$\text{interval } o_i, \quad i \in \mathcal{O}, \tag{21}$$

$$\text{interval } a_{ikr}, \text{opt}, \text{size} = \tilde{p}_{ikr}, \quad i \in \mathcal{O}, k \in \mathcal{F}_i, r = 1, \ldots, |\mathcal{O}_k|. \tag{22}$$

Interval decision variables of the problem are described in (21) and (22). In (21), an interval variable o_i for each operation i is defined. In (22), an *optional* interval variable a_{ikr} is defined for each possible assignment of operation i to a machine $k \in \mathcal{F}_i$ at positions $r = 1, \ldots, |\mathcal{O}_k|$. *Optional* means that the interval variable may exist or not; and the remaining of the constraint says that, in case it exists, its size must be given by $\tilde{p}_{ikr}$. Constraints (17) state that each operation i must be allocated to exactly one machine $k \in \mathcal{F}_i$ in exactly one position r. That is, one and only one interval variable a_{ikr} must be present and the selected interval a_{ikr} must start and end at the same instants as interval o_i. The objective function (15) minimizes the makespan, given by the maximum end value of all the operations represented by the interval variables o_i. Precedence constraints between operations are posted as endBeforeStart constraints between

interval variables in constraints (16). Constraints (18) state that (a) for each machine k, at most one operation can be assigned to each position and (b) operations assigned to different positions cannot overlap. Constraints (19) say that, for any machine and any position, the operation assigned to that position must finish before the operation in the next position starts. Constraints (20) force that empty positions in machine k are all grouped together at the end of the list of operations. In other words, that the operations that machine k will actually process are all grouped together at the beginning of the list, with no empty positions among them.

3 Constructive Heuristics

In this section, we propose two constructive heuristics for the FJS scheduling problem with nonlinear routes and a position-based learning effect. Constructive heuristics build a feasible solution from scratch by iteratively selecting and sequencing one operation at a time. Both heuristics are based on classical dispatching rules widely used in scheduling: the earliest starting time (EST) rule [4] and the earliest completion time (ECT) rule [21]. In both cases, decisions are made by selecting pairs of operations and machines, taking into account precedence constraints, machine availability, and the learning effect. The goal is to use these heuristics to provide an initial feasible solution to the solver for a set of test instances.

The heuristic based on the EST rule gives priority to operation–machine pairs that can start processing as early as possible. At each iteration, among all unscheduled operations whose predecessors have already been scheduled, the algorithm determines the earliest instant at which any operation–machine pair can start processing. All pairs that can start at this earliest instant are considered, and a tie-breaking rule based on the shortest actual processing time is applied, taking into account the learning effect. In Appendix A we describe the heuristic in detail and calculate its complexity.

The heuristic based on the ECT rule follows the same structure as the EST-based method, differing only in the selection criterion. Instead of restricting attention to pairs that can start at the earliest possible time, the ECT-based heuristic selects the operation–machine pair that minimizes the completion time, i.e., the sum of the earliest feasible start time and the actual processing time of the operation. See Appendix A for details on the heuristic and the calculation of its complexity.

The EST-based heuristic gives priority to those operation–machine pairs that can start the earliest. At the beginning of the construction, this corresponds, roughly, to giving priority to all the first operations of each job, which are operations that have no precedents (operations 1, 3 and 7 in the example of Figure 1). Still, due to the intention of scheduling operations as early as possible, preference is given to empty machines, building solutions that use several machines. Prioritizing the initial operations of each job results in more operations having their predecessors scheduled. This increases the number of available options in future steps of the method. On the other hand, the heuristic based on the ECT rule chooses the operation–machine pairs that terminate the earliest, regardless of whether they are the ones that can start the earliest or not. Such a strategy can limit the number of operation–machine pairs available in future iterations, reducing the search space of the method. Moreover, the choice for the operation–machine pair that can finish earliest, combined with the learning effect, leads the method to schedule

operations on machines that already have several operations assigned to them, since the higher the position in the machine, the shortest the processing time (reduced by the positioned-based learning effect). This may lead to the construction of solutions in which not all machines are used. Depending on the chosen learning index α and the density of the DAG of precedences of the instance at hand, one heuristic may be better than the other.

The complete algorithmic descriptions, including pseudocode, auxiliary procedures, and computational complexity analyses, are provided in Appendix A.

4 Numerical experiments

In this section we present numerical experiments. All efforts are to build a set of test instances with optimal solutions. The models and constructive heuristics presented in this paper are used solely to build a set of test instances with optimal solutions; they are not intended to provide a solution method per se. First, we describe the set of instances considered. Second, we evaluate the two introduced constructive heuristics. Third, we assess the correctness of the MILP and CP models and attempt to infer which of the two, or rather which of the exact commercial solvers applied to each of them, is more effective in finding optimal solutions with a zero-gap certificate. Finally, we determine the usefulness of providing a feasible solution to the exact solvers.

The experiments were carried out in an Intel i9-12900K (12th Gen) processor operating at 5.200GHz and 128 GB of RAM. The constructive heuristics were implemented in C++. Models were solved using IBM ILOG CPLEX Optimization Studio version 22.1.2, that includes IBM ILOG CP Optimizer, with concert library and C++. The code was compiled using g++ 10.2.1. Benchmark instances and code are available at `https://github.com/kennedy94/FJS`.

4.1 Benchmark instances

Tractability of the models and performance of the constructive heuristics that we have introduced will be evaluated with the 50 large-scale instances proposed in [4], that were introduced for the FJS scheduling problem with sequence flexibility but without learning effect. In addition to these large-scale instances, a new set with 60 smaller instances, using the generator described in [4], was generated. Instances whose name starts with "Y" correspond to instances in which DAGs that represent the operations' precedences are Y-shaped (like the DAG in the top of Figure 1); while instances whose name starts with "DA" correspond to instances in which DAGs that represent the operations precedences are arbitrary DAGs (like the DAG in the bottom of Figure 1). The former will be called instances of Y-type and the latter will be called instances of DA-type from now on.

For a given instance, we define measures ω_1 and ω_2 of the routes' nonlinearity and the routing flexibility, respectively. Both measures are between 0 and 1 and, the larger their values, the larger the nonlinearity or flexibility they represent. Moreover, the larger the flexibility, the larger the search space and, in consequence, the harder the instance. Let $D^+ = (\mathcal{O}, \widehat{A}^+)$ be the transitive closure of the precedences DAG $D = (\mathcal{O}, \widehat{A})$. Let n_κ denote the number of operations in job κ. These operations form a connected subgraph of D^+. Let a_κ be the number of arcs in this subgraph. Since the minimum and maximum numbers of arcs in a connected DAG are given by $a^{\min}(n_\kappa) = n_\kappa - 1$ and $a^{\max}(n_\kappa) = n_\kappa(n_\kappa - 1)/2$, we have that $a_\kappa^{\min} \leq a_\kappa \leq a_\kappa^{\max}$.

Therefore

$$\omega_1^\kappa := 1 - \frac{a_\kappa - a^{\min}(n_\kappa)}{a^{\max}(n_\kappa) - a^{\min}(n_\kappa)},$$

is such that $\omega_1^\kappa \in [0,1]$ represents the routes' nonlinearity of the operations of job κ. The arithmetic mean among all n jobs of the instance, given by

$$\omega_1 = \frac{1}{n}\sum_{\kappa=1}^{n} \omega_1^\kappa,$$

is also between 0 and 1 and represent the degree of the instance routes' nonlinearity. The larger ω_1, the larger the search space and, therefore, the more difficult the instance. (Of course, any other type of average could be used in the definition of ω_1.)

In a similar way, we define the routing flexibility measure ω_2. It is easy to see that $|\mathcal{O}| \leq \sum_{i\in\mathcal{O}} |\mathcal{F}_i| = \sum_{k\in\mathcal{F}} |\mathcal{O}_k| \leq |\mathcal{O}||\mathcal{F}|$. Therefore,

$$\omega_2 = \frac{\sum_{i\in\mathcal{O}} |\mathcal{F}_i| - |\mathcal{O}|}{|\mathcal{O}||\mathcal{F}| - |\mathcal{O}|},$$

is between 0 and 1. The closer to 1, the larger the search space and, therefore, the harder the instance.

Tables B1 and B2 in Appendix B provide a comprehensive overview of the instances main characteristics. In the small-scale instances of Table B1, the number of binary variables of the MILP models and the number of interval variables of the CP models go up to almost 1,000; while in both models the number of constraints goes up to 13,000. On the other hand, in the large-scale instances of Table B2, the number of binary variables of the MILP models and the number of interval variables of the CP models go up to almost 73,000; while in both models the number of constraints goes up to 4,000,000. Moreover, for each instance, the number of binary variables in its MILP model is very similar to the number of interval variables in its CP model and the two models also have a very similar number of constraints. It is worth noting that in the small-scale instances there is no clear differentiation between the routes' nonlinearity sparsity measure ω_1 of the DA-type and the Y-type instances (see Table B1). On the other hand, in the large-scale instances, Y-type instances have a value of ω_1 clearly lower than the value of ω_1 of DA-type instances.

In all numerical experiments from here on, we will consider the 110 instances (60 small-scale plus 50 large-scale) with the learning index $\alpha \in \{0.1, 0.2, 0.3\}$ for a total of 330 instances. The value $\alpha \approx 0.3$ is used in [6]; this paper introduces, in the scheduling context, the learning model we study in this paper. Furthermore, values of α in $(0, 0.3]$ correspond to typical values in the online print shop scheduling problem (OPS).

4.2 Experiments with the constructive heuristics

In this section, we apply the two constructive heuristics EST and ECT to the 330 instances. In all instances, the constructive heuristics take less than 0.001 seconds of CPU time to build a solution. Tables B3 and B4 in Appendix B show the makespan of the solution computed by each heuristic for each instance, whereas Table 1 reports, for each heuristic, the number of wins

and the average makespan across the different benchmark groups. Considering the small-scale instances, depending on the instance, there may be a significant difference between the solutions found by one and the other constructive heuristic (see Table B3). However, on average, as shown in Table 1, the two heuristics behave basically indistinguishably. For the large-scale instances the comparison is a bit different: there is a clear advantage of the EST constructive heuristic in the DA-type instances, while on the other hand there is a clear advantage of the ECT constructive heuristic in the Y-type instances. This behavior is independent of the learning index α. As mentioned at the end of Section 3, the ECT greedy strategy of choosing the operation-machine pair that terminates first seems to compensate in situations where there is little nonlinearity in the routes, such as the Y-type instances. This is because the greedy choice does not significantly reduce the search space.

instance set			$\alpha = 0.1$		$\alpha = 0.2$		$\alpha = 0.3$	
			EST	ECT	EST	ECT	EST	ECT
small-scale	DA-type	wins	19	13	16	16	16	16
		mean	24,621.93	25,325.67	23,344.07	23,265.40	22,108.63	21,749.73
	Y-type	wins	17	13	12	19	12	19
		mean	44,366.87	45,513.93	40,872.83	40,661.40	37,785.17	37,628.37
large-scale	DA-type	wins	20	10	24	6	28	2
		mean	65,249.50	67,439.80	54,310.83	57,746.00	45,578.87	48,181.97
	Y-type	wins	5	15	7	13	7	13
		mean	87,865.80	80,338.80	74,373.85	68,682.05	65,850.25	59,452.35

Table 1: Performance of the constructive heuristics on the testbed instances: average makespan and number of wins.

4.3 Solving the proposed models with a commercial solver

In this section, we apply the two exact solvers, CPLEX and CP Optimizer, to the 330 instances and assess the quality of the solutions obtained within a given CPU time limit. We consider two settings: one in which the solution produced by the constructive heuristics is provided as an initial solution, and one in which no initial solution is supplied. Since there is no clear winner between the two constructive heuristics and their execution time is negligible, we run both and give the best of the two as initial solution to the exact solver. In the tables and figures, the solvers' runs that receive as input a solution computed by one of the constructive heuristics are identified with the expression "warm start".

A solution is reported as optimal by the solvers when either

$$\text{absolute gap} = \text{best feasible solution} - \text{best lower bound} \leq \epsilon_{\text{abs}} \tag{23}$$

or

$$\text{relative gap} = \frac{|\text{best feasible solution} - \text{best lower bound}|}{10^{-10} + |\text{best feasible solution}|} \leq \epsilon_{\text{rel}}, \tag{24}$$

where, by default, $\epsilon_{\text{abs}} = 10^{-6}$ and $\epsilon_{\text{rel}} = 10^{-4}$, and "best feasible solution" means the smallest value of the makespan related to a feasible solution generated by the method. Since the optimal

makespan of the instances considered in this work always assumes an integer value, we choose to use $\epsilon_{\text{abs}} = 1 - 10^{-6}$ and $\epsilon_{\text{rel}} = 0$. The choice $\epsilon_{\text{rel}} = 0$ avoids premature stops at a solution that may not be optimal. The choice $\epsilon_{\text{abs}} = 1 - 10^{-6}$ allows to stop early when a relative gap smaller than 1 clearly indicates that the optimal solution has already been found. A CPU time limit of 1 hour was imposed. All other solvers parameters were set to their default values.

Tables B5, B6, B7, B8, B9, and B10 in Appendix B show the results in detail when the warm start is not taken into account. Tables B5, B6, and B7 correspond to the small-scale instances with α equal to 0.1, 0.2, and 0.3, respectively; while Tables B8, B9, and B10 show the same thing for the large-scale instances. Tables B11–B16 in Appendix B show the results when the warm start is taken into account. The main results for the different benchmark groups are summarized in Table 2. In the table, "feas" denotes the number of instances for which each solver found a feasible solution, whereas "opt" denotes the number of instances for which each solver found an optimal solution. The results show that, for the small-scale instances, both solvers always found a feasible solution. For the large-scale instances, CPLEX without a warm start failed to find feasible solutions in approximately half of the instances. The results also show that, for both solvers, the number of optimal solutions found decreases as the value of α increases. For the small-scale instances, the solvers found between 53 and 60 optimal solutions out of 60, whereas for the large-scale instances they found at most four optimal solutions, out of 50, depending on the value of α. Consider the subset of instances for which both CPLEX and CP found an optimal solution for all three values of α. The row "$\overline{\text{time}}$ (s)" reports the average CPU time over this subset[2]. The averages indicate that the computational effort increases with α and that, on average, CPLEX requires more time than CP. The row "$\overline{\text{gap}}$ (%)" reports the average optimality gap over the complementary subset, that is, the set of instances for which at least one of the solvers failed to find an optimal solution for some value of α. The results show that the average gap increases with α.

Let us now examine each benchmark group in detail, starting from the 180 small-scale instances. Without a warm start (see Tables B5, B6, and B7), there are 171 instances for which both CPLEX and CP found an optimal solution. In two instances, only CPLEX found an optimal solution, while in two others, only CP Optimizer did so. In five instances (four of them with $\alpha = 0.3$), neither solver was able to find an optimal solution. In these five instances, the makespan of the best feasible solution found by CP Optimizer was lower in four cases and equal in one case compared to CPLEX. In the four cases where CP Optimizer performed better, the makespan was, on average, 3.91% lower. However, CP Optimizer's best bounds (lower bounds) were always smaller, resulting in larger optimality gaps. Of the 171 instances in which both solvers found optimal solutions, CP Optimizer was faster than CPLEX in 158 instances, whereas the opposite occurred in 13 instances. When using a warm start (see Tables B11, B12, and B13), the qualitative results remain virtually unchanged. Both solvers remained unable to close the optimality gap for the same five instances. CPLEX managed to solve two additional instances for the MILP model; surprisingly, CP Optimizer solved two fewer instances for the CP model.

We now consider the 173 instances for which CPLEX found an optimal solution both without

[2]For the large-scale instances, this subset comprises only two instances without a warm start and only one instance with a warm start. Consequently, we report average CPU times for small-scale instances only.

	Warm start		$\alpha = 0.1$		$\alpha = 0.2$		$\alpha = 0.3$	
			CPLEX	CP	CPLEX	CP	CPLEX	CP
small-scale	no	feas	60	60	60	60	60	60
		opt	60	60	58	58	55	55
		$\overline{\text{gap}}(\%)$	–	–	11.66	36.12	14.16	42.60
		$\overline{\text{time}}(s)$	70.51	11.81	218.84	60.82	396.07	257.51
	yes	feas	60	60	60	60	60	60
		opt	60	60	59	58	56	53
		$\overline{\text{gap}}(\%)$	–	–	4.95	36.12	12.01	43.00
		$\overline{\text{time}}(s)$	28.68	7.45	67.80	36.23	153.32	103.56
large-scale	no	feas	27	50	27	50	18	50
		opt	3	4	3	3	2	2
		$\overline{\text{gap}}(\%)$	62.61	38.62	63.90	51.30	72.38	60.38
	yes	feas	50	50	50	50	50	50
		opt	3	4	2	3	2	1
		$\overline{\text{gap}}(\%)$	40.10	36.75	43.74	48.76	50.84	57.08

Table 2: Summary of the main results across the different benchmark groups, focusing on the effect of the learning index α.

and with a warm start. In these cases, the warm start made CPLEX, on average, 54.35% faster. For the 171 instances in which CP Optimizer found an optimal solution both without and with a warm start, the warm start made it, on average, 7.26% slower. Considering the 171 instances in which both CPLEX and CP Optimizer found an optimal solution with a warm start, CPLEX was faster in 79 cases, whereas CP Optimizer was faster in 92 cases. It is well known that CPLEX can significantly benefit from a feasible solution provided as a warm start because it employs branch-and-bound and cutting planes. This solution can be used to set an upper bound, helping to prune the search tree and guide the search toward optimality. In contrast, CP Optimizer relies on constraint propagation and domain reduction. Although it can start from an initial solution, it does not exploit it to tighten bounds or accelerate convergence in the same manner. Our experimental results support these observations.

We conclude our analysis of the small-scale instances by examining the makespan as a function of the learning index α. For this, we used the solutions obtained from CPLEX without warm starts; analyses with other data—CPLEX with warm starts, CP Optimizer with or without warm start—produced virtually identical results. We also include the makespan when $\alpha = 0$, representing the scenario with no learning effect. The top of Figure 6 shows the makespan for each of the 60 instances with no learning and with $\alpha \in \{0.1, 0.2, 0.3\}$. As expected, higher values of α correspond to lower makespans. Y-type instances, which have less nonlinearity in the routes, tend to exhibit larger makespans. The bottom of the figure shows the relative gap, $[C_{\max} - C_{\max}(\alpha)]/C_{\max}$, between the makespan without learning, $C_{\max}$, and with learning, $C_{\max}(\alpha)$. This measures the proportional reduction in makespan due to learning. On average, the reductions are 6.55% for $\alpha = 0.1$, 12.65% for $\alpha = 0.2$, and 18.21% for $\alpha = 0.3$, with no noticeable dependence on instance type (DA or Y).

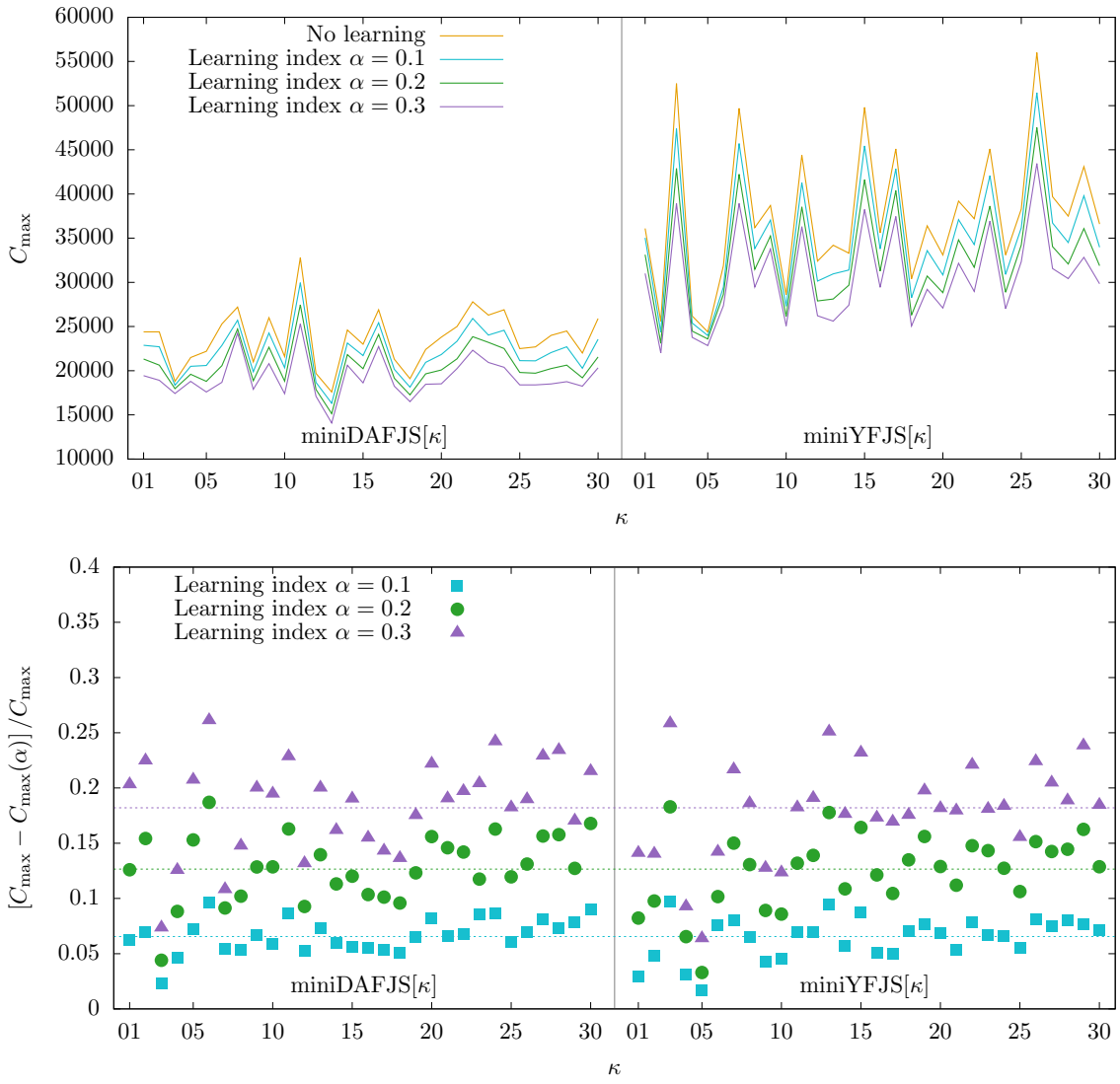


Figure 6: Effect of the learning index α on the makespan of the 60 small-scale instances. Top: makespan values for each instance, including the baseline scenario with no learning ($\alpha = 0$). Bottom: relative reduction in makespan, showing the proportional impact of learning; dashed lines indicate the average reduction for each value of α.

The analysis of the 150 large-scale instances is slightly different. Without a warm start (see Tables B8, B9, and B10), there are eight instances for which both CPLEX and CP found an optimal solution. In one instance, only CP Optimizer found an optimal solution, whereas CPLEX did not do so in any instance. In 141 instances, neither solver was able to find an optimal solution. CP Optimizer was faster in seven of the eight instances in which both solvers found optimal solutions, while CPLEX was faster in the remaining instance. CPLEX found

at least one feasible solution in 85 instances; in the remaining 65, it reached the time limit without calculating a single feasible solution. CP Optimizer found at least one feasible solution in all 150 instances. In all situations where both CPLEX and the CP Optimizer found feasible solutions, the makespan of the solution found by the CP Optimizer was strictly lower (setting aside the eight instances where CPLEX found an optimal solution). On average, the feasible solutions found by CP Optimizer were 50.46% better than those found by CPLEX. Next, we compare the feasible solutions found by the exact solvers with the best solutions obtained by the constructive heuristics EST and ECT. For the following analysis, we consider separately the 85 instances in which CPLEX found a feasible solution and the 65 instances in which it did not. First, let us consider the former. CPLEX's solution is better than the constructive heuristics' solution in 24 instances and worse in 61 instances. The CP Optimizer solution is better than the constructive heuristics' solution in 84 instances and equal in one instance. Thus, in this first set, CP Optimizer outperforms the constructive heuristics, while CPLEX does not. Now, let us consider the second set, consisting of the 65 instances in which CPLEX failed to compute a feasible solution. In this set, the solutions computed by CP Optimizer are better than those of the constructive heuristics in 35 instances and worse in the remaining 30. In nearly half of these cases, CP Optimizer produces feasible solutions within the time limit that are nonetheless inferior to those obtained by the constructive heuristics. These results suggest that the instances in this second set are particularly challenging for the exact solvers. They motivated us to consider using the solutions of the constructive heuristics as an initial solution for the exact methods.

Providing the constructive heuristics' solution to the exact solvers results in one fewer optimal solution for both the MILP and CP models. The question remains as to whether the exact methods are able to improve the solution provided by the constructive heuristics. In the case of the MILP model instances, the initial solution is improved in 42 instances, while in the CP model instances the initial solution is improved in 145 instances. The average improvement is 12.64% in the first case and 13.71% in the second. If we compare the makespan of the feasible solutions found using CPLEX and the CP Optimizer, the solutions found by CPLEX are better in only one instance, worse in 135, and equal in 14. In the instance where the solution found by CPLEX is better than the solution found by the CP Optimizer, the improvement is 0.68%. In the 135 instances where the solution found by the CP Optimizer is better, the average improvement is 10.84%. In short, it is challenging to find an optimal solution for large-scale instances and CP models provide better quality feasible solutions when it is not possible to find an optimal solution. One way or another, whether using the MILP model or the CP model, with or without warm start, it was possible to find optimal solutions in only nine large-scale instances and feasible solutions in all the remaining 141 large-scale instances.

5 Conclusions

In this work, we addressed the FJS scheduling problem with nonlinear routes and position-based learning effect. To the authors' knowledge, this combination, with potential application in a wide range of real-world industrial environments, has never been addressed in the literature. As a first step towards its efficient and effective solution, we introduced a set of 110 instances that

become 330 instances by varying the learning index $\alpha \in \{0.1, 0.2, 0.3\}$. By introducing MILP and CP models, an exact solver aided by constructive heuristics was able to provide 183 optimal solutions. Instances, their solutions, models and constructive heuristics are all available for download at `https://github.com/kennedy94/FJS`. We hope that this benchmark test-set will guide the introduction and evaluation of new effective heuristic and metaheuristic approaches for the FJS scheduling problem with nonlinear routes and position-based learning. In fact, this is the current line of research of the authors.

Acknowledgment: We are grateful to the two anonymous referees for their careful reading and insightful comments, which significantly improved this manuscript. We also sincerely thank Marcelo Marazzi for his careful editing and helpful suggestions that enhanced the quality of the written text in English.

Compliance with ethical standards

Funding: This work was founded by the Brazilian agencies FAPESP (grants 2013/07375-0, 2022/05803-3, and 2023/08706-1) and CNPq (grants 311536/2020-4 and 302073/2022-1).

Disclosure of potential conflicts of interest: All authors declare that they have no conflicts of interest.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

Author contributions: All authors contributed equally in all phases of the development of this work.

Data availability: The datasets generated during and/or analysed during the current study are available in the GitHub repository, `https://github.com/kennedy94/FJS`.

References

[1] R. Alvarez-Valdes, A. Fuertes, J. M. Tamarit, G. Giménez, and R. Ramos. A heuristic to schedule flexible job-shop in a glass factory. *European Journal of Operational Research*, 165(2):525–534, 2005.

[2] J. L. Andrade-Pineda, D. Canca, P. L. Gonzalez-R, and M. Calle. Scheduling a dual-resource flexible job shop with makespan and due date-related criteria. *Annals of Operations Research*, 291(1):5–35, 2020.

[3] A. Azzouz, M. Ennigrou, and L. Ben Said. Scheduling problems under learning effects: classification and cartography. *International Journal of Production Research*, 56(4):1642–1661, 2018.

[4] E. G. Birgin, P. Feofiloff, C. G. Fernandes, E. L. de Melo, M. T. I. Oshiro, and D. P. Ronconi. A MILP model for an extended version of the flexible job shop problem. *Optimization Letters*, 8(4):1417–1431, 2014.

[5] E. G. Birgin, J. E. Ferreira, and D. P. Ronconi. List scheduling and beam search methods for the flexible job shop scheduling problem with sequencing flexibility. *European Journal of Operational Research*, 247(2):421–440, 2015.

[6] D. Biskup. Single-machine scheduling with learning considerations. *European Journal of Operational Research*, 115(1):173–178, 1999.

[7] D. Biskup. A state-of-the-art review on scheduling with learning effects. *European Journal of Operational Research*, 188(2):315–329, 2008.

[8] Z.-C. Cao, C.-R. Lin, and M.-C. Zhou. A knowledge-based cuckoo search algorithm to schedule a flexible job shop with sequencing flexibility. *IEEE Transactions on Automation Science and Engineering*, 18(1):56–69, 2021.

[9] T. C. E. Cheng and G. Wang. Single machine scheduling with learning effect considerations. *Annals of Operations Research*, 98(1/4):273–290, 2000.

[10] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, Cambridge, MA, USA, 4th edition, 2022.

[11] S. Dauzère-Pérès, J. Ding, L. Shen, and K. Tamssaouet. The flexible job shop scheduling problem: A review. *European Journal of Operational Research*, 314(2):409–432, 2024.

[12] P. Y. Gan and K. S. Lee. Scheduling of flexible-sequenced process plans in a mould manufacturing shop. *The International Journal of Advanced Manufacturing Technology*, 20(3):214–222, 2002.

[13] J. Gao, M. Gen, and L. Sun. Scheduling jobs and maintenances in flexible job shop with a hybrid genetic algorithm. *Journal of Intelligent Manufacturing*, 17(4):493–507, 2006.

[14] M. R. Garey, D. S. Johnson, and R. Sethi. The complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research*, 1(2):117–129, 1976.

[15] J. N. D. Gupta and S. K. Gupta. Single facility scheduling with nonlinear processing times. *Computers and Industrial Engineering*, 14(4):387–393, 1988.

[16] A. Janiak, T. Krysiak, and R. Trela. Scheduling problems with learning and ageing effects: A survey. *Decision Making in Manufacturing and Services*, 5(1):19–36, 2011.

[17] G. A. Kasapidis, S. Dauzère-Pérès, D. C. Paraskevopoulos, P. P. Repoussis, and C. D. Tarantilis. On the multiresource flexible job-shop scheduling problem with arbitrary precedence graphs. *Production and Operations Management*, 32(7):2322–2330, 2023.

[18] G. A. Kasapidis, D. C. Paraskevopoulos, P. P. Repoussis, and C. D. Tarantilis. Flexible job shop scheduling problems with arbitrary precedence graphs. *Production and Operations Management*, 30(11):4044–4068, 2021.

[19] Y. K. Kim, K. Park, and J. Ko. A symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling. *Computers and Operations Research*, 30(8):1151–1171, 2003.

[20] P. Laborie, J. Rogerie, P. Shaw, and P. Vilím. IBM ILOG CP optimizer for scheduling: 20+ years of scheduling with constraints at IBM/ILOG. *Constraints*, 23(2):210–250, 2018.

[21] J. Y.-T. Leung, H. Li, and M. Pinedo. Order scheduling in an environment with dedicated resources in parallel. *Journal of Scheduling*, 8(5):355–386, 2005.

[22] W. T. Lunardi, E. G. Birgin, P. Laborie, D. P. Ronconi, and H. Voos. Mixed integer linear programming and constraint programming models for the online printing shop scheduling problem. *Computers and Operations Research*, 123:105020, 2020.

[23] W. T. Lunardi, E. G. Birgin, D. P. Ronconi, and H. Voos. Metaheuristics for the online printing shop scheduling problem. *European Journal of Operational Research*, 293(2):419–441, 2021.

[24] G. Vilcot and J.-C. Billaut. A tabu search and a genetic algorithm for solving a bicriteria general job shop scheduling problem. *European Journal of Operational Research*, 190(2):398–411, 2008.

[25] A. Vital-Soto, A. Azab, and M. F. Baki. Mathematical modeling and a hybridized bacterial foraging optimization algorithm for the flexible job-shop scheduling problem with sequencing flexibility. *Journal of Manufacturing Systems*, 54:74–93, 2020.

[26] H. M. Wagner. An integer linear-programming model for machine scheduling. *Naval Research Logistics Quarterly*, 6(2):131–140, 1959.

[27] J. M. Wilson. Alternative formulations of a flow-shop scheduling problem. *Journal of the Operational Research Society*, 40(4):395–399, 1989.

[28] L. Yu, C. Zhu, J. Shi, and W. Zhang. An extended flexible job shop scheduling model for flight deck scheduling with priority, parallel operations, and sequence flexibility. *Scientific Programming*, 2017:1–15, 2017.

Appendix A: Technical details of the constructive heuristics

This appendix presents the complete technical description of the constructive heuristics introduced in Section 3. It includes the formal notation, detailed pseudocode, auxiliary algorithms, and computational complexity analyses required for full reproducibility.

Algorithm 1 describes the constructive heuristic based on EST rule. The algorithm outputs a complete feasible schedule represented by several interconnected structures. It returns an assignment function f, indicating the machine selected for each operation, and a vector w of actual processing times, computed according to the position-based learning effect. For each machine $k \in \mathcal{F}$, the algorithm produces an ordered list Q_k defining the sequence of operations

processed on that machine. These decisions are encoded in a solution graph $G = (V, A)$, where $V = \mathcal{O} \cup \{s, t\}$ and A contains both the original precedence arcs and the machine sequencing arcs induced by the constructed schedule. Based on this graph, the algorithm computes a critical path $\mathcal{P}$ from s to t, whose length corresponds to the makespan $C_{\max}$. In addition, it returns a topological order $\mathcal{U}$ of G and a vector τ, where each element τ_k identifies the position in Q_k of the last operation on machine k that belongs to the critical path.

In the algorithm, r_v^{op} refers to the ready time of operation v, w_v refers to its actual processing time and c_v to its completion time. On the side of the machines, r_k^{mac} represents the instant in which machine k is released and g_k represents its first free position, which is the one that would be occupied if an operation were assigned to it (both quantities refer to the partial scheduling being constructed). f_v will indicate to which machine operation v was assigned and each machine k will have an ordered list Q_k with the sequence of operations to be processed. After the initializations (lines 2 to 5) comes the main loop, which is executed as long as there are still unscheduled operations. Among the not scheduled ones, the ready time is calculated for all those that already have all the preceding operations scheduled (lines 7 to 9). In line 10, observing the ready times of the operations and machines, the smallest instant $r_{\min}$ in which an operation could be scheduled is calculated and the set E of operation/machine pairs that could start at that instant $r_{\min}$ is constructed. As it was observed in [4], $|E|$ can be quite large and experience shows that a tie-breaking rule can significantly improve the method's performance. Thus, in line 11, among all the operation/machine pairs in E, taking into account the learning effect, the pair $(\hat{v}, \hat{k})$ with the shortest processing time is chosen. In line 12, $w_{\hat{v}}$, $f_{\hat{v}}$, and $c_{\hat{v}}$ are defined and the ready time $r_{\hat{k}}^{\text{mac}}$ and the free position $g_{\hat{k}}$ of machine $\hat{k}$ are updated. In lines 13 and 14 the corresponding machine arc is inserted in the graph G (the arc must not be inserted if operation $\hat{v}$ is the first one of machine $\hat{k}$). Finally, the list of operations assigned to machine $\hat{k}$ is updated and the scheduled operation is removed from the set of operations not yet scheduled. After all the operations have been scheduled, the critical path in G is calculated (line 16) to determine the makespan value $C_{\max}$. This is done with Algorithm 2.

Initializations in lines 2 to 5 of Algorithm 1 have complexity $O(|\mathcal{O}| + |\widehat{\mathcal{A}}| + |\mathcal{F}|)$. Within the main loop (lines 6 to 15), lines 7–9 have complexity $O(|\mathcal{A}|) = O(|\widehat{A}| + |\mathcal{O}| + |\mathcal{F}|)$, lines 10–11 have complexity $O(|\mathcal{O}| + \sum_{i \in \mathcal{O}} |F_i|)$, and line 12 has complexity $O(|\mathcal{O}|)$. Since the main loop is executed $|\mathcal{O}|$ times, its total complexity is $O(|\mathcal{O}|(|\widehat{A}| + |\mathcal{F}| + \sum_{i \in \mathcal{O}} |\mathcal{F}_i|))$. As the complexity of the main loop is larger that the complexity of the initialization as well as the complexity of line 16 (see below), then the complexity of Algorithm 1 is given by the complexity of its main loop. It is important to note that $\gamma = \sum_{i \in \mathcal{O}} |\mathcal{F}_i|$ is between $|\mathcal{O}|$ and $|\mathcal{O}||\mathcal{F}|$, but we prefer to keep the complexity expressed as a function of γ because γ is a measure of the size of the input that depends on the routes' nonlinearity of the instance under consideration. It is also important to note that the complexity of the algorithm depends on the routing flexibility of the operations, i.e., it depends on the number of dependency relations in $\widehat{A}$. Therefore, it is important to represent an instance in such a way that $\widehat{A}$ corresponds to a transitive reduction of the precedences' digraph.

The critical path in the directed graph $G = (V, A)$ can be computed with an adaptation [10, §22.2] of the Bellman-Ford algorithm; see Algorithm 2. In addition to the critical path $\mathcal{P}$, the algorithm returns a topological order $\mathcal{U}$ of the vertices of G and a vector τ of dimension $|\mathcal{F}|$.

Algorithm 1: Computes a solution graph $G = (V, A)$, f, Q, and w by using EST dispatching rule. Then, in G, it computes the largest path $\mathcal{P}$ from s to t and its length $C_{\max}$.

Input: $\mathcal{O}$, $\mathcal{F}$, p, $\widehat{A}$
Output: f, w, Q, $G = (V, A)$, $\mathcal{U}$, $\mathcal{P}$, $C_{\max}$, τ

function EST($\mathcal{O}$, $\mathcal{F}$, p, $\widehat{A}$, f, w, Q, G, $\mathcal{U}$, $\mathcal{P}$, $C_{\max}$, τ)
  Set $A \leftarrow \widehat{A} \cup \{(s, j) \mid (\cdot, j) \notin \widehat{A}\} \cup \{(i, t) \mid (i, \cdot) \notin \widehat{A}\}$ and define $V := \mathcal{O} \cup \{s, t\}$ and $G = (V, A)$.
  Set $r_v^{\text{op}} \leftarrow +\infty$ for all $v \in V$ and define $r_s^{\text{op}} := 0$, $w_s := w_t := 0$, and $c_s := 0$.
  Set $r_k^{\text{mac}} \leftarrow 0$ and $g_k \leftarrow 1$ for all $k \in \mathcal{F}$.
  Initialize $\Pi \leftarrow V \setminus \{s, t\}$ as the set of non-scheduled operations, and Q_k as an empty list for all $k \in \mathcal{F}$.
  while $\Pi \neq \emptyset$ **do**
    for $v \in \Pi$ **do**
      if $\Pi \cap \{i \mid (i, v) \in A\} = \emptyset$ **then**
        $r_v^{\text{op}} \leftarrow \max\{c_i \mid i \in V \setminus \Pi \text{ such that } (i, v) \in A\}$
    Set $r_{\min} = \min\{\max(r_v^{\text{op}}, r_k^{\text{mac}}) \mid v \in \Pi, k \in \mathcal{F}_v\}$ and let E be the set of pairs (v, k) with $v \in \Pi$ and $k \in \mathcal{F}_v$ such that $\max(r_v^{\text{op}}, r_k^{\text{mac}}) = r_{\min}$.
    $(\hat{v}, \hat{k}) \leftarrow \text{argmin}\{r_{\min} + \psi_\alpha(p_{v,k}, g_k) \mid (v, k) \in E\}$.
    Define $w_{\hat{v}} := \psi_\alpha(p_{\hat{v},\hat{k}}, g_{\hat{k}})$, $f_{\hat{v}} := \hat{k}$ and $c_{\hat{v}} := \max(r_{\hat{v}}^{\text{op}}, r_{\hat{k}}^{\text{mac}}) + w_{\hat{v}}$, and set $r_{\hat{k}}^{\text{mac}} \leftarrow c_{\hat{v}}$ and $g_{\hat{k}} \leftarrow g_{\hat{k}} + 1$.
    if $|Q_{\hat{k}}| \neq 0$ **then**
      Let $Q_{\hat{k}} = i_1, \ldots, i_{|Q_{\hat{k}}|}$. Set $A \leftarrow A \cup \{(i_{|Q_{\hat{k}}|}, \hat{v})\}$.
    Insert $\hat{v}$ at the end of $Q_{\hat{k}}$ and set $\Pi \leftarrow \Pi \setminus \{\hat{v}\}$.
  CriticalPath($\mathcal{F}$, f, w, Q, G, $\mathcal{U}$, $\mathcal{P}$, $C_{\max}$, τ).

The vector τ stores, in element τ_k, the largest position in the list Q_k (list of operations assigned to machine k) that contains an operation in the critical path. Algorithm 2 has worst case time complexity $O(|V| + |A|)$. Since $|V| = O(|\mathcal{O}|)$ and $|A| = O(|\widehat{A}| + |\mathcal{O}| + |\mathcal{F}|)$, this complexity translates into $O(|\mathcal{O}| + |\widehat{A}| + |\mathcal{F}|)$. The topological order of Algorithm 2 is calculated with the help of Algorithm 3, which implements a depth-first search. Algorithm 3 computes, optionally, for each $v \in V$, the set $R_v^{\leftarrow}$ formed by the vertices w such that there exists in G a path from w to v, i.e. the vertices that can reach v. Algorithm 3 is recursive and its complexity is $O(|V|+|\widehat{A}|)$.

Algorithm 4 presents the constructive heuristic based on the ECT rule. The algorithm is very similar to Algorithm 1, except for one detail. In the constructive heuristic based on EST, we first compute the instant $r_{\min}$ which is the earliest instant at which an unscheduled operation could be initiated. All operation/machine pairs that could start at that instant are considered and the pair with the shortest processing time is selected. But since they would all start at instant $r_{\min}$, saying that the pair with the shortest processing time is chosen is the same as saying that the pair that ends earliest is selected. This is the idea that is taken to the

Algorithm 2: Computes a critical path $\mathcal{P}$ and its length ξ for a given graph $G = (V, A)$. In addition, if τ is present as an input parameter, determines the last critical operation in each machine.

Input: $\mathcal{F}$, f, w, Q, $G = (V, A)$, τ
Output: $\mathcal{U}$, $\mathcal{P}$, ξ, τ

function CriticalPath($\mathcal{F}$, f, w, Q, G, $\mathcal{U}$, $\mathcal{P}$, ξ, τ)
  Initialize $d_i \leftarrow -\infty$ for all $i \in V \setminus \{s\}$ and define $d_s := 0$ and $\pi_s := 0$.
  Initialize $\mathcal{V} \leftarrow \emptyset$ and $\mathcal{U}$ as an empty list and compute in $\mathcal{U}$ a topological sort of the vertices in V, by calling TopologicalSort+(G, $\mathcal{U}$, s, $\mathcal{V}$).
  for $\ell = 1, \ldots, |V|$ **do**
    Let i be the ℓ-th operation in the topological order given by $\mathcal{U}$.
    for j such that $(i, j) \in A$ **do**
      if $d_j < d_i + w_i$ **then**
        $d_j \leftarrow d_i + w_i$ and $\pi_j \leftarrow i$.
  $\xi := d_t$
  Initialize $i \leftarrow \pi_t$, $\mathcal{P} \leftarrow \emptyset$, and $\tau_k \leftarrow 0$ for all $k \in \mathcal{F}$.
  do
    if $\tau_{f_i} = 0$ **then**
      Let Q_{f_i} be given by the sequence $i_1, \ldots, i_{\ell-1}, i, i_{\ell+1}, \ldots, i_{|Q_{f_i}|}$. Define $\tau_{f_i} := \ell$.
    $\mathcal{P} \leftarrow \mathcal{P} \cup \{i\}$ and $i \leftarrow \pi_i$.
  while $i \neq s$

Algorithm 3: Computes a topological sort $\mathcal{U}$ of the vertices of $G = (V, A)$. In addition, if v and $\mathcal{R}_v^{\leftarrow}$ are present as an input parameter, computes the set $\mathcal{R}_v^{\leftarrow}$ of vertices that reaches v in $G = (V, A)$.

Input: $G = (V, A)$, $\mathcal{U}$, i, $\mathcal{V}$, v, $\mathcal{R}_v^{\leftarrow}$
Output: $\mathcal{U}$, $\mathcal{V}$, $\mathcal{R}_v^{\leftarrow}$

function TopologicalSort+(G, $\mathcal{U}$, i, $\mathcal{V}$, v, $\mathcal{R}_v^{\leftarrow}$)
  Set $\mathcal{V} \leftarrow \mathcal{V} \cup \{i\}$.
  for j such that $(i, j) \in A$ **do**
    if $j \notin \mathcal{V}$ **then**
      TopologicalSort+(G, $\mathcal{U}$, j, $\mathcal{V}$, v, $\mathcal{R}_v^{\leftarrow}$)
    if $i \notin \mathcal{R}_v^{\leftarrow}$ and $j \in \mathcal{R}_v^{\leftarrow}$ **then**
      set $\mathcal{R}_v^{\leftarrow} \leftarrow \mathcal{R}_v^{\leftarrow} \cup \{i\}$.
  Insert i at the beginning of $\mathcal{U}$.

extreme in the constructive heuristic based on the ECT rule: without limiting the choice to the operation/machine pairs that could start as early as possible, the operation/machine pair that will finish earliest is chosen, even if the processing of the operation does not start as early as

possible. The worst case time complexity of Algorithm 4 is the same as that of Algorithm 1.

Algorithm 4: Computes the solution graph $G = (V, A)$, f, Q, and w by using ECT dispatching rule. Then, in G, it computes the largest path $\mathcal{P}$ from s to t and its length $C_{\max}$.

Input: $\mathcal{O}$, $\mathcal{F}$, p, $\widehat{A}$
Output: f, w, Q, $G = (V, A)$, $\mathcal{U}$, $\mathcal{P}$, $C_{\max}$, τ

function ECT($\mathcal{O}$, $\mathcal{F}$, p, $\widehat{A}$, f, w, Q, G, $\mathcal{U}$, $\mathcal{P}$, $C_{\max}$, τ)
  Set $A \leftarrow \widehat{A} \cup \{(s, j) \mid (\cdot, j) \notin \widehat{A}\} \cup \{(i, t) \mid (i, \cdot) \notin \widehat{A}\}$ and define $V := \mathcal{O} \cup \{s, t\}$ and $G = (V, A)$.
  Set $r_v^{\text{op}} \leftarrow +\infty$ and define $r_s^{\text{op}} := 0$, $w_s := w_t := 0$, and $c_s := 0$.
  Set $r_k^{\text{mac}} \leftarrow 0$ and $g_k \leftarrow 1$ for all $k \in \mathcal{F}$.
  Initialize $\Pi \leftarrow V \setminus \{s, t\}$ as the set of non-scheduled operations, and Q_k as an empty list for all $k \in \mathcal{F}$.
  while $\Pi \neq \emptyset$ **do**
    for $v \in \Pi$ **do**
      if $\Pi \cap \{i \mid (i, v) \in A\} = \emptyset$ **then**
        $r_v^{\text{op}} \leftarrow \max\{c_i \mid i \in V \setminus \Pi \text{ such that } (i, v) \in A\}$
    $(\hat{v}, \hat{k}) \leftarrow \text{argmin}\{\max(r_v^{\text{op}}, r_k^{\text{mac}}) + \psi_\alpha(p_{v,k}, g_k) \mid v \in \Pi, k \in \mathcal{F}_v\}$.
    Define $w_{\hat{v}} := \psi_\alpha(p_{\hat{v},\hat{k}}, g_{\hat{k}})$, $f_{\hat{v}} := \hat{k}$ and $c_{\hat{v}} := \max(r_{\hat{v}}^{\text{op}}, r_{\hat{k}}^{\text{mac}}) + w_{\hat{v}}$ and set $r_{\hat{k}}^{\text{mac}} \leftarrow c_{\hat{v}}$ and $g_{\hat{k}} \leftarrow g_{\hat{k}} + 1$.
    if $|Q_{\hat{k}}| \neq 0$ **then**
      Let $Q_{\hat{k}} = i_1, \ldots, i_{|Q_{\hat{k}}|}$. Set $A \leftarrow A \cup \{(i_{|Q_{\hat{k}}|}, \hat{v})\}$.
    Insert $\hat{v}$ at the end of $Q_{\hat{k}}$ and set $\Pi \leftarrow \Pi \setminus \{\hat{v}\}$.
  CriticalPath($\mathcal{F}$, f, w, Q, G, $\mathcal{U}$, $\mathcal{P}$, $C_{\max}$, τ).

Appendix B: Tables with the results of the experiments

Tables B1 and B2 provide a comprehensive overview of the instances main characteristics. In the tables, $|\mathcal{O}|$ is the number of operations, $|\mathcal{F}|$ is the number of machines, n is the number of jobs (connected components in the precedence constraints DAG $D = (\mathcal{O}, \widehat{A})$), $|\widehat{A}|$ is the number of precedence constraints, $\sum_{i \in \mathcal{O}} |\mathcal{F}_i|$ is the number of "true" entries in the logical matrix of dimension $|\mathcal{O}| \times |\mathcal{F}|$ whose (i, k) says whether operation i can be processed by machine k, and ω_1 and ω_2 are the measures of the routes' nonlinearity and routing flexibility described in Section 4.1. In addition, for the MILP and the CP models, the tables show the number of variables and constraints.

Tables B3 and B4 show the results of the EST and ECT constructive heuristics. For each instance, the lowest makespan, among the solutions found by the two constructive heuristics, is shown in bold.

Tables B5, B6, B7, B8, B9, and B10 show the results of applying the two exact solvers, CPLEX and CP Optimizer, to solve the instances of the MILP and CP models, respectively, when the warm start is not taken into account. Tables B5, B6, and B7 and 0.3, respectively; while Tables B8, B9, and B10 show the same thing for the large-scale instances. Tables B11–B16 show the results when the warm start is taken into account. The tables show the information related to the resolution of the MILP and CP models. For each instance, we show the makespan of the best feasible solution found, the best lower bound, and the gap. The gap is calculated as the makespan of the best feasible solution minus the best lower bound, divided by the makespan of the best feasible solution and multiplied by 100. As measures of computational effort, for the MILP solver the tables show the number of iterations, the number of nodes explored in the branch-and-bound search tree, and the CPU time in seconds. For the CP solver, the tables show the number of branches and the CPU time in seconds. If no information is displayed for a particular instance, it means that the solver was unable to find a feasible solution within the specified CPU time limit. Additionally, all tables show in the 'EST/ECT' column the makespan value of the best among the solutions computed by the EST and ECT constructive heuristics. This makespan corresponds to the solution used as the initial solution in the warm start case.

Main instance characteristics								MILP formulation			CP Optimizer formulation	
Instance	$\lvert\mathcal{F}\rvert$	$\lvert\mathcal{O}\rvert$	n	$\lvert\widehat{A}\rvert$	$\sum_{i\in\mathcal{O}}\mathcal{F}_i$	ω_1	ω_2	#binary variables	#continous variables	#constraints	#interval variables	#constraints
miniDAFJS01	5	14	2	14	44	0.54	0.54	392	78	3,754	406	3,564
miniDAFJS02	5	11	2	9	35	0.57	0.55	251	63	2,018	262	1,867
miniDAFJS03	5	10	2	12	31	0.50	0.53	197	57	1,435	207	1,301
miniDAFJS04	5	9	2	9	29	0.25	0.56	175	53	1,234	184	1,109
miniDAFJS05	5	15	2	15	38	0.39	0.38	310	74	2,881	325	2,714
miniDAFJS06	5	14	2	14	46	0.20	0.57	448	80	4,742	462	4,544
miniDAFJS07	5	11	2	10	34	0.23	0.52	240	62	1,924	251	1,777
miniDAFJS08	5	9	2	8	24	0.42	0.42	122	48	770	131	665
miniDAFJS09	5	15	2	13	47	0.63	0.53	465	83	5,018	480	4,815
miniDAFJS10	5	11	2	10	32	0.28	0.48	208	60	1,530	219	1,391
miniDAFJS11	5	18	2	16	43	0.45	0.35	409	85	4,371	427	4,181
miniDAFJS12	5	12	2	12	38	0.55	0.54	298	68	2,602	310	2,438
miniDAFJS13	5	10	2	8	27	0.58	0.43	149	53	979	159	861
miniDAFJS14	5	14	2	12	44	0.50	0.54	394	78	3,800	408	3,610
miniDAFJS15	5	11	2	11	35	0.40	0.55	261	63	2,242	272	2,091
miniDAFJS16	5	13	2	12	42	0.18	0.56	358	74	3,302	371	3,121
miniDAFJS17	5	11	2	11	30	0.13	0.43	194	58	1,491	205	1,360
miniDAFJS18	5	11	2	10	34	0.38	0.52	234	62	1,798	245	1,651
miniDAFJS19	5	12	2	10	36	0.30	0.50	272	66	2,338	284	2,182
miniDAFJS20	5	13	2	11	40	0.54	0.52	350	72	3,447	363	3,274
miniDAFJS21	5	19	3	20	60	0.52	0.54	730	104	9,298	749	9,039
miniDAFJS22	5	18	3	18	57	0.22	0.54	665	99	8,223	683	7,977
miniDAFJS23	5	21	3	18	66	0.53	0.54	874	114	11,934	895	11,649
miniDAFJS24	5	18	3	17	56	0.36	0.53	642	98	7,809	660	7,567
miniDAFJS25	5	17	3	17	51	0.12	0.50	535	91	6,024	552	5,803
miniDAFJS26	5	17	3	15	53	0.36	0.53	573	93	6,560	590	6,331
miniDAFJS27	5	19	3	16	57	0.46	0.50	721	101	10,155	740	9,908
miniDAFJS28	5	16	3	15	48	0.54	0.50	468	86	4,871	484	4,663
miniDAFJS29	5	14	3	15	47	0.33	0.59	451	81	4,634	465	4,432
miniDAFJS30	5	22	3	21	61	0.32	0.44	791	111	11,020	813	10,754
miniYFJS01	7	16	4	12	54	0.25	0.40	420	94	3,554	436	3,322
miniYFJS02	7	16	4	12	44	0.33	0.29	294	84	2,298	310	2,106
miniYFJS03	7	16	4	12	45	0.08	0.30	333	85	2,915	349	2,719
miniYFJS04	7	16	4	12	53	0.25	0.39	439	93	4,083	455	3,855
miniYFJS05	7	16	4	12	55	0.50	0.41	457	95	4,249	473	4,013
miniYFJS06	7	16	4	12	51	0.42	0.36	407	91	3,773	423	3,553
miniYFJS07	7	16	4	12	48	0.17	0.33	350	88	2,912	366	2,704
miniYFJS08	7	16	4	12	49	0.50	0.34	363	89	3,037	379	2,825
miniYFJS09	7	16	4	12	51	0.33	0.36	399	91	3,569	415	3,349
miniYFJS10	7	16	4	12	59	0.17	0.45	509	99	4,767	525	4,515
miniYFJS11	7	20	5	15	56	0.13	0.30	464	104	4,241	484	3,997
miniYFJS12	7	20	5	15	68	0.13	0.40	716	116	8,363	736	8,071
miniYFJS13	7	20	5	15	69	0.40	0.41	723	117	8,272	743	7,976
miniYFJS14	7	20	5	15	59	0.60	0.33	509	107	4,772	529	4,516
miniYFJS15	7	20	5	15	53	0.33	0.28	429	101	3,944	449	3,712
miniYFJS16	7	20	5	15	63	0.40	0.36	617	111	6,838	637	6,566
miniYFJS17	7	20	5	15	57	0.27	0.31	485	105	4,576	505	4,328
miniYFJS18	7	20	5	15	51	0.07	0.26	395	99	3,514	415	3,290
miniYFJS19	7	20	5	15	58	0.47	0.32	512	106	5,031	532	4,779
miniYFJS20	7	20	5	15	62	0.53	0.35	638	110	7,295	658	7,027
miniYFJS21	7	24	6	18	72	0.28	0.33	756	128	8,466	780	8,154
miniYFJS22	7	24	6	18	82	0.28	0.40	996	138	12,962	1,020	12,610
miniYFJS23	7	24	6	18	78	0.17	0.38	908	134	11,280	932	10,944
miniYFJS24	7	24	6	18	62	0.28	0.26	588	118	6,136	612	5,864
miniYFJS25	7	24	6	18	76	0.17	0.36	836	132	9,686	860	9,358
miniYFJS26	7	24	6	18	67	0.33	0.30	693	123	7,841	717	7,549
miniYFJS27	7	24	6	18	81	0.39	0.40	951	137	11,709	975	11,361
miniYFJS28	7	24	6	18	67	0.22	0.30	661	123	7,001	685	6,709
miniYFJS29	7	24	6	18	80	0.39	0.39	934	136	11,536	958	11,192
miniYFJS30	7	24	6	18	72	0.44	0.33	800	128	9,756	824	9,444

Table B1: Main features of the proposed sixty small-scale instances.

Main instance characteristics								MILP formulation			CP Optimizer formulation	
Instance	$\lvert\mathcal{F}\rvert$	$\lvert\mathcal{O}\rvert$	n	$\lvert\widehat{A}\rvert$	$\sum_{i\in\mathcal{O}}\mathcal{F}_i$	ω_1	ω_2	#binary variables	#continous variables	#constraints	#interval variables	#constraints
DAFJS01	5	26	4	26	82	0.54	0.54	1,358	140	23,102	1,384	22,748
DAFJS02	5	25	4	23	79	0.45	0.54	1,273	135	21,348	1,298	21,007
DAFJS03	10	55	4	52	279	0.32	0.45	7,849	400	223,911	7,904	222,740
DAFJS04	10	43	4	40	220	0.25	0.46	4,960	317	115,208	5,003	114,285
DAFJS05	5	39	6	34	104	0.35	0.42	2,242	188	50,228	2,281	49,773
DAFJS06	5	44	6	41	136	0.38	0.52	3,724	230	103,253	3,768	102,665
DAFJS07	10	85	6	82	431	0.30	0.45	18,695	612	817,681	18,780	815,872
DAFJS08	10	85	6	82	403	0.31	0.42	16,357	584	670,739	16,442	669,042
DAFJS09	5	45	8	42	135	0.40	0.50	3,755	231	107,667	3,800	107,082
DAFJS10	5	58	8	52	168	0.40	0.47	5,764	290	201,720	5,822	200,990
DAFJS11	10	113	8	108	534	0.40	0.41	28,648	771	1,546,720	28,761	1,544,471
DAFJS12	10	117	8	114	603	0.49	0.46	36,513	848	2,223,141	36,630	2,220,612
DAFJS13	5	62	10	55	193	0.41	0.53	7,511	323	295,582	7,573	294,748
DAFJS14	5	69	10	62	206	0.37	0.50	8,578	350	361,926	8,647	361,033
DAFJS15	10	120	10	117	595	0.32	0.44	35,811	846	2,184,302	35,931	2,181,802
DAFJS16	10	120	10	114	602	0.33	0.45	36,344	853	2,203,066	36,464	2,200,538
DAFJS17	5	82	12	77	246	0.43	0.50	12,244	416	617,041	12,326	615,975
DAFJS18	5	74	12	64	231	0.41	0.53	10,785	385	509,435	10,859	508,437
DAFJS19	7	70	8	66	283	0.34	0.51	11,507	431	471,949	11,577	470,747
DAFJS20	7	92	10	87	361	0.36	0.49	18,709	553	976,026	18,801	974,490
DAFJS21	7	107	12	102	425	0.38	0.50	25,853	647	1,577,783	25,960	1,575,976
DAFJS22	7	116	12	109	450	0.39	0.48	29,296	690	1,932,281	29,412	1,930,365
DAFJS23	9	76	8	71	367	0.31	0.48	15,103	529	628,938	15,179	627,394
DAFJS24	9	92	8	87	463	0.31	0.50	23,893	657	1,238,922	23,985	1,236,978
DAFJS25	9	123	10	119	619	0.31	0.50	42,753	875	2,967,430	42,876	2,964,831
DAFJS26	9	119	10	116	606	0.34	0.51	41,026	854	2,794,800	41,145	2,792,257
DAFJS27	9	127	12	118	625	0.27	0.49	43,461	889	3,029,045	43,588	3,026,418
DAFJS28	10	91	8	89	457	0.32	0.45	21,065	650	980,790	21,156	978,871
DAFJS29	10	95	8	94	468	0.34	0.44	22,450	669	1,109,378	22,545	1,107,411
DAFJS30	10	98	10	94	509	0.20	0.47	26,059	716	1,344,045	26,157	1,341,911
YFJS01	7	40	4	36	104	0.10	0.27	1,824	192	38,286	1,864	37,830
YFJS02	7	40	4	36	104	0.17	0.27	1,568	192	24,498	1,608	24,042
YFJS03	7	24	6	18	63	0.28	0.27	611	119	6,561	635	6,285
YFJS04	7	28	7	21	71	0.19	0.26	813	135	10,260	841	9,948
YFJS05	7	32	8	24	81	0.33	0.26	1,003	153	13,417	1,035	13,061
YFJS06	7	36	9	27	95	0.19	0.27	1,365	175	21,214	1,401	20,798
YFJS07	7	36	9	27	93	0.26	0.26	1,279	173	18,588	1,315	18,180
YFJS08	12	36	9	27	100	0.26	0.16	888	185	8,879	924	8,443
YFJS09	12	36	9	27	219	0.22	0.46	4,079	304	78,474	4,115	77,562
YFJS10	12	40	10	30	113	0.17	0.17	1,169	206	13,593	1,209	13,101
YFJS11	10	50	10	40	134	0.22	0.19	1,860	245	27,378	1,910	26,792
YFJS12	10	50	10	40	133	0.12	0.18	1,915	244	30,085	1,965	29,503
YFJS13	10	50	10	40	137	0.15	0.19	1,895	248	27,177	1,945	26,579
YFJS14	26	221	13	208	641	0.24	0.08	16,603	1,110	454,179	16,824	451,394
YFJS15	26	221	13	208	648	0.23	0.08	16,620	1,117	441,752	16,841	438,939
YFJS16	26	221	13	208	633	0.13	0.07	16,037	1,102	424,253	16,258	421,500
YFJS17	26	289	17	272	1328	0.15	0.14	68,502	1,933	3,572,342	68,791	3,566,741
YFJS18	26	289	17	272	1362	0.15	0.15	72,354	1,967	3,901,378	72,643	3,895,641
YFJS19	26	289	17	272	1347	0.20	0.15	70,527	1,952	3,737,737	70,816	3,732,060
YFJS20	26	289	17	272	1343	0.12	0.15	70,371	1,948	3,742,757	70,660	3,737,096

Table B2: Main features of the fifty large-scale instances from [4].

instance	$\alpha = 0.1$		$\alpha = 0.2$		$\alpha = 0.3$	
	EST	ECT	EST	ECT	EST	ECT
miniDAFJS01	**23,264**	23,948	**22,156**	23,199	**20,757**	21,535
miniDAFJS02	**23,242**	23,860	**22,161**	22,775	**21,152**	21,461
miniDAFJS03	**18,363**	**18,363**	**17,972**	**17,972**	**17,621**	**17,621**
miniDAFJS04	**21,690**	**21,690**	21,233	**21,121**	20,824	**20,391**
miniDAFJS05	24,279	**22,598**	21,815	**20,826**	20,623	**19,253**
miniDAFJS06	23,726	**23,370**	23,781	**22,193**	**21,148**	21,480
miniDAFJS07	**28,644**	33,413	27,551	**25,088**	26,600	**24,767**
miniDAFJS08	**19,878**	25,100	**18,857**	22,921	**17,927**	20,943
miniDAFJS09	**25,425**	27,806	**23,830**	25,400	**22,398**	23,295
miniDAFJS10	22,847	**21,563**	21,730	**21,021**	20,349	**20,176**
miniDAFJS11	35,166	**33,911**	32,214	**31,269**	29,739	**28,718**
miniDAFJS12	**20,342**	21,435	**19,624**	20,370	**19,094**	19,394
miniDAFJS13	**18,313**	18,681	**17,143**	17,181	16,077	**15,425**
miniDAFJS14	**26,503**	27,525	**24,958**	25,919	**23,552**	24,467
miniDAFJS15	24,961	**22,472**	23,710	**21,969**	22,545	**20,524**
miniDAFJS16	25,845	**25,691**	24,699	**24,593**	26,750	**23,797**
miniDAFJS17	**21,070**	21,144	**20,519**	**20,519**	**19,941**	**19,941**
miniDAFJS18	19,716	**19,308**	**18,829**	19,133	**17,784**	18,395
miniDAFJS19	**21,293**	21,329	**20,107**	20,340	**19,030**	19,426
miniDAFJS20	**23,570**	26,421	21,759	**21,286**	20,140	**19,587**
miniDAFJS21	25,764	**25,020**	24,040	**22,689**	23,171	**21,299**
miniDAFJS22	**28,122**	31,892	**26,289**	29,188	**24,671**	26,829
miniDAFJS23	**27,566**	33,389	**25,592**	30,242	**23,847**	28,000
miniDAFJS24	**26,932**	27,581	29,407	**24,500**	27,040	**22,524**
miniDAFJS25	**23,370**	25,879	**22,613**	24,209	**21,923**	22,976
miniDAFJS26	26,408	**25,974**	24,476	**23,630**	22,771	**21,613**
miniDAFJS27	**28,084**	31,060	**25,429**	26,652	**23,089**	23,595
miniDAFJS28	**28,836**	30,947	**26,587**	27,477	26,831	**24,429**
miniDAFJS29	27,569	**21,151**	25,923	**19,789**	24,444	**18,560**
miniDAFJS30	27,870	**27,249**	25,318	**24,491**	**21,421**	22,071
wins	19	13	16	16	16	16
mean	24,621.93	25,325.67	23,344.07	23,265.40	22,108.63	21,749.73

instance	$\alpha = 0.1$		$\alpha = 0.2$		$\alpha = 0.3$	
	EST	ECT	EST	ECT	EST	ECT
miniYFJS01	40,456	**35,243**	38,132	**34,443**	36,008	**33,697**
miniYFJS02	34,009	**28,688**	31,940	**27,557**	27,987	**25,969**
miniYFJS03	**62,704**	69,005	**57,950**	60,302	53,679	**52,865**
miniYFJS04	32,133	**25,394**	30,048	**24,669**	24,428	**24,017**
miniYFJS05	29,009	**27,650**	26,828	**26,135**	27,047	**24,743**
miniYFJS06	49,001	**30,952**	43,387	**29,080**	40,871	**27,366**
miniYFJS07	**51,782**	57,652	**47,743**	53,702	**44,128**	50,110
miniYFJS08	**35,336**	51,013	**33,178**	40,884	**31,133**	40,567
miniYFJS09	**38,626**	39,463	40,411	**37,593**	38,480	**34,357**
miniYFJS10	**32,975**	34,363	**30,651**	31,742	**28,513**	29,397
miniYFJS11	51,600	**51,212**	48,172	**47,079**	45,079	**43,254**
miniYFJS12	**38,889**	39,024	**36,036**	36,691	**33,495**	35,223
miniYFJS13	38,448	**35,177**	34,046	**31,720**	**26,536**	28,751
miniYFJS14	**41,560**	42,764	**39,254**	39,862	**37,157**	37,177
miniYFJS15	**57,017**	58,693	**51,504**	**51,504**	**46,583**	**46,583**
miniYFJS16	39,097	**35,831**	36,810	**33,285**	34,799	**30,576**
miniYFJS17	**53,052**	62,576	**48,745**	57,600	**44,916**	50,137
miniYFJS18	**37,217**	39,284	34,053	**31,911**	31,154	**29,764**
miniYFJS19	46,640	**46,478**	41,729	**41,451**	42,825	**37,034**
miniYFJS20	**42,408**	42,465	39,852	**39,600**	37,509	**37,063**
miniYFJS21	50,076	**44,016**	45,541	**40,765**	40,884	**35,905**
miniYFJS22	**43,722**	47,389	40,428	**32,485**	31,887	**30,212**
miniYFJS23	**53,046**	59,584	**49,209**	52,628	**45,823**	46,589
miniYFJS24	**42,268**	45,974	**39,883**	41,759	**37,651**	37,998
miniYFJS25	**46,936**	68,698	**40,933**	61,260	**35,966**	54,247
miniYFJS26	55,031	**55,022**	49,428	**49,412**	**53,604**	54,006
miniYFJS27	43,319	**41,060**	39,256	**36,561**	35,655	**33,799**
miniYFJS28	46,385	**43,232**	42,230	**40,325**	39,363	**37,547**
miniYFJS29	**48,806**	55,473	43,757	**41,834**	39,279	**39,146**
miniYFJS30	**49,458**	52,043	**45,051**	46,003	41,116	**40,752**
wins	17	13	12	19	12	19
mean	44,366.87	45,513.93	40,872.83	40,661.40	37,785.17	37,628.37

Table B3: Makespan values for the small-scale set of instance solved by constructive heuristics.

instance	$\alpha = 0.1$		$\alpha = 0.2$		$\alpha = 0.3$	
	EST	ECT	EST	ECT	EST	ECT
DAFJS01	**29,769**	41,358	**28,920**	31,657	**22,616**	28,222
DAFJS02	**32,467**	33,155	29,089	**28,154**	**26,183**	26,949
DAFJS03	**53,688**	53,834	**48,555**	51,717	**43,964**	45,501
DAFJS04	**54,082**	54,150	**48,461**	48,503	**43,281**	44,094
DAFJS05	52,651	**45,862**	**44,790**	44,919	**38,237**	38,457
DAFJS06	**51,925**	56,012	**44,228**	46,403	**38,007**	41,229
DAFJS07	**57,193**	59,491	51,019	**50,802**	**41,285**	44,518
DAFJS08	**62,159**	67,107	53,998	**53,439**	**46,323**	47,173
DAFJS09	**48,680**	60,565	48,833	**42,537**	**41,418**	42,518
DAFJS10	**58,695**	60,374	**46,257**	49,140	**38,771**	41,125
DAFJS11	**67,594**	92,639	**56,847**	75,992	**47,805**	56,547
DAFJS12	70,287	**68,322**	58,100	**55,575**	**45,735**	49,991
DAFJS13	63,386	**61,794**	**51,317**	53,486	**42,319**	44,724
DAFJS14	83,362	**76,862**	**62,724**	63,887	**49,459**	50,737
DAFJS15	78,413	**72,288**	**54,353**	69,279	**49,206**	54,179
DAFJS16	78,289	**76,204**	**65,550**	72,885	57,213	**56,727**
DAFJS17	**73,219**	84,719	**63,177**	65,307	**54,527**	54,887
DAFJS18	82,129	**78,862**	**61,831**	67,753	**51,823**	52,325
DAFJS19	**66,412**	67,169	**58,001**	63,933	49,746	**46,880**
DAFJS20	**78,778**	80,562	**63,588**	68,034	**51,686**	54,531
DAFJS21	**78,320**	83,933	**63,202**	66,878	**53,010**	54,906
DAFJS22	77,853	**70,892**	**56,115**	62,199	**45,005**	49,723
DAFJS23	**49,969**	53,123	**47,616**	50,639	**39,249**	41,475
DAFJS24	**57,411**	62,038	**49,019**	50,345	**44,851**	46,119
DAFJS25	89,248	**82,055**	**66,262**	67,533	**51,964**	60,266
DAFJS26	**81,480**	83,635	75,230	**72,226**	**57,182**	60,803
DAFJS27	**81,470**	88,629	**64,797**	71,903	**58,138**	58,145
DAFJS28	**62,568**	64,560	**52,639**	53,110	**42,898**	46,294
DAFJS29	74,841	**72,938**	**59,792**	66,900	**51,606**	58,809
DAFJS30	**61,147**	70,062	**55,015**	67,245	**43,859**	47,605
wins	20	10	24	6	28	2
mean	65,249.50	67,439.80	54,310.83	57,746.00	45,578.87	48,181.97

instance	$\alpha = 0.1$		$\alpha = 0.2$		$\alpha = 0.3$	
	EST	ECT	EST	ECT	EST	ECT
YFJS01	**87,203**	106,117	**84,152**	92,107	**70,402**	80,419
YFJS02	87,462	**81,579**	73,957	**66,853**	68,075	**61,111**
YFJS03	42,457	**40,197**	**35,380**	37,159	**31,680**	33,077
YFJS04	**47,467**	50,724	**43,614**	44,467	39,249	**38,898**
YFJS05	**46,138**	55,871	**40,851**	49,893	**41,586**	45,012
YFJS06	53,210	**52,487**	**48,564**	54,660	**46,811**	47,173
YFJS07	63,320	**54,457**	56,313	**51,261**	**39,896**	44,004
YFJS08	51,818	**49,626**	47,218	**44,074**	**39,325**	39,469
YFJS09	38,836	**28,354**	36,123	**26,072**	32,120	**24,027**
YFJS10	**42,583**	59,808	**40,658**	52,811	**39,607**	48,821
YFJS11	65,011	**59,356**	58,106	**51,469**	51,943	**45,051**
YFJS12	**70,830**	74,978	**59,898**	62,299	53,275	**52,398**
YFJS13	53,601	**50,805**	48,729	**45,084**	41,303	**40,119**
YFJS14	151,365	**129,428**	116,157	**109,469**	111,313	**92,457**
YFJS15	152,375	**138,196**	120,600	**112,424**	107,006	**96,547**
YFJS16	144,976	**127,055**	131,230	**106,855**	118,515	**92,811**
YFJS17	133,982	**109,112**	110,203	**85,736**	98,045	**73,682**
YFJS18	154,214	**133,703**	121,563	**99,429**	104,338	**87,059**
YFJS19	133,142	**107,055**	110,125	**89,561**	91,287	**74,431**
YFJS20	137,326	**97,868**	104,036	**91,958**	91,229	**72,481**
wins	5	15	7	13	7	13
mean	87,865.80	80,338.80	74,373.85	68,682.05	65,850.25	59,452.35

Table B4: Makespan values for the testbed set of instance solved by constructive heuristics.

Instance	EST/ECT	CPLEX						CP Optimizer				
		Makespan	Best Bound	Gap(%)	#iter	#B&B Nodes	CPU	Makespan	Best bound	Gap(%)	#branches	CPU
miniDAFJS01	23,264	22,875	22,875.0	0.00	278,358	2,630	20.12	22,875	22,875	0.00	175,238	3.03
miniDAFJS02	23,242	22,708	22,708.0	0.00	42,653	579	1.62	22,708	22,708	0.00	29,890	0.39
miniDAFJS03	18,363	18,363	18,363.0	0.00	10,016	153	0.45	18,363	18,363	0.00	10,032	0.12
miniDAFJS04	21,690	20,498	20,498.0	0.00	4,002	133	0.21	20,498	20,498	0.00	14,529	0.15
miniDAFJS05	22,598	20,593	20,593.0	0.00	44,305	555	2.42	20,593	20,593	0.00	63,587	1.06
miniDAFJS06	23,370	22,867	22,867.0	0.00	180,943	1,070	10.72	22,867	22,867	0.00	44,822	1.07
miniDAFJS07	28,644	25,715	25,715.0	0.00	18,460	197	0.96	25,715	25,715	0.00	24,541	0.19
miniDAFJS08	19,878	19,878	19,878.0	0.00	2,120	21	0.12	19,878	19,878	0.00	3,005	0.03
miniDAFJS09	25,425	24,267	24,267.0	0.00	225,648	1,247	12.74	24,267	24,267	0.00	343,445	6.54
miniDAFJS10	21,563	20,336	20,336.0	0.00	15,969	352	0.70	20,336	20,336	0.00	14,767	0.16
miniDAFJS11	33,911	29,968	29,968.0	0.00	599,704	2,896	37.92	29,968	29,968	0.00	246,046	4.43
miniDAFJS12	20,342	18,670	18,670.0	0.00	18,022	400	1.22	18,670	18,670	0.00	33,512	0.45
miniDAFJS13	18,313	16,313	16,313.0	0.00	2,056	83	0.10	16,313	16,313	0.00	2,616	0.03
miniDAFJS14	26,503	23,140	23,140.0	0.00	109,787	994	6.89	23,140	23,140	0.00	31,941	0.55
miniDAFJS15	22,472	21,715	21,715.0	0.00	15,342	251	0.77	21,715	21,715	0.00	33,798	0.48
miniDAFJS16	25,691	25,426	25,426.0	0.00	183,188	2,074	8.15	25,426	25,426	0.00	35,157	0.59
miniDAFJS17	21,070	20,155	20,155.0	0.00	1,930	57	0.28	20,155	20,155	0.00	15,047	0.16
miniDAFJS18	19,308	18,135	18,135.0	0.00	26,554	493	1.11	18,135	18,135	0.00	29,699	0.29
miniDAFJS19	21,293	20,945	20,945.0	0.00	50,938	518	1.89	20,945	20,945	0.00	23,203	0.33
miniDAFJS20	23,570	21,838	21,838.0	0.00	96,605	660	4.65	21,838	21,838	0.00	29,913	0.55
miniDAFJS21	25,020	23,344	23,343.2	0.00	3,343,147	19,080	527.41	23,344	23,344	0.00	9,456,242	210.85
miniDAFJS22	28,122	25,923	25,923.0	0.00	1,997,133	10,147	326.21	25,923	25,923	0.00	1,293,933	29.95
miniDAFJS23	27,566	24,038	24,038.0	0.00	8,609,240	22,211	1,646.96	24,038	24,038	0.00	18,995,339	448.40
miniDAFJS24	26,932	24,579	24,579.0	0.00	653,031	2,661	94.56	24,579	24,579	0.00	447,657	12.70
miniDAFJS25	23,370	21,143	21,142.9	0.00	677,724	5,764	119.35	21,143	21,143	0.00	152,959	3.67
miniDAFJS26	25,974	21,120	21,120.0	0.00	2,000,288	10,378	260.89	21,120	21,120	0.00	493,466	10.35
miniDAFJS27	28,084	22,050	22,049.1	0.00	4,105,982	23,180	716.68	22,050	22,050	0.00	2,561,331	57.24
miniDAFJS28	28,836	22,708	22,707.3	0.00	559,433	3,820	46.03	22,708	22,708	0.00	206,070	4.14
miniDAFJS29	21,151	20,278	20,278.0	0.00	56,304	407	3.54	20,278	20,278	0.00	40,983	0.88
miniDAFJS30	27,249	23,558	23,558.0	0.00	4,820,496	13,471	918.23	23,558	23,558	0.00	9,015,463	244.45
miniYFJS01	35,243	35,046	35,046.0	0.00	26,991	188	2.24	35,046	35,046	0.00	38,571	0.57
miniYFJS02	28,688	24,359	24,359.0	0.00	13,715	130	0.90	24,359	24,359	0.00	16,008	0.20
miniYFJS03	62,704	47,391	47,391.0	0.00	18,760	149	1.08	47,391	47,391	0.00	118,013	2.24
miniYFJS04	25,394	25,394	25,394.0	0.00	116,336	935	5.98	25,394	25,394	0.00	26,552	0.44
miniYFJS05	27,650	23,985	23,985.0	0.00	31,811	260	1.95	23,985	23,985	0.00	17,094	0.32
miniYFJS06	30,952	29,469	29,469.0	0.00	84,139	673	4.53	29,469	29,469	0.00	45,982	0.84
miniYFJS07	51,782	45,705	45,705.0	0.00	40,428	529	1.88	45,705	45,705	0.00	67,529	1.24
miniYFJS08	35,336	33,829	33,829.0	0.00	114,680	1,115	5.27	33,829	33,829	0.00	55,189	0.82
miniYFJS09	38,626	37,049	37,049.0	0.00	138,409	873	6.39	37,049	37,049	0.00	110,794	1.69
miniYFJS10	32,975	27,310	27,310.0	0.00	55,973	399	5.50	27,310	27,310	0.00	97,059	1.68
miniYFJS11	51,212	41,300	41,300.0	0.00	79,689	610	5.61	41,300	41,300	0.00	48,926	0.80
miniYFJS12	38,889	30,145	30,145.0	0.00	385,442	1,258	33.33	30,145	30,145	0.00	373,995	8.61
miniYFJS13	35,177	30,962	30,962.0	0.00	124,224	1,000	13.57	30,962	30,962	0.00	370,936	8.98
miniYFJS14	41,560	31,398	31,398.0	0.00	154,413	1,533	13.57	31,398	31,398	0.00	187,646	3.19
miniYFJS15	57,017	45,442	45,442.0	0.00	94,003	579	6.29	45,442	45,442	0.00	132,867	2.07
miniYFJS16	35,831	33,791	33,791.0	0.00	352,489	2,686	37.62	33,791	33,791	0.00	215,214	4.27
miniYFJS17	53,052	42,838	42,838.0	0.00	419,682	1,831	31.47	42,838	42,838	0.00	295,424	5.50
miniYFJS18	37,217	28,247	28,247.0	0.00	89,973	788	4.87	28,247	28,247	0.00	51,185	0.82
miniYFJS19	46,478	33,601	33,601.0	0.00	406,894	2,149	41.56	33,601	33,601	0.00	215,603	4.04
miniYFJS20	42,408	30,837	30,837.0	0.00	125,514	667	11.25	30,837	30,837	0.00	99,426	2.35
miniYFJS21	44,016	37,096	37,095.4	0.00	1,832,086	5,140	264.53	37,096	37,096	0.00	724,123	13.87
miniYFJS22	43,722	34,282	34,282.0	0.00	1,327,054	6,027	301.92	34,282	34,282	0.00	1,438,230	42.29
miniYFJS23	53,046	42,079	42,079.0	0.00	7,194,309	20,361	1,323.46	42,079	42,079	0.00	20,403,691	459.39
miniYFJS24	42,268	30,905	30,905.0	0.00	471,757	2,707	84.03	30,905	30,905	0.00	1,299,866	17.36
miniYFJS25	46,936	36,170	36,170.0	0.00	2,998,191	12,325	515.55	36,170	36,170	0.00	9,119,130	189.71
miniYFJS26	55,022	51,466	51,466.0	0.00	858,752	4,932	119.64	51,466	51,466	0.00	3,405,611	72.44
miniYFJS27	41,060	36,719	36,719.0	0.00	1,273,692	4,261	266.23	36,719	36,719	0.00	486,016	14.10
miniYFJS28	43,232	34,509	34,509.0	0.00	415,319	1,990	72.65	34,509	34,509	0.00	264,187	4.39
miniYFJS29	48,806	39,798	39,797.6	0.00	2,891,318	8,825	479.19	39,798	39,798	0.00	5,229,743	129.70
miniYFJS30	49,458	33,974	33,973.4	0.00	903,909	4,828	160.89	33,974	33,974	0.00	751,686	17.74
mean	33,309.52	28,637.37	28,637.30	0.00	863,156	3,604	143.26	28,637.37	28,637.37	0.00	1,492,909	34.25

Table B5: Solutions found and computational cost of solving the small-scale instances with learning index $\alpha = 0.1$ using CPLEX and CP Optimizer with no warm start.

Instance	EST/ECT	CPLEX						CP Optimizer					
		Makespan	Best Bound	Gap(%)	#iter	#B&B Nodes	CPU	Makespan	Best bound	Gap(%)	#branches	CPU
miniDAFJS01	22,156	21,327	21,327.0	0.00	546,297	4,285	31.12	21,327	21,327	0.00	1,331,435	23.68
miniDAFJS02	22,161	20,635	20,635.0	0.00	81,821	640	2.58	20,635	20,635	0.00	25,894	0.42
miniDAFJS03	17,972	17,972	17,972.0	0.00	19,340	305	0.69	17,972	17,972	0.00	32,641	0.28
miniDAFJS04	21,121	19,602	19,602.0	0.00	7,746	226	0.33	19,602	19,602	0.00	14,724	0.16
miniDAFJS05	20,826	18,803	18,803.0	0.00	98,953	1,120	4.30	18,803	18,803	0.00	153,026	2.60
miniDAFJS06	22,193	20,568	20,568.0	0.00	450,187	2,639	37.44	20,568	20,568	0.00	271,995	6.20
miniDAFJS07	25,088	24,715	24,715.0	0.00	52,190	642	1.92	24,715	24,715	0.00	18,976	0.28
miniDAFJS08	18,857	18,857	18,857.0	0.00	3,493	50	0.10	18,857	18,857	0.00	2,746	0.02
miniDAFJS09	23,830	22,660	22,660.0	0.00	1,001,850	6,554	70.61	22,660	22,660	0.00	607,820	12.82
miniDAFJS10	21,021	18,823	18,823.0	0.00	10,901	169	0.53	18,823	18,823	0.00	12,973	0.17
miniDAFJS11	31,269	27,455	27,455.0	0.00	672,082	3,718	48.57	27,455	27,455	0.00	427,853	8.48
miniDAFJS12	19,624	17,874	17,874.0	0.00	70,638	1,078	3.62	17,874	17,874	0.00	101,593	1.60
miniDAFJS13	17,143	15,143	15,143.0	0.00	6,291	160	0.17	15,143	15,143	0.00	13,669	0.12
miniDAFJS14	24,958	21,817	21,817.0	0.00	332,619	2,896	24.23	21,817	21,817	0.00	97,386	1.92
miniDAFJS15	21,969	20,236	20,236.0	0.00	69,714	883	2.72	20,236	20,236	0.00	134,447	2.13
miniDAFJS16	24,593	24,114	24,114.0	0.00	357,217	3,894	29.75	24,114	24,114	0.00	194,554	3.74
miniDAFJS17	20,519	19,145	19,145.0	0.00	19,035	378	0.80	19,145	19,145	0.00	15,158	0.21
miniDAFJS18	18,829	17,270	17,270.0	0.00	32,161	673	1.37	17,270	17,270	0.00	47,891	0.51
miniDAFJS19	20,107	19,642	19,642.0	0.00	71,718	942	3.26	19,642	19,642	0.00	34,360	0.58
miniDAFJS20	21,286	20,086	20,086.0	0.00	190,012	2,064	14.69	20,086	20,086	0.00	127,113	2.45
miniDAFJS21	22,689	21,352	21,351.2	0.00	9,965,539	37,095	1,442.12	21,352	21,352	0.00	40,689,078	990.33
miniDAFJS22	26,289	23,852	23,852.0	0.00	3,464,491	19,183	464.46	23,852	23,852	0.00	4,215,904	107.54
miniDAFJS23	25,592	23,209	18,537.6	20.13	19,297,413	101,750	3,600.00	22,390	14,739	34.17	138,647,206	3,600.00
miniDAFJS24	24,500	22,521	22,521.0	0.00	2,377,420	12,640	330.40	22,521	22,521	0.00	17,411,521	405.43
miniDAFJS25	22,613	19,809	19,808.8	0.00	1,886,995	12,894	238.57	19,809	19,809	0.00	1,139,912	23.55
miniDAFJS26	23,630	19,724	19,723.0	0.00	2,193,024	10,772	252.99	19,724	19,724	0.00	3,214,623	64.87
miniDAFJS27	25,429	20,245	19,599.0	3.19	21,678,115	79,151	3,600.00	20,245	20,245	0.00	23,704,164	595.59
miniDAFJS28	26,587	20,635	20,635.0	0.00	934,303	6,462	87.89	20,635	20,635	0.00	1,156,551	22.26
miniDAFJS29	19,789	19,201	19,201.0	0.00	217,357	1,998	20.53	19,201	19,201	0.00	235,077	4.22
miniDAFJS30	24,491	21,552	21,551.2	0.00	16,829,226	70,869	3,424.44	21,552	21,552	0.00	114,562,241	3,347.05
miniYFJS01	34,443	33,132	33,132.0	0.00	48,887	325	3.86	33,132	33,132	0.00	62,344	1.14
miniYFJS02	27,557	23,100	23,100.0	0.00	24,831	219	1.51	23,100	23,100	0.00	27,069	0.33
miniYFJS03	57,950	42,896	42,896.0	0.00	98,915	1,106	4.78	42,896	42,896	0.00	502,236	8.17
miniYFJS04	24,669	24,485	24,485.0	0.00	58,780	370	3.14	24,485	24,485	0.00	41,642	0.81
miniYFJS05	26,135	23,597	23,597.0	0.00	104,135	634	5.89	23,597	23,597	0.00	42,254	0.76
miniYFJS06	29,080	28,655	28,655.0	0.00	44,226	318	2.80	28,655	28,655	0.00	141,943	2.17
miniYFJS07	47,743	42,239	42,239.0	0.00	37,239	321	1.78	42,239	42,239	0.00	67,444	1.28
miniYFJS08	33,178	31,471	31,471.0	0.00	60,079	724	3.30	31,471	31,471	0.00	111,457	2.03
miniYFJS09	37,593	35,250	35,250.0	0.00	182,095	1,770	10.34	35,250	35,250	0.00	202,187	3.52
miniYFJS10	30,651	26,145	26,145.0	0.00	190,450	1,068	12.01	26,145	26,145	0.00	226,106	3.95
miniYFJS11	47,079	38,545	38,545.0	0.00	374,613	1,492	20.78	38,545	38,545	0.00	107,493	1.56
miniYFJS12	36,036	27,895	27,895.0	0.00	917,827	3,772	166.43	27,895	27,895	0.00	438,457	11.45
miniYFJS13	31,720	28,120	28,120.0	0.00	401,361	1,771	70.14	28,120	28,120	0.00	1,740,741	41.54
miniYFJS14	39,254	29,682	29,682.0	0.00	598,499	4,869	53.98	29,682	29,682	0.00	583,529	8.98
miniYFJS15	51,504	41,619	41,619.0	0.00	55,926	407	5.37	41,619	41,619	0.00	359,587	5.08
miniYFJS16	33,285	31,280	31,280.0	0.00	541,733	1,770	44.44	31,280	31,280	0.00	309,885	6.41
miniYFJS17	48,745	40,388	40,388.0	0.00	323,307	1,758	31.31	40,388	40,388	0.00	833,567	13.39
miniYFJS18	31,911	26,297	26,297.0	0.00	388,013	2,147	19.13	26,297	26,297	0.00	90,325	1.40
miniYFJS19	41,451	30,717	30,716.4	0.00	1,063,227	3,558	97.04	30,717	30,717	0.00	927,002	16.87
miniYFJS20	39,600	28,832	28,832.0	0.00	247,520	1,808	68.05	28,832	28,832	0.00	600,081	12.09
miniYFJS21	40,765	34,811	34,811.0	0.00	4,210,905	14,690	492.45	34,811	34,811	0.00	5,415,595	106.86
miniYFJS22	32,485	31,702	31,702.0	0.00	8,483,765	29,798	1,828.40	31,702	31,702	0.00	33,061,321	955.04
miniYFJS23	49,209	38,639	38,639.0	0.00	20,876,702	38,396	3,569.36	38,639	23,930	38.07	119,735,989	3,600.00
miniYFJS24	39,883	28,884	28,884.0	0.00	739,015	2,875	102.09	28,884	28,884	0.00	582,657	11.28
miniYFJS25	40,933	34,231	34,230.0	0.00	21,582,095	81,909	3,490.98	34,231	34,231	0.00	33,935,628	702.08
miniYFJS26	49,412	47,519	47,519.0	0.00	3,913,717	12,549	473.14	47,519	47,519	0.00	2,100,339	49.56
miniYFJS27	36,561	34,042	34,041.7	0.00	4,103,916	19,748	814.72	34,042	34,042	0.00	1,441,976	39.60
miniYFJS28	40,325	32,080	32,080.0	0.00	1,009,480	3,669	167.51	32,080	32,080	0.00	1,004,092	18.57
miniYFJS29	41,834	36,093	36,093.0	0.00	2,406,169	11,100	400.19	36,093	36,093	0.00	30,525,344	796.84
miniYFJS30	45,051	31,888	31,888.0	0.00	1,142,515	3,913	195.41	31,888	31,888	0.00	8,954,616	185.56
mean	30,719.55	26,717.97	26,629.26	0.39	2,619,468	10,583	431.68	26,704.32	26,331.65	1.20	9,880,257	263.96

Table B6: Solutions found and computational cost of solving the small-scale instances with learning index $\alpha = 0.2$ using CPLEX and CP Optimizer with no warm start.

Instance	EST/ECT	CPLEX						CP Optimizer				
		Makespan	Best Bound	Gap(%)	#iter	#B&B Nodes	CPU	Makespan	Best bound	Gap(%)	#branches	CPU
miniDAFJS01	20,757	19,443	19,443.0	0.00	1,706,660	13,258	91.07	19,443	19,443	0.00	1,368,672	24.80
miniDAFJS02	21,152	18,916	18,916.0	0.00	53,092	696	1.92	18,916	18,916	0.00	54,800	0.84
miniDAFJS03	17,621	17,419	17,419.0	0.00	17,807	254	0.64	17,419	17,419	0.00	24,795	0.31
miniDAFJS04	20,391	18,800	18,800.0	0.00	5,444	141	0.28	18,800	18,800	0.00	23,354	0.33
miniDAFJS05	19,253	17,596	17,596.0	0.00	149,857	1,357	25.75	17,596	17,596	0.00	283,148	4.96
miniDAFJS06	21,148	18,692	18,692.0	0.00	1,036,732	8,974	93.70	18,692	18,692	0.00	621,331	13.26
miniDAFJS07	24,767	24,256	24,256.0	0.00	223,460	2,563	12.25	24,256	24,256	0.00	51,513	0.77
miniDAFJS08	17,927	17,900	17,900.0	0.00	2,439	36	0.11	17,900	17,900	0.00	7,352	0.06
miniDAFJS09	22,398	20,797	20,796.6	0.00	1,511,997	8,333	119.45	20,797	20,797	0.00	2,558,130	51.43
miniDAFJS10	20,176	17,395	17,395.0	0.00	14,964	318	0.83	17,395	17,395	0.00	41,579	0.51
miniDAFJS11	28,718	25,304	25,304.0	0.00	1,300,871	6,954	91.45	25,304	25,304	0.00	540,330	11.16
miniDAFJS12	19,094	17,105	17,105.0	0.00	102,571	2,392	5.46	17,105	17,105	0.00	222,845	3.49
miniDAFJS13	15,425	14,077	14,077.0	0.00	4,468	83	0.18	14,077	14,077	0.00	22,499	0.20
miniDAFJS14	23,552	20,620	20,620.0	0.00	274,434	2,260	33.10	20,620	20,620	0.00	193,962	3.95
miniDAFJS15	20,524	18,625	18,625.0	0.00	108,476	1,319	3.60	18,625	18,625	0.00	210,053	3.22
miniDAFJS16	23,797	22,734	22,734.0	0.00	688,597	4,387	51.27	22,734	22,734	0.00	315,638	5.80
miniDAFJS17	19,941	18,253	18,253.0	0.00	63,437	1,442	2.02	18,253	18,253	0.00	34,487	0.47
miniDAFJS18	17,784	16,495	16,495.0	0.00	69,463	1,219	2.33	16,495	16,495	0.00	51,441	0.67
miniDAFJS19	19,030	18,474	18,474.0	0.00	50,222	395	2.07	18,474	18,474	0.00	59,081	0.95
miniDAFJS20	19,587	18,521	18,521.0	0.00	288,899	2,465	17.67	18,521	18,521	0.00	249,219	4.66
miniDAFJS21	21,299	20,240	16,459.5	18.68	26,343,116	146,565	3,600.00	19,430	11,665	39.96	123,388,382	3,600.00
miniDAFJS22	24,671	22,322	22,322.0	0.00	15,610,815	69,887	2,152.07	22,322	22,322	0.00	22,502,951	562.39
miniDAFJS23	23,847	20,932	17,226.3	17.70	20,429,096	59,127	3,600.00	20,932	11,409	45.49	131,854,569	3,600.00
miniDAFJS24	22,524	20,389	20,388.2	0.00	8,183,476	36,358	972.95	20,389	20,389	0.00	67,098,935	1,754.22
miniDAFJS25	21,923	18,400	18,399.6	0.00	4,384,003	28,822	629.75	18,400	18,400	0.00	1,597,492	35.13
miniDAFJS26	21,613	18,396	18,395.4	0.00	12,778,269	84,472	1,639.90	18,396	18,396	0.00	16,194,486	378.04
miniDAFJS27	23,089	18,501	16,902.1	8.64	22,061,469	96,388	3,600.00	18,501	18,501	0.00	116,791,733	3,275.39
miniDAFJS28	24,429	18,762	18,762.0	0.00	4,953,263	28,751	345.26	18,762	18,762	0.00	1,134,178	24.56
miniDAFJS29	18,560	18,253	18,253.0	0.00	339,168	2,281	21.88	18,253	18,253	0.00	483,842	10.68
miniDAFJS30	21,421	20,328	16,902.5	16.85	23,876,813	61,606	3,600.00	19,504	11,670	40.17	126,192,772	3,600.00
miniYFJS01	33,697	31,008	31,008.0	0.00	143,875	1,615	9.38	31,008	31,008	0.00	78,398	1.38
miniYFJS02	25,969	22,010	22,010.0	0.00	47,592	623	2.25	22,010	22,010	0.00	42,175	0.53
miniYFJS03	52,865	38,935	38,935.0	0.00	48,719	350	2.40	38,935	38,935	0.00	708,282	12.80
miniYFJS04	24,017	23,774	23,774.0	0.00	125,088	2,363	9.63	23,774	23,774	0.00	70,611	1.33
miniYFJS05	24,743	22,843	22,843.0	0.00	166,373	2,320	10.62	22,843	22,843	0.00	60,272	1.22
miniYFJS06	27,366	27,366	27,366.0	0.00	92,146	965	5.51	27,366	27,366	0.00	238,369	3.68
miniYFJS07	44,128	38,932	38,932.0	0.00	37,562	391	1.94	38,932	38,932	0.00	101,984	1.69
miniYFJS08	31,133	29,464	29,464.0	0.00	189,028	2,554	14.87	29,464	29,464	0.00	201,875	3.36
miniYFJS09	34,357	33,763	33,763.0	0.00	478,771	3,515	30.48	33,763	33,763	0.00	202,791	3.71
miniYFJS10	28,513	25,072	25,072.0	0.00	544,968	2,549	41.55	25,072	25,072	0.00	488,425	9.01
miniYFJS11	43,254	36,307	36,307.0	0.00	318,588	2,654	25.52	36,307	36,307	0.00	246,190	4.46
miniYFJS12	33,495	26,219	26,219.0	0.00	887,337	4,393	220.70	26,219	26,219	0.00	2,023,656	48.03
miniYFJS13	26,536	25,619	25,619.0	0.00	156,044	1,210	16.13	25,619	25,619	0.00	1,441,224	33.35
miniYFJS14	37,157	27,428	27,428.0	0.00	765,456	3,252	56.53	27,428	27,428	0.00	1,849,265	32.86
miniYFJS15	46,583	38,256	38,256.0	0.00	92,806	556	6.69	38,256	38,256	0.00	810,033	13.48
miniYFJS16	30,576	29,442	29,442.0	0.00	1,139,925	3,790	121.47	29,442	29,442	0.00	1,249,348	27.21
miniYFJS17	44,916	37,465	37,465.0	0.00	688,837	4,042	89.04	37,465	37,465	0.00	1,117,783	20.11
miniYFJS18	29,764	25,067	25,067.0	0.00	472,503	2,385	30.34	25,067	25,067	0.00	287,975	4.74
miniYFJS19	37,034	29,207	29,207.0	0.00	1,139,515	3,644	84.64	29,207	29,207	0.00	954,474	18.52
miniYFJS20	37,063	27,091	27,091.0	0.00	774,284	4,104	109.33	27,091	27,091	0.00	1,269,293	26.01
miniYFJS21	35,905	32,166	32,165.3	0.00	24,225,142	64,060	3,394.18	32,166	32,166	0.00	24,556,626	469.45
miniYFJS22	30,212	28,985	28,984.6	0.00	9,251,532	43,999	1,695.19	28,985	17,383	40.03	124,846,662	3,600.00
miniYFJS23	45,823	36,940	33,583.8	9.09	23,755,475	55,156	3,600.00	35,441	18,601	47.52	130,348,733	3,600.00
miniYFJS24	37,651	27,023	27,022.6	0.00	1,614,894	6,496	168.83	27,023	27,023	0.00	1,138,650	20.65
miniYFJS25	35,966	32,346	32,346.0	0.00	12,373,904	57,874	1,768.68	32,346	32,346	0.00	104,169,487	2,382.27
miniYFJS26	53,604	43,452	43,452.0	0.00	763,526	4,207	120.49	43,452	43,452	0.00	6,497,674	142.89
miniYFJS27	33,799	31,571	31,571.0	0.00	2,413,599	7,536	538.69	31,571	31,571	0.00	1,546,654	42.83
miniYFJS28	37,547	30,428	30,428.0	0.00	1,848,244	6,076	200.67	30,428	30,428	0.00	1,774,220	35.20
miniYFJS29	39,146	32,826	32,826.0	0.00	9,098,054	37,829	1,656.42	32,826	32,826	0.00	108,920,721	3,296.71
miniYFJS30	40,752	29,848	29,847.0	0.00	19,245,922	57,808	3,129.86	29,848	29,848	0.00	58,043,735	1,333.06
mean	28,666.48	24,979.97	24,715.44	1.18	4,325,551	17,690	631.32	24,927.75	24,035.02	3.55	19,824,319	536.05

Table B7: Solutions found and computational cost of solving the small-scale instances with learning index $\alpha = 0.3$ using CPLEX and CP Optimizer with no warm start.

Instance	EST/ECT	CPLEX						CP Optimizer				
		Makespan	Best Bound	Gap(%)	#iter	#B&B Nodes	CPU	Makespan	Best bound	Gap(%)	#branches	CPU
DAFJS01	29,769	24,493	21,111.3	13.81	8,781,936	18,717	3,600.00	23,633	18,602	21.29	84,299,330	3,600.00
DAFJS02	32,467	27,446	23,671.1	13.75	4,871,827	17,791	3,600.00	26,869	18,533	31.02	108,672,879	3,600.00
DAFJS03	53,688							51,097	41,271	19.23	5,386,744	3,600.06
DAFJS04	54,082	131,061	45,327.6	65.41	341,806	136	3,600.00	52,731	44,062	16.44	12,373,138	3,600.04
DAFJS05	45,862	140,937	30,804.0	78.14	1,006,477	2,426	3,600.00	36,377	23,378	35.73	31,390,900	3,600.00
DAFJS06	51,925	169,875	31,921.8	81.21	494,084	533	3,600.00	36,445	23,364	35.89	13,767,734	3,600.03
DAFJS07	57,193							50,032	33,601	32.84	1,761,156	3,600.14
DAFJS08	62,159							57,771	43,331	25.00	3,237,105	3,600.04
DAFJS09	48,680	192,570	36,379.4	81.11	451,811	880	3,600.00	41,278	22,460	45.59	11,458,085	3,600.04
DAFJS10	58,695	191,176	39,597.3	79.29	169,982	11	3,600.00	44,459	23,440	47.28	9,562,211	3,600.08
DAFJS11	67,594							59,414	44,061	25.84	2,741,100	3,600.09
DAFJS12	68,322							61,544	35,263	42.70	1,053,680	3,600.82
DAFJS13	61,794	244,658	50,246.7	79.46	118,193	43	3,600.00	55,604	20,981	62.27	8,846,248	3,600.15
DAFJS14	76,862	249,678	55,318.8	77.84	123,253	35	3,600.00	62,795	23,996	61.79	4,691,949	3,600.02
DAFJS15	72,288							68,881	33,814	50.91	875,435	3,600.19
DAFJS16	76,204							69,710	42,465	39.08	974,906	3,600.20
DAFJS17	73,219	280,020	57,432.0	79.49	88,004	122	3,600.00	68,369	20,260	70.37	4,118,413	3,600.06
DAFJS18	78,862	297,443	58,268.2	80.41	76,754	45	3,600.00	68,079	21,838	67.92	3,771,659	3,600.06
DAFJS19	66,412	303,143	36,125.6	88.08	78,253	5	3,600.00	49,318	35,267	28.49	2,823,244	3,600.11
DAFJS20	78,778							64,164	29,175	54.53	2,045,264	3,600.07
DAFJS21	78,320							75,419	33,405	55.71	1,206,198	3,600.13
DAFJS22	70,892							70,542	30,309	57.03	1,221,034	3,600.16
DAFJS23	49,969	289,291	31,559.4	89.09	64,875	2	3,600.00	46,006	31,047	32.52	2,031,883	3,600.05
DAFJS24	57,411	373,414	36,660.8	90.18	63,446	0	3,600.00	52,931	32,194	39.18	1,713,420	3,600.11
DAFJS25	82,055							80,504	38,155	52.60	802,401	3,600.28
DAFJS26	81,480							76,796	36,963	51.87	682,457	3,600.30
DAFJS27	81,470							83,925	32,947	60.74	709,483	3,600.28
DAFJS28	62,568							51,156	36,392	28.86	2,178,284	3,600.07
DAFJS29	72,938							63,660	40,983	35.62	1,066,928	3,600.12
DAFJS30	61,147							53,221	31,515	40.78	1,802,511	3,600.20
YFJS01	87,203	184,164	58,146.2	68.43	3,063,321	9,717	3,600.00	69,403	53,512	22.90	50,998,093	3,600.00
YFJS02	81,579	191,281	65,922.1	65.54	5,550,478	10,720	3,600.00	72,763	61,321	15.72	78,743,815	3,600.00
YFJS03	40,197	32,538	32,538.0	0.00	1,195,848	6,780	209.18	32,538	32,538	0.00	1,421,509	24.83
YFJS04	47,467	35,883	35,883.0	0.00	7,984,013	32,396	2,972.90	35,883	35,883	0.00	48,950,271	1,074.70
YFJS05	46,138	41,522	37,703.0	9.20	11,674,681	41,800	3,600.00	41,034	32,158	21.63	155,914,078	3,600.00
YFJS06	52,487	44,351	35,400.1	20.18	6,999,899	19,936	3,600.00	41,644	25,961	37.66	101,348,083	3,600.00
YFJS07	54,457	52,323	35,942.9	31.31	6,558,365	20,349	3,600.00	41,394	32,925	20.46	112,824,997	3,600.00
YFJS08	49,626	32,573	32,573.0	0.00	3,051,237	7,177	971.36	32,573	32,573	0.00	1,245,413	23.40
YFJS09	28,354	40,126	17,754.6	55.75	764,751	1,076	3,600.00	22,681	15,159	33.16	46,816,726	3,600.00
YFJS10	42,583	39,092	31,349.0	19.81	11,978,526	22,942	3,600.00	37,372	37,372	0.00	53,123,788	1,270.38
YFJS11	59,356	62,967	43,587.9	30.78	2,633,022	6,649	3,600.00	49,156	39,044	20.57	86,805,695	3,600.00
YFJS12	70,830	106,077	39,627.1	62.64	2,803,547	8,213	3,600.00	48,556	33,937	30.11	83,825,600	3,600.00
YFJS13	50,805	246,392	30,402.2	87.66	1,096,420	1,897	3,600.00	36,926	28,036	24.08	95,769,393	3,600.00
YFJS14	129,428							115,656	94,991	17.87	10,763,556	3,600.09
YFJS15	138,196							113,665	89,818	20.98	9,621,369	3,600.12
YFJS16	127,055							121,406	86,618	28.65	6,764,003	3,600.11
YFJS17	109,112							154,434	76,416	50.52	925,151	3,600.24
YFJS18	133,703							142,632	81,982	42.52	1,161,476	3,600.32
YFJS19	107,055							168,667	62,129	63.16	957,747	3,600.27
YFJS20	97,868							151,902	65,048	57.18	900,652	3,600.39
mean	69,332.08	149,055.33	38,935.30	53.65	3,040,178	8,533	3,353.83	64,660.30	39,209.86	35.53	25,522,944	3,359.97

Table B8: Solutions found and computational cost of solving the large-scale instances with learning index $\alpha = 0.1$ using CPLEX and CP Optimizer with no warm start.

Instance	EST/ECT	CPLEX						CP Optimizer				
		Makespan	Best Bound	Gap(%)	#iter	#B&B Nodes	CPU	Makespan	Best bound	Gap(%)	#branches	CPU
DAFJS01	28,920	23,373	17,450.6	25.34	8,562,041	30,260	3,600.00	21,913	14,141	35.47	81,829,489	3,600.00
DAFJS02	28,154	31,131	20,402.0	34.46	5,900,727	10,765	3,600.00	24,463	13,966	42.91	99,525,592	3,600.00
DAFJS03	48,555							46,115	29,519	35.99	4,572,229	3,600.08
DAFJS04	48,461	120,313	33,881.5	71.84	377,344	223	3,600.00	45,528	32,038	29.63	13,662,597	3,600.05
DAFJS05	44,790	120,692	25,530.6	78.85	417,139	776	3,600.00	31,933	17,025	46.69	37,866,349	3,600.01
DAFJS06	44,228	140,367	25,465.5	81.86	392,713	300	3,600.00	31,342	16,745	46.57	13,423,397	3,600.04
DAFJS07	50,802	294,164	23,994.1	91.84	68,214	0	3,600.00	45,829	23,042	49.72	1,576,917	3,600.08
DAFJS08	53,439							47,798	29,902	37.44	2,457,489	3,600.06
DAFJS09	42,537	145,250	28,978.9	80.05	396,646	165	3,600.00	36,311	16,016	55.89	12,151,354	3,600.05
DAFJS10	46,257	156,114	30,054.2	80.75	234,482	100	3,600.00	38,697	16,354	57.74	8,071,265	3,600.02
DAFJS11	56,847							53,644	29,499	45.01	1,458,110	3,600.15
DAFJS12	55,575							58,139	23,462	59.65	861,239	3,600.17
DAFJS13	51,317	196,510	39,222.2	80.04	145,744	71	3,600.00	46,036	14,433	68.65	6,842,372	3,600.11
DAFJS14	62,724	198,907	43,933.3	77.91	84,553	25	3,600.00	53,263	16,518	68.99	4,586,006	3,600.03
DAFJS15	54,353							59,303	22,324	62.36	1,194,814	3,600.28
DAFJS16	65,550							60,585	28,167	53.51	1,050,613	3,600.23
DAFJS17	63,177	220,059	42,564.3	80.66	55,859	11	3,600.00	57,654	13,683	76.27	3,232,160	3,600.09
DAFJS18	61,831	235,050	45,689.7	80.56	47,310	27	3,600.00	56,720	14,813	73.88	3,388,488	3,600.04
DAFJS19	58,001	240,355	27,775.1	88.44	74,729	52	3,600.00	41,446	24,250	41.49	3,080,620	3,600.09
DAFJS20	63,588							53,891	19,568	63.69	2,032,692	3,600.07
DAFJS21	63,202							65,384	22,142	66.14	1,048,490	3,600.14
DAFJS22	56,115							60,257	19,800	67.14	985,444	3,600.22
DAFJS23	47,616	233,720	23,925.4	89.76	61,210	11	3,600.00	40,344	21,422	46.90	2,650,502	3,600.06
DAFJS24	49,019							45,424	21,763	52.09	1,978,045	3,600.10
DAFJS25	66,262							78,761	24,843	68.46	871,563	3,600.30
DAFJS26	72,226							71,373	24,182	66.12	724,046	3,600.32
DAFJS27	64,797							76,421	21,584	71.76	807,436	3,600.35
DAFJS28	52,639							47,129	24,712	47.57	1,899,553	3,600.09
DAFJS29	59,792							58,033	27,497	52.62	1,366,537	3,600.13
DAFJS30	55,015							49,441	21,222	57.08	1,539,764	3,600.15
YFJS01	84,152	229,124	48,427.6	78.86	1,494,123	3,334	3,600.00	63,055	39,902	36.72	40,484,030	3,600.00
YFJS02	66,853	104,886	56,637.0	46.00	5,311,023	15,016	3,600.00	66,853	46,715	30.12	87,985,152	3,600.00
YFJS03	35,380	30,073	30,073.0	0.00	1,963,718	8,908	266.97	30,073	30,073	0.00	1,621,891	28.56
YFJS04	43,614	32,670	32,669.9	0.00	14,982,850	48,869	3,325.16	32,670	32,670	0.00	151,680,918	3,531.40
YFJS05	40,851	39,647	32,101.7	19.03	6,958,890	25,300	3,600.00	37,682	25,093	33.41	146,200,511	3,600.00
YFJS06	48,564	43,728	31,087.9	28.91	7,191,534	23,158	3,600.00	38,456	19,541	49.19	107,402,514	3,600.00
YFJS07	51,261	42,747	32,996.0	22.81	7,884,139	22,737	3,600.00	37,159	26,063	29.86	118,578,621	3,600.00
YFJS08	44,074	30,184	30,184.0	0.00	15,326,236	30,527	2,563.80	30,184	30,184	0.00	8,412,375	161.96
YFJS09	26,072	34,653	15,072.2	56.51	813,927	1,607	3,600.00	21,363	11,324	46.99	31,560,620	3,600.01
YFJS10	40,658	36,223	29,456.5	18.68	12,699,118	21,740	3,600.00	34,378	22,371	34.93	145,397,101	3,600.00
YFJS11	51,469	78,286	35,486.7	54.67	2,906,338	6,972	3,600.00	44,294	30,073	32.11	91,646,730	3,600.00
YFJS12	59,898	169,068	31,482.6	81.38	1,774,627	3,612	3,600.00	43,656	26,939	38.29	71,468,473	3,600.00
YFJS13	45,084	158,251	24,895.5	84.27	1,945,690	3,709	3,600.00	34,178	21,361	37.50	78,743,496	3,600.00
YFJS14	109,469							105,170	68,423	34.94	7,745,065	3,600.12
YFJS15	112,424							116,282	64,923	44.17	6,969,354	3,600.03
YFJS16	106,855							122,041	62,865	48.49	6,765,937	3,600.03
YFJS17	85,736							123,064	51,540	58.12	1,085,511	3,600.35
YFJS18	99,429							162,427	55,101	66.08	734,894	3,600.31
YFJS19	89,561							141,367	41,690	70.51	1,259,425	3,600.27
YFJS20	91,958							157,878	43,711	72.31	1,347,634	3,600.30
mean	58,983.02	125,390.56	31,831.04	56.80	3,632,256	9,566	3,427.00	58,908.14	27,903.88	48.22	28,476,508	3,458.54

Table B9: Solutions found and computational cost of solving the large-scale instances with learning index $\alpha = 0.2$ using CPLEX and CP Optimizer with no warm start.

Instance	EST/ECT	CPLEX						CP Optimizer				
		Makespan	Best Bound	Gap(%)	#iter	#B&B Nodes	CPU	Makespan	Best bound	Gap(%)	#branches	CPU
DAFJS01	22,616	31,250	15,899.2	49.12	5,151,334	20,990	3,600.00	20,017	10,711	46.49	73,337,941	3,600.00
DAFJS02	26,183	26,935	16,860.9	37.40	4,028,549	13,918	3,600.00	22,464	10,500	53.26	85,220,203	3,600.00
DAFJS03	43,964	165,343	22,536.2	86.37	152,014	7	3,600.00	41,680	21,152	49.25	5,341,688	3,600.08
DAFJS04	43,281	120,636	24,891.1	79.37	303,358	100	3,600.00	40,763	23,278	42.89	10,043,152	3,600.06
DAFJS05	38,237	104,599	20,630.8	80.28	856,284	1,385	3,600.00	27,851	12,401	55.47	32,427,307	3,600.01
DAFJS06	38,007	101,490	20,177.4	80.12	422,183	490	3,600.00	27,311	12,016	56.00	12,872,596	3,600.08
DAFJS07	41,285	243,203	16,904.5	93.05	63,397	2	3,600.00	40,867	15,801	61.34	1,582,359	3,600.09
DAFJS08	46,323	249,435	21,333.6	91.45	91,149	0	3,600.00	44,108	20,633	53.22	1,968,937	3,600.05
DAFJS09	41,418	128,596	22,202.1	82.73	490,560	253	3,600.00	31,352	11,423	63.57	11,660,834	3,600.06
DAFJS10	38,771	125,666	22,505.3	82.09	185,446	51	3,600.00	33,898	11,408	66.35	6,741,671	3,600.01
DAFJS11	47,805							47,112	19,709	58.17	1,963,366	3,600.16
DAFJS12	45,735							50,736	15,611	69.23	869,141	3,600.22
DAFJS13	42,319	162,727	29,196.3	82.06	118,430	14	3,600.00	38,636	9,929	74.30	4,586,122	3,600.02
DAFJS14	49,459	160,464	32,943.1	79.47	94,227	49	3,600.00	44,412	11,318	74.52	6,262,270	3,600.02
DAFJS15	49,206							59,111	14,671	75.18	935,778	3,600.26
DAFJS16	56,727							53,663	18,661	65.23	844,962	3,600.46
DAFJS17	54,527	174,560	29,949.1	82.84	50,892	22	3,600.00	46,210	9,240	80.00	2,507,750	3,600.12
DAFJS18	51,823	188,164	32,363.0	82.80	56,426	21	3,600.00	47,911	10,047	79.03	4,029,469	3,600.10
DAFJS19	46,880	174,022	21,110.1	87.87	77,653	38	3,600.00	38,047	16,675	56.17	2,979,111	3,600.03
DAFJS20	51,686	217,235	25,800.6	88.12	59,728	6	3,600.00	48,092	13,124	72.71	1,705,921	3,600.08
DAFJS21	53,010							54,154	14,664	72.92	1,395,006	3,600.46
DAFJS22	45,005							52,654	12,934	75.44	1,064,185	3,601.03
DAFJS23	39,249	191,695	18,149.8	90.53	59,605	34	3,600.00	36,571	14,782	59.58	2,006,642	3,600.05
DAFJS24	44,851							44,069	14,688	66.67	1,062,880	3,600.12
DAFJS25	51,964							74,172	16,174	78.19	1,092,932	3,600.36
DAFJS26	57,182							70,062	15,820	77.42	974,305	3,600.97
DAFJS27	58,138							76,530	14,138	81.53	773,314	3,600.40
DAFJS28	42,898	246,790	18,542.3	92.49	65,296	0	3,600.00	40,469	16,781	58.53	1,902,640	3,600.32
DAFJS29	51,606							47,216	18,452	60.92	1,362,656	3,600.16
DAFJS30	43,859							44,092	14,275	67.62	1,210,195	3,600.08
YFJS01	70,402	192,220	38,607.3	79.92	1,893,481	3,507	3,600.00	58,140	29,808	48.73	38,558,149	3,600.00
YFJS02	61,111	135,858	47,508.9	65.03	2,730,203	6,927	3,600.00	59,001	35,588	39.68	70,693,013	3,600.00
YFJS03	31,680	27,686	27,686.0	0.00	1,251,566	4,381	309.01	27,686	27,686	0.00	889,816	17.74
YFJS04	38,898	30,112	27,527.3	8.58	19,439,650	70,782	3,600.00	29,692	18,161	38.84	135,837,875	3,600.00
YFJS05	41,586	35,708	27,881.3	21.92	13,621,378	49,900	3,600.00	34,430	19,314	43.90	127,849,686	3,600.00
YFJS06	46,811	41,805	26,558.7	36.47	4,642,087	14,000	3,600.00	34,158	14,714	56.92	81,411,905	3,600.00
YFJS07	39,896	37,401	25,921.0	30.69	9,470,197	15,391	3,600.00	33,059	20,397	38.30	99,127,090	3,600.00
YFJS08	39,325	28,192	28,192.0	0.00	13,667,243	32,817	2,269.32	28,192	28,192	0.00	75,897,229	1,616.11
YFJS09	24,027	58,421	12,129.3	79.24	714,288	535	3,600.00	20,220	9,525	52.89	28,005,321	3,600.01
YFJS10	39,607	34,020	26,634.0	21.71	14,208,216	46,656	3,600.00	32,159	17,808	44.63	108,596,217	3,600.00
YFJS11	45,051	49,595	32,834.8	33.79	4,868,955	16,076	3,600.00	40,319	23,170	42.53	82,198,235	3,600.00
YFJS12	52,398	187,516	25,930.7	86.17	1,853,113	4,044	3,600.00	38,480	21,335	44.56	61,413,112	3,600.00
YFJS13	40,119	127,522	21,700.9	82.98	3,512,730	6,809	3,600.00	31,486	16,668	47.06	57,270,529	3,600.00
YFJS14	92,457							99,589	49,252	50.54	7,476,196	3,600.03
YFJS15	96,547							93,789	46,947	49.94	7,959,592	3,600.04
YFJS16	92,811							92,080	45,648	50.43	7,418,259	3,600.03
YFJS17	73,682							152,448	34,763	77.20	1,196,752	3,600.31
YFJS18	87,059							128,753	37,042	71.23	937,136	3,600.24
YFJS19	74,431							102,607	27,976	72.73	1,236,409	3,600.28
YFJS20	72,481							126,310	29,376	76.74	1,572,668	3,600.24
mean	50,487.86	122,544.06	25,197.02	64.34	3,361,279	9,974	3,450.91	52,136.56	19,887.72	57.96	25,525,410	3,488.82

Table B10: Solutions found and computational cost of solving the large-scale instances with learning index $\alpha = 0.3$ using CPLEX and CP Optimizer with no warm start.

Instance	EST/ECT	CPLEX						CP Optimizer				
		Makespan	Best Bound	Gap(%)	#iter	#B&B Nodes	CPU	Makespan	Best bound	Gap(%)	#branches	CPU
miniDAFJS01	23,264	22,875	22,875.0	0.00	27,130	272	2.11	22,875	22,875	0.00	161,470	3.33
miniDAFJS02	23,242	22,708	22,708.0	0.00	9,960	85	0.55	22,708	22,708	0.00	42,281	0.66
miniDAFJS03	18,363	18,363	18,363.0	0.00	0	0	0.01	18,363	18,363	0.00	5,384	0.07
miniDAFJS04	21,690	20,498	20,498.0	0.00	155	0	0.03	20,498	20,498	0.00	14,007	0.12
miniDAFJS05	22,598	20,593	20,593.0	0.00	27,452	195	1.55	20,593	20,593	0.00	117,284	1.70
miniDAFJS06	23,370	22,867	22,867.0	0.00	4,996	28	0.51	22,867	22,867	0.00	54,279	1.34
miniDAFJS07	28,644	25,715	25,715.0	0.00	368	3	0.06	25,715	25,715	0.00	29,563	0.27
miniDAFJS08	19,878	19,878	19,878.0	0.00	0	0	0.00	19,878	19,878	0.00	2,937	0.03
miniDAFJS09	25,425	24,267	24,267.0	0.00	81,679	379	5.18	24,267	24,267	0.00	257,703	5.36
miniDAFJS10	21,563	20,336	20,336.0	0.00	725	0	0.07	20,336	20,336	0.00	35,476	0.28
miniDAFJS11	33,911	29,968	29,967.4	0.00	199,078	963	9.22	29,968	29,968	0.00	163,527	2.91
miniDAFJS12	20,342	18,670	18,670.0	0.00	2,687	48	0.20	18,670	18,670	0.00	19,537	0.33
miniDAFJS13	18,313	16,313	16,313.0	0.00	225	0	0.02	16,313	16,313	0.00	3,470	0.04
miniDAFJS14	26,503	23,140	23,140.0	0.00	8,076	76	0.62	23,140	23,140	0.00	28,105	0.53
miniDAFJS15	22,472	21,715	21,715.0	0.00	5,253	86	0.35	21,715	21,715	0.00	39,595	0.61
miniDAFJS16	25,691	25,426	25,426.0	0.00	4,790	40	0.59	25,426	25,426	0.00	37,821	0.69
miniDAFJS17	21,070	20,155	20,155.0	0.00	126	0	0.02	20,155	20,155	0.00	12,783	0.14
miniDAFJS18	19,308	18,135	18,135.0	0.00	525	0	0.05	18,135	18,135	0.00	45,244	0.38
miniDAFJS19	21,293	20,945	20,945.0	0.00	54	0	0.04	20,945	20,945	0.00	11,785	0.24
miniDAFJS20	23,570	21,838	21,838.0	0.00	9,612	81	0.56	21,838	21,838	0.00	88,726	1.22
miniDAFJS21	25,020	23,344	23,343.2	0.00	3,271,549	13,944	405.89	23,344	23,344	0.00	9,663,552	220.19
miniDAFJS22	28,122	25,923	25,923.0	0.00	1,458,988	5,824	185.31	25,923	25,923	0.00	977,268	23.31
miniDAFJS23	27,566	24,038	24,038.0	0.00	7,249,423	29,201	1,100.45	24,038	24,038	0.00	19,287,619	437.58
miniDAFJS24	26,932	24,579	24,579.0	0.00	482,880	2,202	59.12	24,579	24,579	0.00	685,027	15.76
miniDAFJS25	23,370	21,143	21,143.0	0.00	205,001	1,097	17.97	21,143	21,143	0.00	176,931	4.21
miniDAFJS26	25,974	21,120	21,120.0	0.00	678,221	3,053	72.44	21,120	21,120	0.00	611,055	12.03
miniDAFJS27	28,084	22,050	22,050.0	0.00	1,879,171	8,143	293.83	22,050	22,050	0.00	1,508,081	37.57
miniDAFJS28	28,836	22,708	22,707.2	0.00	1,861,961	13,443	149.02	22,708	22,708	0.00	398,994	6.96
miniDAFJS29	21,151	20,278	20,278.0	0.00	7,016	36	0.54	20,278	20,278	0.00	34,555	0.80
miniDAFJS30	27,249	23,558	23,558.0	0.00	2,906,092	9,291	448.41	23,558	23,558	0.00	11,821,323	309.42
miniYFJS01	35,243	35,046	35,046.0	0.00	103	0	0.07	35,046	35,046	0.00	15,101	0.29
miniYFJS02	28,688	24,359	24,359.0	0.00	3,241	50	0.22	24,359	24,359	0.00	29,373	0.25
miniYFJS03	62,704	47,391	47,391.0	0.00	18,165	76	1.17	47,391	47,391	0.00	149,216	2.56
miniYFJS04	25,394	25,394	25,394.0	0.00	0	0	0.04	25,394	25,394	0.00	12,214	0.22
miniYFJS05	27,650	23,985	23,985.0	0.00	6,849	149	0.42	23,985	23,985	0.00	36,189	0.51
miniYFJS06	30,952	29,469	29,469.0	0.00	6,924	85	0.40	29,469	29,469	0.00	85,570	1.34
miniYFJS07	51,782	45,705	45,705.0	0.00	9,538	46	0.65	45,705	45,705	0.00	55,813	1.08
miniYFJS08	35,336	33,829	33,829.0	0.00	25,821	97	1.26	33,829	33,829	0.00	55,917	0.95
miniYFJS09	38,626	37,049	37,049.0	0.00	18,799	72	1.22	37,049	37,049	0.00	35,769	0.74
miniYFJS10	32,975	27,310	27,310.0	0.00	57,031	322	3.79	27,310	27,310	0.00	67,671	1.32
miniYFJS11	51,212	41,300	41,300.0	0.00	12,019	39	1.22	41,300	41,300	0.00	74,802	1.31
miniYFJS12	38,889	30,145	30,145.0	0.00	258,506	2,268	50.34	30,145	30,145	0.00	182,826	4.60
miniYFJS13	35,177	30,962	30,962.0	0.00	102,796	961	10.50	30,962	30,962	0.00	467,877	10.83
miniYFJS14	41,560	31,398	31,398.0	0.00	183,367	1,750	10.82	31,398	31,398	0.00	189,572	3.19
miniYFJS15	57,017	45,442	45,442.0	0.00	82,620	330	4.15	45,442	45,442	0.00	105,496	1.85
miniYFJS16	35,831	33,791	33,791.0	0.00	54,694	430	5.36	33,791	33,791	0.00	148,894	2.83
miniYFJS17	53,052	42,838	42,838.0	0.00	179,623	955	10.31	42,838	42,838	0.00	289,190	4.60
miniYFJS18	37,217	28,247	28,247.0	0.00	42,054	291	3.52	28,247	28,247	0.00	43,942	0.60
miniYFJS19	46,478	33,601	33,601.0	0.00	290,798	2,054	18.84	33,601	33,601	0.00	279,742	5.06
miniYFJS20	42,408	30,837	30,837.0	0.00	130,053	955	12.73	30,837	30,837	0.00	254,603	4.71
miniYFJS21	44,016	37,096	37,096.0	0.00	1,036,679	3,506	131.13	37,096	37,096	0.00	1,366,812	25.93
miniYFJS22	43,722	34,282	34,282.0	0.00	383,245	2,448	69.67	34,282	34,282	0.00	2,040,579	52.57
miniYFJS23	53,046	42,079	42,079.0	0.00	1,420,031	4,353	212.11	42,079	42,079	0.00	12,642,435	284.93
miniYFJS24	42,268	30,905	30,905.0	0.00	508,302	2,030	54.71	30,905	30,905	0.00	1,573,618	21.90
miniYFJS25	46,936	36,170	36,170.0	0.00	1,222,188	4,762	204.91	36,170	36,170	0.00	6,936,740	146.35
miniYFJS26	55,022	51,466	51,466.0	0.00	174,383	1,754	18.89	51,466	51,466	0.00	1,685,393	38.15
miniYFJS27	41,060	36,719	36,719.0	0.00	1,046,266	4,945	204.41	36,719	36,719	0.00	1,137,659	29.22
miniYFJS28	43,232	34,509	34,509.0	0.00	355,782	2,272	43.76	34,509	34,509	0.00	344,255	6.02
miniYFJS29	48,806	39,798	39,798.0	0.00	763,339	2,722	154.77	39,798	39,798	0.00	4,215,665	105.92
miniYFJS30	49,458	33,974	33,974.0	0.00	1,209,837	3,517	219.24	33,974	33,974	0.00	728,205	16.17
mean	33,309.52	28,637.37	28,637.33	0.00	499,771	2,195	70.09	28,637.37	28,637.37	0.00	1,359,775	31.07

Table B11: Solutions found and computational cost of solving the small-scale instances with learning index $\alpha = 0.1$ using CPLEX and CP Optimizer with warm start.

Instance	EST/ECT	CPLEX						CP Optimizer				
		Makespan	Best Bound	Gap(%)	#iter	#B&B Nodes	CPU	Makespan	Best bound	Gap(%)	#branches	CPU
miniDAFJS01	22,156	21,327	21,327.0	0.00	539,475	4,331	32.03	21,327	21,327	0.00	828,753	15.42
miniDAFJS02	22,161	20,635	20,635.0	0.00	30,346	207	1.27	20,635	20,635	0.00	37,188	0.62
miniDAFJS03	17,972	17,972	17,972.0	0.00	561	0	0.05	17,972	17,972	0.00	9,077	0.14
miniDAFJS04	21,121	19,602	19,602.0	0.00	655	0	0.10	19,602	19,602	0.00	30,291	0.23
miniDAFJS05	20,826	18,803	18,803.0	0.00	58,235	514	2.78	18,803	18,803	0.00	122,294	2.09
miniDAFJS06	22,193	20,568	20,568.0	0.00	100,122	455	6.63	20,568	20,568	0.00	458,139	8.80
miniDAFJS07	25,088	24,715	24,715.0	0.00	949	0	0.08	24,715	24,715	0.00	17,208	0.26
miniDAFJS08	18,857	18,857	18,857.0	0.00	0	0		18,857	18,857	0.00	1,628	0.02
miniDAFJS09	23,830	22,660	22,659.0	0.00	186,037	926	10.72	22,660	22,660	0.00	1,750,735	32.58
miniDAFJS10	21,021	18,823	18,823.0	0.00	5,108	109	0.37	18,823	18,823	0.00	36,151	0.40
miniDAFJS11	31,269	27,455	27,455.0	0.00	300,137	1,078	16.04	27,455	27,455	0.00	303,517	5.66
miniDAFJS12	19,624	17,874	17,874.0	0.00	12,831	131	1.04	17,874	17,874	0.00	70,280	1.35
miniDAFJS13	17,143	15,143	15,143.0	0.00	2,141	55	0.11	15,143	15,143	0.00	28,490	0.18
miniDAFJS14	24,958	21,817	21,817.0	0.00	70,714	493	4.31	21,817	21,817	0.00	47,183	0.98
miniDAFJS15	21,969	20,236	20,236.0	0.00	37,460	459	1.34	20,236	20,236	0.00	118,277	1.80
miniDAFJS16	24,593	24,114	24,114.0	0.00	64,350	482	3.31	24,114	24,114	0.00	158,860	3.13
miniDAFJS17	20,519	19,145	19,145.0	0.00	829	0	0.15	19,145	19,145	0.00	27,187	0.30
miniDAFJS18	18,829	17,270	17,270.0	0.00	1,065	0	0.22	17,270	17,270	0.00	22,589	0.33
miniDAFJS19	20,107	19,642	19,642.0	0.00	1,547	0	0.18	19,642	19,642	0.00	21,729	0.44
miniDAFJS20	21,286	20,086	20,086.0	0.00	29,709	216	1.82	20,086	20,086	0.00	85,561	1.65
miniDAFJS21	22,689	21,352	21,352.0	0.00	3,482,214	16,116	450.50	21,352	21,352	0.00	36,625,027	865.34
miniDAFJS22	26,289	23,852	23,852.0	0.00	2,278,871	8,970	334.17	23,852	23,852	0.00	4,055,406	101.80
miniDAFJS23	25,592	22,695	21,572.0	4.95	21,578,150	82,101	3,600.00	22,390	14,739	34.17	145,855,405	3,600.00
miniDAFJS24	24,500	22,521	22,521.0	0.00	867,832	3,978	105.04	22,521	22,521	0.00	27,825,339	626.15
miniDAFJS25	22,613	19,809	19,809.0	0.00	1,379,354	6,519	143.08	19,809	19,809	0.00	335,391	7.92
miniDAFJS26	23,630	19,724	19,724.0	0.00	1,032,993	5,282	125.28	19,724	19,724	0.00	2,969,433	71.81
miniDAFJS27	25,429	20,245	20,244.2	0.00	18,060,295	57,086	2,444.81	20,245	20,245	0.00	19,311,020	511.38
miniDAFJS28	26,587	20,635	20,635.0	0.00	1,901,205	7,073	120.03	20,635	20,635	0.00	454,841	9.28
miniDAFJS29	19,789	19,201	19,201.0	0.00	31,346	132	2.78	19,201	19,201	0.00	202,976	4.01
miniDAFJS30	24,491	21,552	21,552.0	0.00	7,324,212	28,602	1,133.00	21,552	21,552	0.00	122,811,197	3,295.75
miniYFJS01	34,443	33,132	33,132.0	0.00	3,140	10	0.30	33,132	33,132	0.00	45,548	0.73
miniYFJS02	27,557	23,100	23,100.0	0.00	7,994	68	0.60	23,100	23,100	0.00	46,474	0.56
miniYFJS03	57,950	42,896	42,896.0	0.00	30,087	187	1.84	42,896	42,896	0.00	431,301	7.21
miniYFJS04	24,669	24,485	24,485.0	0.00	1,988	0	0.23	24,485	24,485	0.00	34,380	0.69
miniYFJS05	26,135	23,597	23,597.0	0.00	6,272	130	0.45	23,597	23,597	0.00	29,785	0.65
miniYFJS06	29,080	28,655	28,655.0	0.00	14,327	71	0.95	28,655	28,655	0.00	103,180	1.96
miniYFJS07	47,743	42,239	42,239.0	0.00	13,236	256	1.05	42,239	42,239	0.00	64,640	1.12
miniYFJS08	33,178	31,471	31,471.0	0.00	105,081	743	6.96	31,471	31,471	0.00	201,948	2.52
miniYFJS09	37,593	35,250	35,250.0	0.00	57,100	909	3.92	35,250	35,250	0.00	170,531	2.71
miniYFJS10	30,651	26,145	26,145.0	0.00	77,158	308	5.36	26,145	26,145	0.00	243,120	4.33
miniYFJS11	47,079	38,545	38,545.0	0.00	24,787	119	2.23	38,545	38,545	0.00	116,166	1.89
miniYFJS12	36,036	27,895	27,895.0	0.00	869,409	4,180	126.16	27,895	27,895	0.00	1,020,471	23.22
miniYFJS13	31,720	28,120	28,120.0	0.00	66,786	456	7.91	28,120	28,120	0.00	1,455,458	29.64
miniYFJS14	39,254	29,682	29,682.0	0.00	430,434	3,103	24.52	29,682	29,682	0.00	869,839	13.47
miniYFJS15	51,504	41,619	41,619.0	0.00	59,634	221	3.41	41,619	41,619	0.00	416,969	6.20
miniYFJS16	33,285	31,280	31,280.0	0.00	93,638	810	9.73	31,280	31,280	0.00	441,264	8.77
miniYFJS17	48,745	40,388	40,388.0	0.00	327,916	2,106	29.47	40,388	40,388	0.00	1,109,528	17.20
miniYFJS18	31,911	26,297	26,297.0	0.00	88,571	326	4.96	26,297	26,297	0.00	99,857	1.79
miniYFJS19	41,451	30,717	30,717.0	0.00	835,718	3,653	112.09	30,717	30,717	0.00	864,107	15.46
miniYFJS20	39,600	28,832	28,832.0	0.00	433,320	2,419	70.26	28,832	28,832	0.00	184,894	4.16
miniYFJS21	40,765	34,811	34,811.0	0.00	3,479,276	12,599	420.37	34,811	34,811	0.00	3,982,189	78.92
miniYFJS22	32,485	31,702	31,702.0	0.00	1,605,541	5,599	315.73	31,702	31,702	0.00	26,868,527	691.85
miniYFJS23	49,209	38,639	38,638.6	0.00	10,902,431	28,364	1,767.21	38,639	23,930	38.07	136,924,000	3,600.00
miniYFJS24	39,883	28,884	28,884.0	0.00	739,701	2,619	96.17	28,884	28,884	0.00	1,890,082	29.96
miniYFJS25	40,933	34,231	34,231.0	0.00	6,115,589	22,332	970.00	34,231	34,231	0.00	22,625,728	487.04
miniYFJS26	49,412	47,519	47,519.0	0.00	465,569	3,359	77.27	47,519	47,519	0.00	4,008,598	98.05
miniYFJS27	36,561	34,042	34,042.0	0.00	516,212	1,856	117.25	34,042	34,042	0.00	883,166	27.03
miniYFJS28	40,325	32,080	32,080.0	0.00	413,378	1,225	31.61	32,080	32,080	0.00	672,862	13.21
miniYFJS29	41,834	36,093	36,093.0	0.00	1,179,857	4,089	214.55	36,093	36,093	0.00	33,978,884	829.38
miniYFJS30	45,051	31,888	31,888.0	0.00	3,071,590	8,773	487.83	31,888	31,888	0.00	6,128,809	144.46
mean	30,719.55	26,709.40	26,690.65	0.08	1,523,575	5,603	224.19	26,704.32	26,331.65	1.20	10,175,558	255.23

Table B12: Solutions found and computational cost of solving the small-scale instances with learning index $\alpha = 0.2$ using CPLEX and CP Optimizer with warm start.

Instance	EST/ECT	CPLEX						CP Optimizer				
		Makespan	Best Bound	Gap(%)	#iter	#B&B Nodes	CPU	Makespan	Best bound	Gap(%)	#branches	CPU
miniDAFJS01	20,757	19,443	19,443.0	0.00	698,096	4,945	39.06	19,443	19,443	0.00	1,304,215	24.46
miniDAFJS02	21,152	18,916	18,916.0	0.00	43,060	324	1.78	18,916	18,916	0.00	74,181	1.11
miniDAFJS03	17,621	17,419	17,419.0	0.00	6,368	91	0.40	17,419	17,419	0.00	28,742	0.38
miniDAFJS04	20,391	18,800	18,800.0	0.00	4,986	135	0.24	18,800	18,800	0.00	35,140	0.42
miniDAFJS05	19,253	17,596	17,596.0	0.00	174,306	2,110	7.65	17,596	17,596	0.00	391,357	5.86
miniDAFJS06	21,148	18,692	18,691.7	0.00	443,238	2,499	22.84	18,692	18,692	0.00	560,844	12.09
miniDAFJS07	24,767	24,256	24,256.0	0.00	13,304	243	0.77	24,256	24,256	0.00	40,595	0.71
miniDAFJS08	17,927	17,900	17,900.0	0.00	111	0	0.01	17,900	17,900	0.00	4,286	0.04
miniDAFJS09	22,398	20,797	20,797.0	0.00	414,070	1,767	22.76	20,797	20,797	0.00	4,397,314	85.23
miniDAFJS10	20,176	17,395	17,395.0	0.00	4,313	63	0.44	17,395	17,395	0.00	53,240	0.68
miniDAFJS11	28,718	25,304	25,304.0	0.00	879,945	4,672	54.21	25,304	25,304	0.00	504,180	11.10
miniDAFJS12	19,094	17,105	17,105.0	0.00	44,853	368	1.85	17,105	17,105	0.00	104,365	1.90
miniDAFJS13	15,425	14,077	14,077.0	0.00	1,996	31	0.17	14,077	14,077	0.00	14,170	0.15
miniDAFJS14	23,552	20,620	20,620.0	0.00	228,333	1,904	18.40	20,620	20,620	0.00	173,836	3.37
miniDAFJS15	20,524	18,625	18,624.8	0.00	43,116	709	2.14	18,625	18,625	0.00	156,266	2.66
miniDAFJS16	23,797	22,734	22,734.0	0.00	192,603	1,353	8.15	22,734	22,734	0.00	562,668	10.10
miniDAFJS17	19,941	18,253	18,253.0	0.00	4,414	116	0.32	18,253	18,253	0.00	46,881	0.67
miniDAFJS18	17,784	16,495	16,495.0	0.00	18,129	299	0.82	16,495	16,495	0.00	42,022	0.57
miniDAFJS19	19,030	18,474	18,474.0	0.00	8,267	49	0.81	18,474	18,474	0.00	72,980	1.13
miniDAFJS20	19,587	18,521	18,521.0	0.00	76,034	409	3.84	18,521	18,521	0.00	452,599	7.21
miniDAFJS21	21,299	19,430	19,429.0	0.01	12,431,780	94,510	2,062.47	19,430	11,665	39.96	137,469,215	3,600.00
miniDAFJS22	24,671	22,322	22,321.2	0.00	3,599,003	16,569	503.51	22,322	22,322	0.00	24,356,344	575.69
miniDAFJS23	23,847	21,372	18,192.4	14.88	22,085,662	81,901	3,600.00	21,031	11,409	45.75	141,406,414	3,600.00
miniDAFJS24	22,524	20,389	20,389.0	0.00	5,343,230	22,851	631.61	20,389	20,389	0.00	58,316,562	1,352.68
miniDAFJS25	21,923	18,400	18,400.0	0.00	1,455,883	5,536	160.36	18,400	18,400	0.00	2,095,941	46.88
miniDAFJS26	21,613	18,396	18,396.0	0.00	3,994,877	35,050	480.75	18,396	18,396	0.00	18,863,883	405.75
miniDAFJS27	23,089	18,501	18,501.0	0.00	12,966,109	37,754	1,642.21	18,501	10,545	43.00	129,872,618	3,600.00
miniDAFJS28	24,429	18,762	18,762.0	0.00	4,874,294	33,717	338.85	18,762	18,762	0.00	4,427,662	82.30
miniDAFJS29	18,560	18,253	18,253.0	0.00	32,091	183	2.94	18,253	18,253	0.00	531,960	10.24
miniDAFJS30	21,421	19,735	18,098.1	8.29	23,459,868	67,487	3,600.00	19,504	11,670	40.17	123,510,677	3,600.00
miniYFJS01	33,697	31,008	31,008.0	0.00	10,554	42	0.89	31,008	31,008	0.00	72,892	1.34
miniYFJS02	25,969	22,010	22,010.0	0.00	12,966	86	0.72	22,010	22,010	0.00	58,057	0.71
miniYFJS03	52,865	38,935	38,935.0	0.00	36,885	205	1.74	38,935	38,935	0.00	617,418	10.18
miniYFJS04	24,017	23,774	23,774.0	0.00	3,571	20	0.35	23,774	23,774	0.00	68,927	1.28
miniYFJS05	24,743	22,843	22,843.0	0.00	19,695	119	1.74	22,843	22,843	0.00	64,107	1.25
miniYFJS06	27,366	27,366	27,366.0	0.00	20,266	49	1.42	27,366	27,366	0.00	170,880	2.60
miniYFJS07	44,128	38,932	38,932.0	0.00	8,969	120	0.84	38,932	38,932	0.00	132,293	2.22
miniYFJS08	31,133	29,464	29,464.0	0.00	105,069	606	4.66	29,464	29,464	0.00	162,492	2.67
miniYFJS09	34,357	33,763	33,763.0	0.00	228,616	1,229	9.71	33,763	33,763	0.00	239,790	4.40
miniYFJS10	28,513	25,072	25,071.6	0.00	330,870	2,304	22.64	25,072	25,072	0.00	780,644	13.16
miniYFJS11	43,254	36,307	36,307.0	0.00	47,377	224	3.65	36,307	36,307	0.00	210,733	3.21
miniYFJS12	33,495	26,219	26,218.9	0.00	899,683	3,949	120.85	26,219	26,219	0.00	1,490,833	33.09
miniYFJS13	26,536	25,619	25,619.0	0.00	64,547	257	6.98	25,619	25,619	0.00	2,517,616	57.67
miniYFJS14	37,157	27,428	27,428.0	0.00	1,341,266	10,416	97.44	27,428	27,428	0.00	1,364,158	21.68
miniYFJS15	46,583	38,256	38,256.0	0.00	113,347	531	5.77	38,256	38,256	0.00	559,735	9.26
miniYFJS16	30,576	29,442	29,442.0	0.00	129,890	593	11.24	29,442	29,442	0.00	850,216	16.85
miniYFJS17	44,916	37,465	37,465.0	0.00	831,549	2,535	48.87	37,465	37,465	0.00	551,109	10.39
miniYFJS18	29,764	25,067	25,067.0	0.00	109,409	268	5.82	25,067	25,067	0.00	238,385	3.51
miniYFJS19	37,034	29,207	29,207.0	0.00	692,645	2,990	60.79	29,207	29,207	0.00	1,273,076	24.05
miniYFJS20	37,063	27,091	27,091.0	0.00	273,694	2,124	36.57	27,091	27,091	0.00	1,522,617	34.02
miniYFJS21	35,905	32,166	32,166.0	0.00	14,778,868	57,153	2,605.59	32,166	32,166	0.00	17,294,309	347.83
miniYFJS22	30,212	28,985	28,985.0	0.00	4,982,381	14,455	890.42	28,985	17,383	40.03	139,564,139	3,600.00
miniYFJS23	45,823	36,562	27,468.2	24.87	22,147,433	72,111	3,600.00	35,441	18,601	47.52	141,030,458	3,600.00
miniYFJS24	37,651	27,023	27,023.0	0.00	1,487,508	5,429	179.90	27,023	27,023	0.00	1,170,101	20.04
miniYFJS25	35,966	32,346	32,346.0	0.00	10,970,288	46,815	1,745.69	32,346	32,346	0.00	64,427,569	1,373.88
miniYFJS26	53,604	43,452	43,452.0	0.00	896,287	3,959	140.71	43,452	43,452	0.00	5,659,611	126.53
miniYFJS27	33,799	31,571	31,571.0	0.00	1,067,122	3,564	211.41	31,571	31,571	0.00	1,557,013	46.11
miniYFJS28	37,547	30,428	30,428.0	0.00	919,151	2,475	96.71	30,428	30,428	0.00	1,404,981	24.79
miniYFJS29	39,146	32,826	32,826.0	0.00	5,557,857	9,166	932.45	32,826	17,217	47.55	138,565,968	3,600.00
miniYFJS30	40,752	29,848	29,848.0	0.00	2,521,758	9,078	398.67	29,848	29,848	0.00	27,658,346	652.34
mean	28,666.48	24,957.62	24,725.73	0.80	2,735,864	11,175	407.56	24,929.40	23,642.27	5.07	20,019,160	511.47

Table B13: Solutions found and computational cost of solving the small-scale instances with learning index $\alpha = 0.3$ using CPLEX and CP Optimizer with warm start.

Instance	EST/ECT	CPLEX						CP Optimizer				
		Makespan	Best Bound	Gap(%)	#iter	#B&B Nodes	CPU	Makespan	Best bound	Gap(%)	#branches	CPU
DAFJS01	29,769	24,020	21,819.4	9.16	7,452,098	31,212	3,600.00	23,919	18,588	22.29	87,677,764	3,600.00
DAFJS02	32,467	29,175	23,473.8	19.54	8,982,073	22,949	3,600.00	26,575	18,533	30.26	115,514,402	3,600.00
DAFJS03	53,688	53,688	42,752.5	20.37	175,398	0	3,600.00	51,168	41,271	19.34	8,818,648	3,600.04
DAFJS04	54,082	54,082	45,641.9	15.61	438,547	100	3,600.00	52,438	44,062	15.97	13,931,028	3,600.04
DAFJS05	45,862	45,862	30,532.0	33.43	772,163	745	3,600.00	36,043	23,378	35.14	36,348,602	3,600.00
DAFJS06	51,925	51,925	31,798.3	38.76	388,755	112	3,600.00	37,045	23,364	36.93	14,312,410	3,600.03
DAFJS07	57,193	57,193	34,222.1	40.16	62,280	0	3,600.00	49,207	33,601	31.71	1,976,990	3,600.15
DAFJS08	62,159	62,159	43,685.0	29.72	96,814	0	3,600.00	54,868	43,331	21.03	3,041,092	3,600.20
DAFJS09	48,680	48,680	36,346.8	25.34	430,651	194	3,600.00	41,051	22,460	45.29	13,977,822	3,600.04
DAFJS10	58,695	58,695	39,541.3	32.63	177,913	90	3,600.00	45,523	23,440	48.51	8,119,499	3,600.09
DAFJS11	67,594	67,594	44,308.5	34.45	80,721	0	3,600.00	59,135	44,061	25.49	1,689,372	3,600.13
DAFJS12	68,322	68,322	38,374.4	43.83	80,132	0	3,600.00	65,342	35,263	46.03	762,522	3,600.41
DAFJS13	61,794	61,794	50,238.7	18.70	82,485	68	3,600.00	53,674	20,981	60.91	6,791,225	3,600.08
DAFJS14	76,862	76,862	55,258.5	28.11	88,444	23	3,600.00	61,329	23,996	60.87	7,208,096	3,600.03
DAFJS15	72,288	72,288	0.0	100.00	149,168	0	3,600.00	67,345	33,814	49.79	2,488,269	3,600.21
DAFJS16	76,204	76,204	42,614.1	44.08	98,717	0	3,600.00	73,036	42,465	41.86	971,091	3,600.35
DAFJS17	73,219	73,219	57,424.6	21.57	90,946	20	3,600.00	67,140	20,260	69.82	2,952,587	3,600.06
DAFJS18	78,862	78,862	58,244.2	26.14	64,440	34	3,600.00	67,983	21,838	67.88	4,079,001	3,600.05
DAFJS19	66,412	66,412	36,122.7	45.61	79,488	17	3,600.00	47,831	35,267	26.27	2,969,187	3,600.17
DAFJS20	78,778	78,778	49,551.0	37.10	45,224	0	3,600.00	64,042	29,175	54.44	2,695,930	3,600.09
DAFJS21	78,320	78,320	52,221.0	33.32	70,472	0	3,600.00	71,099	33,405	53.02	1,212,043	3,600.11
DAFJS22	70,892	70,892	45,864.2	35.30	73,365	0	3,600.00	60,696	30,309	50.06	1,061,438	3,600.48
DAFJS23	49,969	49,110	31,583.5	35.69	66,636	2	3,600.00	46,049	31,047	32.58	1,848,191	3,600.06
DAFJS24	57,411	57,411	36,555.4	36.33	64,282	0	3,600.00	53,865	32,194	40.23	1,571,128	3,600.12
DAFJS25	82,055	82,055	45,918.6	44.04	106,844	0	3,600.00	76,059	38,155	49.84	1,771,046	3,600.35
DAFJS26	81,480	81,480	46,118.3	43.40	88,562	0	3,600.00	71,737	36,963	48.47	1,013,669	3,600.29
DAFJS27	81,470	81,470	0.0	100.00	85,073	0	3,600.00	79,602	32,947	58.61	1,538,272	3,600.71
DAFJS28	62,568	62,568	36,566.3	41.56	132,517	0	3,600.00	50,336	36,392	27.70	2,150,631	3,600.15
DAFJS29	72,938	72,938	42,988.3	41.06	73,477	0	3,600.00	63,538	40,983	35.50	2,816,597	3,600.10
DAFJS30	61,147	61,147	36,940.9	39.59	120,370	0	3,600.00	54,837	31,515	42.53	1,437,024	3,600.13
YFJS01	87,203	80,523	58,325.9	27.57	2,231,804	3,577	3,600.00	69,301	53,512	22.78	57,905,698	3,600.00
YFJS02	81,579	77,409	69,500.1	10.22	6,162,169	14,824	3,600.00	72,763	61,375	15.65	104,830,218	3,600.00
YFJS03	40,197	32,538	32,538.0	0.00	705,691	4,291	77.25	32,538	32,538	0.00	3,538,957	52.94
YFJS04	47,467	35,883	35,883.0	0.00	8,538,098	25,417	2,588.40	35,883	35,883	0.00	34,304,206	726.44
YFJS05	46,138	41,034	37,942.0	7.54	17,274,764	42,657	3,600.00	41,315	32,271	21.89	151,055,958	3,600.00
YFJS06	52,487	44,064	35,226.2	20.06	6,770,129	15,500	3,600.00	41,178	25,961	36.95	108,207,193	3,600.00
YFJS07	54,457	42,121	36,422.6	13.53	6,094,597	15,382	3,600.00	41,394	32,925	20.46	119,103,804	3,600.00
YFJS08	49,626	32,573	32,572.5	0.00	3,973,967	5,394	669.55	32,573	32,573	0.00	1,563,189	26.84
YFJS09	28,354	25,250	18,627.4	26.23	1,117,559	1,217	3,600.00	22,681	17,207	24.13	53,649,039	3,600.00
YFJS10	42,583	37,374	34,677.2	7.22	13,053,596	22,861	3,600.00	37,372	37,372	0.00	56,769,122	1,280.13
YFJS11	59,356	55,057	44,028.4	20.03	2,394,673	3,104	3,600.00	48,783	39,155	19.74	98,945,654	3,600.00
YFJS12	70,830	65,954	38,580.6	41.50	3,797,975	6,334	3,600.00	49,831	33,937	31.90	86,991,624	3,600.00
YFJS13	50,805	49,500	32,024.8	35.30	3,121,769	4,355	3,600.00	38,679	27,343	29.31	99,873,366	3,600.00
YFJS14	129,428	129,428	95,515.0	26.20	121,181	0	3,600.00	119,371	94,991	20.42	11,111,915	3,600.08
YFJS15	138,196	138,196	90,156.9	34.76	106,751	0	3,600.00	121,469	89,818	26.06	9,522,875	3,600.18
YFJS16	127,055	127,055	0.0	100.00	105,543	0	3,600.00	122,909	86,618	29.53	8,567,302	3,600.09
YFJS17	109,112	109,112	0.0	100.00	97,301	0	3,600.00	109,112	76,416	29.97	700,178	3,600.31
YFJS18	133,703	133,703	0.0	100.00	108,178	0	3,600.00	131,973	81,982	37.88	714,129	3,600.29
YFJS19	107,055	107,055	0.0	100.00	85,934	0	3,600.00	107,055	62,129	41.97	646,096	3,600.29
YFJS20	97,868	97,868	0.0	100.00	87,338	0	3,600.00	97,473	65,048	33.27	598,635	3,600.31
mean	69,332.08	67,298.04	36,974.54	37.69	1,933,341	4,410	3,450.70	60,923.30	39,242.24	33.81	27,226,895	3,353.85

Table B14: Solutions found and computational cost of solving the large-scale instances with learning index $\alpha = 0.1$ using CPLEX and CP Optimizer with warm start.

Instance	EST/ECT	CPLEX						CP Optimizer				
		Makespan	Best Bound	Gap(%)	#iter	#B&B Nodes	CPU	Makespan	Best bound	Gap(%)	#branches	CPU
DAFJS01	28,920	25,283	16,997.2	32.77	9,189,402	31,324	3,600.00	22,426	14,141	36.94	78,661,578	3,600.00
DAFJS02	28,154	26,381	20,539.3	22.14	9,725,651	19,297	3,600.00	24,235	13,966	42.37	93,746,155	3,600.00
DAFJS03	48,555	48,555	31,370.8	35.39	166,514	0	3,600.00	46,243	29,519	36.17	4,979,539	3,600.05
DAFJS04	48,461	48,461	34,101.7	29.63	512,319	400	3,600.00	45,138	32,038	29.02	10,853,381	3,600.04
DAFJS05	44,790	41,089	25,260.7	38.52	1,000,351	1,218	3,600.00	31,848	17,025	46.54	33,396,644	3,600.01
DAFJS06	44,228	44,228	25,488.6	42.37	479,423	260	3,600.00	32,242	16,745	48.06	14,085,957	3,600.04
DAFJS07	50,802	50,802	23,876.5	53.00	72,790	2	3,600.00	46,088	23,042	50.00	2,639,211	3,600.07
DAFJS08	53,439	53,439	30,550.4	42.83	69,465	2	3,600.00	51,188	29,902	41.58	2,006,577	3,600.05
DAFJS09	42,537	42,537	28,905.4	32.05	428,592	497	3,600.00	36,710	16,016	56.37	13,415,934	3,600.05
DAFJS10	46,257	46,257	30,020.2	35.10	110,576	10	3,600.00	38,128	16,354	57.11	9,361,700	3,600.02
DAFJS11	56,847	56,847	29,861.3	47.47	58,766	0	3,600.00	52,705	29,499	44.03	1,726,553	3,600.12
DAFJS12	55,575	55,491	27,830.8	49.85	104,499	0	3,600.00	52,557	23,462	55.36	791,353	3,600.19
DAFJS13	51,317	51,317	39,199.3	23.61	104,610	70	3,600.00	46,814	14,433	69.17	5,087,215	3,600.13
DAFJS14	62,724	62,724	43,921.5	29.98	69,352	100	3,600.00	50,757	16,518	67.46	6,832,407	3,600.03
DAFJS15	54,353	54,353	0.0	100.00	119,809	0	3,600.00	53,545	22,324	58.31	1,362,621	3,600.21
DAFJS16	65,550	65,550	28,822.4	56.03	99,659	0	3,600.00	56,689	28,167	50.31	2,643,950	3,600.21
DAFJS17	63,177	63,177	42,492.7	32.74	59,945	45	3,600.00	53,281	13,683	74.32	3,724,897	3,600.04
DAFJS18	61,831	61,831	45,629.1	26.20	58,606	23	3,600.00	56,222	14,813	73.65	4,117,461	3,600.02
DAFJS19	58,001	58,001	27,727.7	52.19	78,059	17	3,600.00	41,977	24,250	42.23	2,932,057	3,600.14
DAFJS20	63,588	63,588	37,371.6	41.23	45,462	0	3,600.00	52,206	19,568	62.52	2,572,205	3,600.06
DAFJS21	63,202	63,202	36,581.5	42.12	61,074	0	3,600.00	60,824	22,142	63.60	1,204,968	3,600.41
DAFJS22	56,115	56,115	31,355.8	44.12	73,881	0	3,600.00	55,609	19,800	64.39	968,545	3,600.51
DAFJS23	47,616	47,616	23,845.2	49.92	59,629	12	3,600.00	39,346	21,422	45.55	3,467,835	3,600.05
DAFJS24	49,019	49,019	26,578.4	45.78	57,505	0	3,600.00	48,490	21,763	55.12	1,018,699	3,600.10
DAFJS25	66,262	66,262	31,724.8	52.12	132,844	0	3,600.00	63,020	24,843	60.58	632,297	3,600.34
DAFJS26	72,226	72,226	32,136.9	55.51	94,729	0	3,600.00	63,293	24,182	61.79	918,715	3,600.27
DAFJS27	64,797	64,797	36,164.1	44.19	87,932	0	3,600.00	64,397	21,584	66.48	432,593	3,601.27
DAFJS28	52,639	52,639	25,210.9	52.11	60,885	0	3,600.00	47,949	24,712	48.46	2,212,702	3,600.06
DAFJS29	59,792	59,792	33,769.8	43.52	72,213	0	3,600.00	54,082	27,497	49.16	2,507,626	3,600.12
DAFJS30	55,015	55,015	27,332.4	50.32	95,898	0	3,600.00	51,358	21,222	58.68	1,231,068	3,600.12
YFJS01	84,152	84,152	48,050.6	42.90	1,576,648	2,189	3,600.00	63,979	39,902	37.63	47,435,397	3,600.00
YFJS02	66,853	66,853	59,364.2	11.20	6,076,106	15,067	3,600.00	66,853	46,715	30.12	100,390,515	3,600.00
YFJS03	35,380	30,073	30,073.0	0.00	1,173,565	4,411	158.81	30,073	30,073	0.00	1,718,667	26.69
YFJS04	43,614	32,670	31,272.0	4.28	18,312,983	51,565	3,600.00	32,670	32,670	0.00	145,285,245	3,351.91
YFJS05	40,851	38,682	34,501.6	10.81	16,893,635	45,965	3,600.00	37,255	24,921	33.11	143,272,534	3,600.00
YFJS06	48,564	40,577	31,264.4	22.95	6,716,264	15,400	3,600.00	37,834	19,541	48.35	105,933,515	3,600.00
YFJS07	51,261	41,143	32,385.1	21.29	8,555,612	22,795	3,600.00	37,700	26,063	30.87	108,205,656	3,600.00
YFJS08	44,074	30,184	30,184.0	0.00	6,630,083	13,438	1,122.24	30,184	30,184	0.00	10,026,991	189.26
YFJS09	26,072	23,430	15,202.7	35.11	1,025,961	811	3,600.00	21,363	11,324	46.99	41,544,117	3,600.00
YFJS10	40,658	36,188	28,977.5	19.93	13,577,417	17,976	3,600.00	34,364	22,363	34.92	151,110,218	3,600.00
YFJS11	51,469	49,781	35,759.1	28.17	2,397,423	3,278	3,600.00	47,486	30,073	36.67	105,847,458	3,600.00
YFJS12	59,898	59,898	32,801.5	45.24	2,742,850	3,376	3,600.00	43,976	26,939	38.74	80,285,903	3,600.00
YFJS13	45,084	43,510	27,082.1	37.76	3,964,419	4,771	3,600.00	34,609	21,361	38.28	83,647,263	3,600.00
YFJS14	109,469	109,469	68,790.4	37.16	99,060	0	3,600.00	105,548	68,423	35.17	9,415,378	3,600.03
YFJS15	112,424	112,424	65,710.6	41.55	107,775	0	3,600.00	105,767	64,923	38.62	7,511,943	3,600.04
YFJS16	106,855	104,886	64,483.7	38.52	91,355	0	3,600.00	103,685	62,865	39.37	7,096,820	3,600.09
YFJS17	85,736	85,736	0.0	100.00	129,523	0	3,600.00	85,563	51,540	39.76	680,233	3,600.38
YFJS18	99,429	99,429	0.0	100.00	140,428	0	3,600.00	98,982	55,101	44.33	927,140	3,600.38
YFJS19	89,561	89,561	0.0	100.00	141,276	0	3,600.00	89,384	41,690	53.36	2,082,026	3,600.30
YFJS20	91,958	91,958	0.0	100.00	187,122	0	3,600.00	87,353	43,711	49.96	1,498,192	3,600.27
mean	58,983.02	57,543.96	30,611.39	41.99	2,277,799	5,086	3,481.62	52,679.90	27,900.28	45.83	29,268,953	3,455.48

Table B15: Solutions found and computational cost of solving the large-scale instances with learning index $\alpha = 0.2$ using CPLEX and CP Optimizer with warm start.

Instance	EST/ECT	CPLEX						CP Optimizer				
		Makespan	Best Bound	Gap(%)	#iter	#B&B Nodes	CPU	Makespan	Best bound	Gap(%)	#branches	CPU
DAFJS01	22,616	22,616	13,167.4	41.78	3,827,740	5,238	3,600.00	20,403	10,711	47.50	68,269,786	3,600.00
DAFJS02	26,183	26,183	17,072.7	34.79	4,198,368	8,285	3,600.00	22,433	10,500	53.19	86,906,491	3,600.00
DAFJS03	43,964	43,964	22,718.5	48.32	144,500	15	3,600.00	41,422	21,152	48.94	4,627,550	3,600.07
DAFJS04	43,281	43,281	24,986.7	42.27	343,206	100	3,600.00	40,114	23,278	41.97	12,053,217	3,600.05
DAFJS05	38,237	36,903	20,487.4	44.48	645,659	704	3,600.00	28,210	12,401	56.04	32,077,883	3,600.01
DAFJS06	38,007	38,007	20,173.8	46.92	335,127	400	3,600.00	27,597	12,016	56.46	13,082,455	3,600.08
DAFJS07	41,285	41,285	16,886.3	59.10	65,985	2	3,600.00	37,617	15,801	58.00	1,850,241	3,600.08
DAFJS08	46,323	46,323	21,333.9	53.95	102,025	2	3,600.00	42,793	20,633	51.78	1,993,744	3,600.05
DAFJS09	41,418	41,418	22,184.7	46.44	383,268	15	3,600.00	32,061	11,423	64.37	11,722,588	3,600.06
DAFJS10	38,771	38,771	22,462.8	42.06	135,769	14	3,600.00	33,173	11,408	65.61	5,772,604	3,600.01
DAFJS11	47,805	47,805	20,144.8	57.86	87,419	0	3,600.00	44,530	19,709	55.74	1,677,928	3,600.17
DAFJS12	45,735	45,735	18,894.1	58.69	82,318	0	3,600.00	44,535	15,611	64.95	553,670	3,600.25
DAFJS13	42,319	42,319	29,209.6	30.98	128,682	26	3,600.00	38,166	9,929	73.98	5,044,925	3,600.03
DAFJS14	49,459	49,459	32,978.7	33.32	83,001	37	3,600.00	43,122	11,318	73.75	5,275,961	3,600.07
DAFJS15	49,206	49,206	0.0	100.00	125,540	0	3,600.00	44,806	14,671	67.26	854,825	3,600.28
DAFJS16	56,727	56,727	20,217.7	64.36	103,463	0	3,600.00	55,865	18,661	66.60	573,262	3,600.14
DAFJS17	54,527	54,527	29,861.8	45.23	45,959	23	3,600.00	45,227	9,240	79.57	2,993,651	3,600.05
DAFJS18	51,823	51,823	32,270.5	37.73	61,906	13	3,600.00	46,748	10,047	78.51	3,493,038	3,600.04
DAFJS19	46,880	46,880	21,073.3	55.05	63,928	20	3,600.00	38,336	16,675	56.50	2,672,499	3,600.04
DAFJS20	51,686	51,686	25,710.8	50.26	58,783	0	3,600.00	44,964	13,124	70.81	1,734,801	3,600.10
DAFJS21	53,010	53,010	0.0	100.00	74,236	0	3,600.00	48,203	14,664	69.58	779,219	3,600.16
DAFJS22	45,005	45,005	21,311.2	52.65	92,999	0	3,600.00	42,689	12,934	69.70	976,310	3,600.17
DAFJS23	39,249	39,249	18,008.0	54.12	71,380	20	3,600.00	35,197	14,782	58.00	2,382,355	3,600.06
DAFJS24	44,851	44,851	19,016.8	57.60	63,275	0	3,600.00	42,422	14,688	65.38	1,735,518	3,600.09
DAFJS25	51,964	51,964	0.0	100.00	131,327	0	3,600.00	51,260	16,174	68.45	515,962	3,600.25
DAFJS26	57,182	57,182	22,482.9	60.68	143,324	0	3,600.00	57,182	15,820	72.33	425,120	3,600.22
DAFJS27	58,138	58,138	25,112.9	56.80	108,824	0	3,600.00	56,395	14,138	74.93	394,641	3,600.25
DAFJS28	42,898	42,898	18,443.6	57.01	64,153	0	3,600.00	40,329	16,781	58.39	1,751,266	3,600.10
DAFJS29	51,606	51,606	23,009.5	55.41	62,133	0	3,600.00	51,489	18,452	64.16	1,058,684	3,600.35
DAFJS30	43,859	43,859	19,165.2	56.30	89,030	0	3,600.00	40,340	14,275	64.61	1,633,798	3,600.14
YFJS01	70,402	66,116	39,167.5	40.76	1,158,062	2,210	3,600.00	56,433	29,808	47.18	44,860,548	3,600.00
YFJS02	61,111	59,001	50,179.4	14.95	3,539,312	8,216	3,600.00	59,001	35,588	39.68	87,354,520	3,600.00
YFJS03	31,680	27,686	27,686.0	0.00	1,406,947	4,582	300.73	27,686	27,686	0.00	978,295	18.26
YFJS04	38,898	30,112	28,392.0	5.71	19,297,552	53,724	3,600.00	29,692	18,161	38.84	145,538,736	3,600.00
YFJS05	41,586	35,508	28,650.2	19.31	15,950,848	56,285	3,600.00	34,430	19,314	43.90	133,399,300	3,600.00
YFJS06	46,811	36,319	26,718.0	26.44	5,935,128	13,303	3,600.00	33,276	14,714	55.78	109,619,712	3,600.00
YFJS07	39,896	34,329	27,316.2	20.43	9,435,320	22,470	3,600.00	33,092	20,397	38.36	120,661,325	3,600.00
YFJS08	39,325	28,192	28,192.0	0.00	23,691,336	30,023	3,493.74	28,192	17,787	36.91	176,578,372	3,600.00
YFJS09	24,027	24,027	13,510.6	43.77	1,149,267	1,097	3,600.00	20,077	9,525	52.56	34,152,226	3,600.00
YFJS10	39,607	32,718	28,401.0	13.19	13,991,103	24,326	3,600.00	31,559	17,808	43.57	141,539,969	3,600.00
YFJS11	45,051	45,051	32,748.3	27.31	4,876,919	6,985	3,600.00	40,350	23,170	42.58	88,371,000	3,600.00
YFJS12	52,398	52,398	29,073.7	44.51	2,259,926	2,346	3,600.00	39,460	21,335	45.93	60,218,602	3,600.00
YFJS13	40,119	38,382	23,819.3	37.94	2,398,021	3,540	3,600.00	31,396	16,668	46.91	82,931,361	3,600.00
YFJS14	92,457	92,457	49,959.4	45.96	107,697	0	3,600.00	89,217	49,252	44.80	8,439,056	3,600.03
YFJS15	96,547	96,547	47,830.0	50.46	110,149	0	3,600.00	85,781	46,947	45.27	7,360,300	3,600.05
YFJS16	92,811	92,811	49,439.6	46.73	81,730	0	3,600.00	85,251	45,648	46.45	8,426,120	3,600.03
YFJS17	73,682	73,682	0.0	100.00	74,507	0	3,600.00	73,467	34,763	52.68	814,189	3,600.31
YFJS18	87,059	87,059	0.0	100.00	90,558	0	3,600.00	85,942	37,042	56.90	1,232,717	3,600.34
YFJS19	74,431	74,431	0.0	100.00	167,702	0	3,600.00	74,431	27,976	62.41	805,780	3,600.17
YFJS20	72,481	72,481	29,700.2	59.02	187,909	0	3,600.00	71,952	29,376	59.17	1,100,863	3,600.27
mean	50,487.86	49,239.74	23,183.38	48.81	2,356,666	4,880	3,531.89	45,366.26	19,679.62	55.94	30,585,340	3,528.46

Table B16: Solutions found and computational cost of solving the large-scale instances with learning index $\alpha = 0.3$ using CPLEX and CP Optimizer with warm start.