Christ, Q., Dauzère-Pérès, S., & Lepelletier, G. (2023). A three-step approach for decision support in operational production planning of complex manufacturing systems. International Journal of Production Research, 61(17), 5860-5885.

https://doi.org/10.1080/00207543.2022.2118387

An Improved Decision Support Model for Scheduling Production in an Engineer-To-Order Manufacturer

Stéphane Dauzère-Pérès[1,2] Sigrid Lise Nonås[3]

[1]Mines Saint-Etienne, Univ Clermont Auvergne
CNRS, UMR 6158 LIMOS
CMP, Department of Manufacturing Sciences and Logistics
F-13541 Gardanne, France
E-mail: dauzere-peres@emse.fr

[2]Department of Accounting and Operations Management
BI Norwegian Business School
0484 Oslo, Norway

[3]Department of Business and Management Science
NHH Norwegian School of Economics
5045 Bergen, Norway
E-mail: Sigrid-Lise.Nonas@nhh.no

Abstract

This paper outlines a mathematical model to solve a scheduling problem for a company engineering and producing propellers to order. Nonås and Olsen (2005) have previously introduced a Mixed Integer Programming model for this production setting with the objective of minimizing the total tardiness. The mathematical model could however not be used to solve realistic sized problem instances, because of the very large solution time. We propose a new time indexed formulation that can solve most industrial problem instances in less than 10 minutes. This work is further extended by taking into account limited storage capacity and by proposing different methods to balance between total tardiness and maximum tardiness. We illustrate how the solution time and the criteria change for different setups of the mathematical model and suggest which setup to use for different scenarios. The paper also discusses how the new model can be extended to include unexpected events such as emergency orders and unavailable production equipment.

Keywords: Scheduling; Engineer-To-Order; Optimization; Mathematical Modeling

1. Introduction

This paper outlines a mathematical model that is able to solve scheduling problems for a foundry (an engineer-to-order company) producing propellers for ships. The objective of the company is to find a production plan that minimizes both the total tardiness and the maximum tardiness, while taking into account capacity restrictions on labor, tools and space. Since minimizing total tardiness and minimizing maximum tardiness can be conflicting objectives, we have to find a production plan with a good balance between these objectives. The problem is previously studied by Nonås and Olsen

(2005), where a Mixed Integer Linear Programming (MILP) model and a heuristic solution procedure are proposed where the main focus is on minimizing total tardiness. The MILP model can however not be used to solve realistic sized problem instances due to a very large solution time. We here extend their work by proposing a new and more efficient time indexed formulation, so that medium-sized scheduling problems on a one-month horizon can be solved within a reasonable time frame. We further improve their work by studying how to find a production plan with the right balance between total tardiness and maximum tardiness. The main objective of the company is to minimize the total number of days the jobs are late, but they would in addition also like to avoid orders that are delivered too late, because that would make the customer unsatisfied and might impact future demand. Formally, this translates into keeping the maximum tardiness low, in addition to the main objective. Different methods are suggested and the results for each method, both in relation to the solution time and the balance between total tardiness and maximum tardiness, are presented. Suggestions are given for which alternative to choose based on which market the foundry faces. The new mathematical model is also extended so that the production plan can take into account the limited storage capacity at the foundry. Constraints that specify the earliest possible completion date for each job are included in the model to meet this restriction. The company meets from time to time challenges as how to deal with orders that cannot be processed as planned due to unavailable equipment and how to schedule emergency orders in a currently fully booked schedule. We here propose to meet these challenges by including simple constraints in the new model.

The paper is organized as follows. First, the related literature is reviewed in Section 2. In Section 3, a new and more efficient formulation for the scheduling problem is proposed. We discuss how to get the right balance between total tardiness and maximum tardiness in Section 4. Numerical results that illustrate the huge improvement in solution time and how the solution time differs with different objectives are presented in Section 5. In Section 6, we make suggestions on the variant of the new model the company should use. In addition, we briefly discuss in Section 7 how to take into account emergency orders and forced changes in the schedule due to missing equipment. A summary and a conclusion are given in Section 8.

2. Literature review

Few works have been published that outline mathematical models that can solve scheduling problems for real world foundry cases. Teixeira et al. (2010) describe the development and solution of binary integer formulations for production scheduling problems in market-driven foundries. They claim that the characteristics and constraints involved in a typical production environment at these industries challenge the formulation of mathematical programming models that can be computationally solved when considering real applications. Matibevic et al. (2008) develop a new mathematical model for scheduling foundry operations based on the MRP II (Manufacturing Resource Planning), JIT (Just in Time) and OPT (Optimized Production Technology) concepts. The proposed model is successfully implemented into the ERP (Enterprise Resource Planning) system of Aluminium Ltd. in Mostar. The authors do not specify the size of the problem, but they claim that the time required to develope a production plan using the mathematical model takes 15 minutes, while it takes 60 minutes to do it manually. Gauri (2009) presents a weighted integer goal programming model for the product-mix planning, developed in the context of a small scale iron foundry. The implementation of the model is illustrated using real life data from an Indian foundry. Hans and van de Velde (2011) study the monthly planning and scheduling of the sand-casting department in a metal foundry. The problem can be characterized as a single-level multi-item capacitated lot-sizing model with a variety of additional process-specific constraints. The main objective is to smooth production. They present a hierarchical solution approach (MILP, shortest path, local improvement) that is by far superior to the quality of the schedules constructed by an expert production planner with no other tool than a plan board. They consider problems with around 40–50 jobs, two different product specific production lines, and a scheduling horizon of 20 days. CPLEX was not able to solve most of their real-life instances to optimality, not even with more than several hours of computing time. However, it consistently managed to find very satisfactory feasible solutions within a practical time limit of 10 minutes. Ballestín et al. (2012) study a multi-objective production scheduling problem for a medium sized foundry delivering a wide range of castings. To solve this problem, the authors present two methodological approaches, a sequence of Mixed Integer Linear Programs (MILP) on a rolling horizon and a metaheuristic algorithm that models the problem as a project scheduling problem. In the first approach, each MILP is solved for a restricted set of customer orders, to minimize the time required to find the optimal solution. More specificaly, each MILP aims to determine the set of units to be molded when the furnace is next tapped (the furnace can be tapped up to 6 times per day). Ballestin et al. (2012) compare the two solutions approaches for a planning horizon of 10, 15 and 30 days, with different types of workloads. The average solution times for the MILP approach vary from 37.78 seconds to 29.24 seconds. Nonås and Olsen (2005) study a scheduling problem for an engineer-to-order company producing propellers for ships. The objective is to find a production plan that minimizes total tardiness, taking into account capacity restriction on labor, tool, and space. The problem presented can be viewed as a scheduling problem with limited resources used in parallel. The common way to solve this problem is to use priority rules for scheduling jobs on machines and for assigning auxiliary resources to jobs processed in machines. Instead of dealing with a combination of different assignment rules, one assignment rule is considered that determines the "capacity combination" to use at the set of machines. In this paper, we extend the work of Nonås and Olsen (2005) by proposing a new and more efficient time indexed formulation for the scheduling problem.

Time indexed formulations for machine scheduling problems have received a great deal of attention; mainly due to the strong lower bounds often provided by their linear programming relaxations.

Dyer and Wolsey (1990) introduce the time-indexed formulation and prove its strength relative to some other formulations. Sousa and Wolsey (1992) derive some valid inequalities and perform some computational experiments. It is based on a time discretization of a time horizon M and uses binary variables x_{jt} to indicate that job j in J is completed at time t ($x_{jt} = 1$). For a more general study of time indexed formulations and polyhedral approaches for machine scheduling problems, we refer to the comprehensive survey of Queyranne and Schulz (1994). Several researchers (van den Akker et al. (1999), Chen and Powell (1999b), Chen and Powell (1999a), van den Akker et al. (2000), Bigras et al. (2008)) have used time indexed formulations for scheduling jobs on a single machine or on parallel machines using column generation and considering weighted sum objective functions for parallel machines (e.g. weighted sum of completion times and weighted sum of tardy jobs). Ku and Beck (2016) recently showed that the disjunctive formulation of the job-shop scheduling problem proposed by Manne (1960) performs better with a standard Mixed Integer Programming solver than time indexed formulations. However, this is true because as soon as jobs have routes (i.e. multiple operations to perform), precedence constraints are required and those weaken time indexed formulations. Hence, these formulations are more effective for single machine scheduling problems and parallel machine scheduling problems such as our problem. Detienne et al. (2011) introduce an integer linear programming model relying on time indexed variables for a parallel machine scheduling problem with the objective of minimizing a regular step total cost function. Obeid et al. (2014) are for instance proposing time indexed formulations to schedule job families on non-identical parallel machines with time constraints that can solve problem instances with up to 70 jobs. Boland et al. (2016) present a generalization of the classical time indexed formulation for single machine scheduling problems in which at most two jobs can be processed in each time period. Deghdak et al. (2016) introduce a time indexed formulation for a large scheduling problem of evacuation operations after a major disaster. The model is using this formulation within a matheuristic. Velez et al. (2017) propose and compare different ways of modeling sequence-dependent changeovers in time indexed formulations when tasks are scheduled on parallel machines. In a recent paper, Pan and Liang (2017) cover very well the literature on time indexed formulations, and in particular their relaxations, for single machine scheduling problems. They develop a novel approach to compute optimal dual solutions that are used in a branch-and-bound procedure to minimize the total weighted tardiness. Finally, Artigues (2017) wrote an interesting note to analyze the strength of different time indexed formulations for the resource-constrained project scheduling problem.

In Nonås and Olsen (2005), the main focus was to minimize total tardiness. However, the company in consideration wants to minimize both total tardiness and maximum tardiness. A common choice in the literature when facing different performance indicators is to investigate different MILP formulations with single objective functions in order to find a good compromise among the different indicators. In a recent paper, Samà et al. (2017) discuss multiple performance indicators in the aircraft industry.

They investigate microscopic MILP formulations of the aircraft scheduling problem with different objective functions and examine the differences between the solutions in terms of various performance indicators. We here extend the work of Nonås and Olsen (2005) by proposing a new Mixed Integer Linear Programming model, relying on time indexed binary variables, and by considering different types of solution approaches to balance total tardiness and maximum tardiness. Using the new model, larger problem instances can be solved to optimality and more complex objective functions can be considered. Modeling the problem with the other classical types of binary variables for scheduling problems, sequencing variables as in Ku and Beck (2016) for the job-shop scheduling problem or positional variables as in Dauzère-Pérès (1995) to minimize the number of late jobs on a single machine, seems impossible because of the characteristics of the problem described in the following section. This is discussed in more details in the beginning of Section 3.2.

3. Problem formulation

3.1. Description of the problem

The production of propeller blades is engineer- or made-to-order, and no inventory is maintained. Finished propeller blades are transported by truck or shipped directly to the customer.

Each order consists of a set of identical propeller blades. To produce a propeller blade, a model of the propeller blade is first manufactured in wood by a CAD/CAM process. The wood model is then placed in a two-part metal box, and a sand-fixture solution is used to make a mold of the upper and lower parts of the blade. When the sand has hardened, the model is removed and the upper and lower parts of the box are combined, defining a space that is filled with nickel-aluminum alloy. The box is then moved to a place where the alloy or propeller blade can harden. The time it takes for the alloy to harden depends on the size of the propeller blade, the time increases as the size of the propeller blade increases. The firm uses five different types of boxes ($b = 1, \dots, 5$) to produce a propeller blade. In order to save sand and space, the smallest box that can fit the blade is used. This process is then repeated for every blade of the job. Each job i in the system is defined by the number of blades (O_i) to be produced, the box type b that fits the blade, and the due date (DD_i) for the job.

	Box type (b)	Number (O_i)	Due Date (DD_i)
Job (i)	2	7	8

Table 1. Data for job i.

Due to space limitation and technical product considerations, holding and handling more than one wood model for each box type is difficult. This implies that all blades for a job have to be produced successively. Further, at most one job can be processed for each box type on a given day. An exception occurs when a job does not need all the capacity available on its last production day. A new job of a similar box type can then use the excess capacity.

The company also faces constraints on limited availability of work area, labor, tools and machines. These constraints, in addition to the knowledge of how long a propeller blade needs to harden, are formulated as a set of possible box combination usages on a day. Table 2 lists the combinations of which and how many boxes that can be handled on one day (a shift). An example, selecting box combination 3 in one day means that it is possible to make 1 box of type 1, 3 boxes of type 2, 2 boxes of type 3 and 0 box of types 4 and 5. A different set of box type combinations is considered if the company operates with more than one shift per day.

	Box combinations									
Box type	1	2	3	4	5	6	7	8	9	10
1	7	5	1	2	2	2	0	0	1	0
2	0	0	3	2	0	0	2	1	0	1
3	0	0	2	0	2	2	2	2	0	0
4	0	2	0	1	1	0	1	0	1	1
5	0	0	0	0	0	1	0	1	1	1

Table 2. Capacity restrictions: Allowed combinations of boxes.

The main objective for the company is to find a production schedule that satisfies the production requirements and minimizes total tardiness while keeping the maximum tardiness low.

Example: A simplified production schedule

To illustrate a possible schedule, the data for five different jobs can be found below, where each job is defined by its specified box type, the number of propeller blades to produce and its due date.

Job i	Box type b	Number O_i	Due Date DD_i
1	1	5	1
2	2	4	3
3	2	4	4
4	3	8	5
5	4	6	5

The schedule below includes for each day, which box combination to use and which jobs to produce in the selected combination of boxes.

Day	Box combination					Scheduled jobs				
d	b=1	b=2	b=3	b=4	b=5	b=1	b=2	b=3	b=4	b=5
1	5	0	0	2	0	1			5	
2	0	2	2	1	0		2	4	5	
3	0	2	2	1	0		2	4	5	
4	0	2	2	1	0		3	4	5	
5	0	2	2	1	0		3	4	5	

From the schedule above, we can see that at day 3, box combination 7 is used, and jobs 2, 4, and 5 are scheduled with respectively, 2, 2 and 1 propeller blades. Furthermore, note that job 4 is scheduled at days 2, 3, 4, and 5, with two propeller blades in each day. Note also that only job 3 is tardy in the schedule.

3.2. Outline of the new model

The Mixed Integer Linear Programming (MILP) model outlined in Nonås and Olsen (2005) uses binary variables β_{ij} to indicate if job i is scheduled at day j. In the new model, binary decision variables are used to indicate the first and last production days of job i:

- $s_{ij} \in \{1, 0\}$ - s_{ij} is equal to 1 if the processing of job i starts at day j, and 0 otherwise,
- $e_{ij} \in \{1, 0\}$ - e_{ij} is equal to 1 if the processing of job i ends at day j, and 0 otherwise.

By using a time indexed formulation, the number of variables in the problem is doubled, but solution times are strongly decreased as shown in the computational experiments. As discussed in Section 2, time indexed formulations are known to give strong lower bounds for the linear relaxation of the MILP model (Dyer and Wolsey (1990) and Queyranne and Schulz (1994)). Moreover, because of the choice of one and only one box combination per day, we do not believe it is possible to model the problem using sequencing or positional binary variables. More precisely, time indexed binary variables are required to decide if a box combination is selected in a day, and temporal constraints (see model below) are required to ensure that only one box combination is allowed per day and that the jobs performed in a day satisfy the selected box combination.

The full menu of parameters and decision variables for the new formulation is summarized below. We use uppercase letters for parameters and lowercase letters for variables.

Parameters:

J^b	Set of jobs to be produced in box type b,
N	Number of jobs,
B	Number of box types,
K	Number of box combinations,
D	Upper bound on the number of days needed to schedule all jobs,
O_i	Number of propeller blades requested for job i,
DD_i	Date when the processing of job i is due to be completed (*due date*),
$COMB_k^b$	Number of boxes of type b in box combination k.

Decision variables:

c_i	*Completion day* of job i,
t_i	*Tardiness* of job i, $t_i = \max\{0, c_i - DD_i\}$,
x_{kj}	Indicator variable that flags if box combination k is used at day j,
s_{ij}	Indicator variable that flags if job i starts at day j,
e_{ij}	Indicator variable that flags if job i ends at day j,
$\delta_{i,j}$	Number of propeller blades of job i scheduled on day j.

The mutually exlusive sets J^b, $b = 1, \ldots, B$ partition the set $J = \{1, \ldots, N\}$. Thus, $\bigcup_{b=1}^{B} J^b = J$, $J^{b'} \bigcap J^b = \emptyset$ for all $b, b' \in \{1, \ldots, B\}$ where $b \neq b'$. Further, we assume that the jobs in each set J^b are ordered according to their due dates. Let us first assume an objective function (1) that minimizes total tardiness (as in Nonås and Olsen (2005)). Other objectives will be consider later in this paper.

The new time indexed based formulation can then be formulated as follows:

$$\left\{\begin{array}{lll}
\min \sum_{i=1}^{N} t_i & & (1)\\
\sum_{d=1}^{D} s_{id} = 1 & i = 1,\dots,N & (2)\\
\sum_{d=1}^{D} e_{id} = 1 & i = 1,\dots,N & (3)\\
\sum_{j=1}^{d} e_{ij} - \sum_{j=1}^{d} s_{ij} \leq 0 & i = 1,\dots,N,\ d = 1,\dots,D & (4)\\
\delta_{id} - \sum_{j=1}^{d} s_{ij} + \sum_{j=1}^{d-1} e_{ij} \geq 0 & i = 1,\dots,N,\ d = 1,\dots,D & (5)\\
\sum_{j=1}^{d} \delta_{ij} - O_i \sum_{j=1}^{d} s_{ij} \leq 0 & i = 1,\dots,N,\ d = 1,\dots,D & (6)\\
\sum_{j=d}^{D} \delta_{ij} - O_i \sum_{j=d}^{D} e_{ij} \leq 0 & i = 1,\dots,N,\ d = 1,\dots,D & (7)\\
\sum_{d=1}^{D} \delta_{id} = O_i & i = 1,\dots,N & (8)\\
\sum_{i\in J^b} (\sum_{j=1}^{d} s_{ij} - \sum_{j=1}^{d} e_{ij}) \leq 1 & b = 1,\dots,B,\ d = 1,\dots,D & (9)\\
(\sum_{j=1}^{d-1} s_{ij} - \sum_{j=1}^{d-1} e_{ij}) + (\sum_{j=1}^{d} s_{ij} - \sum_{j=1}^{d} e_{ij}) + s_{kd} \leq 2 & b = 1,\dots,B,\ d = 2,\dots,D, & (10)\\
 & \forall\ i \in J^b, \forall\ k \in J^b, k \neq i & \\
\sum_{i\in J^b} (e_{i,d} - s_{i,d}) + \sum_{i\in J^b} s_{i,d} \leq 2, & d = 1,\dots,D & (11)\\
\sum_{k=1}^{K} x_{kd} = 1 & d = 1,\dots,D & (12)\\
\sum_{i\in J^b} \delta_{id} - \sum_{k=1}^{K} COMB_k^b x_{kd} \leq 0 & b = 1,\dots,B,\ d = 1,\dots,D & (13)\\
c_i - \sum_{d=1}^{D} e_{id} d = 0 & i = 1,\dots,N & (14)\\
t_i - c_i + DD_i \geq 0 & i = 1,\dots,N & (15)\\
s_{id},\ e_{id} \in \{0,1\},\ \delta_{id},\ c_i,\ t_i \geq 0 & i = 1,\dots,N,\ d = 1,\dots,D & (16)\\
x_{kd} \in \{0,1\} & k = 1,\dots,K,\ d = 1,\dots,D & (17)
\end{array}\right.$$

Constraint (2) ensures that each job should start on one and only one day, and Constraint (3) that each job should end on one and only one day. Constraint (4) ensures that a job cannot end before starting. This constraint is not necessary but helps to tighten the formulation to get better linear programming relaxation bounds. Constraint (5) guarantees that each job must be produced between its start day and end day. Constraint (6) ensures that no production can be performed before a job starts, and Constraint (7) that no production can be performed after a job ends. Constraint (8) ensures that all propeller blades in each job should be manufactured. Due to limited storage space and specific production requirements, two jobs of the same box type cannot be scheduled in parallel. Constraints (9) and (10) ensure that there are no jobs in parallel for the same box type. Constraint (9) makes sure that at most one job of the same box type can start at or before day d and is completed after day d. Constraint (10) makes sure that we have no parallel production in a single day, if there is a job that starts at or before day $d-1$ and is completed at or after day $d+1$, then there is no other job of the same box type that can start and be completed at day d. Constraint (11) allows only two jobs of the same box type to be started and completed in the same day. In addition, Constraint (11) makes sure that, in the completion date d of job j, only one job extra of the same box type can be scheduled. Constraint (12) ensures that one and only one combination is chosen in each day. Constraint (13) guarantees that the number of blades of a given box type produced in a given day has to be lower than or equal to the available capacity. Constraint (14) determines the completion times, and Constraint (15) the tardiness.

3.3. Avoid orders scheduled too early

Due to limited storage space both at the foundry and at its customer sites, it may sometimes be necessary to limit the number of days an order can be produced ahead of its due date. To ensure that no jobs are scheduled too early, the following constraint can be included in the new time indexed formulation:

$$c_i \geq DD_i - early_i,\ i = 1, \ldots, N \qquad (18)$$

where $early_i$ is the number of days job i can be scheduled ahead of its due date (DD_i). Constraint (18) cannot be included in the mathematical model of Nonås and Olsen (2005) because the completion date is accurate only for tardy jobs (jobs with a positive tardiness). The completion date in Nonås and Olsen (2005) is only determined by one larger than or equal to expression:

$$c_i \geq j(\beta_{i,j} - \beta_{i,j+1}),\ \ j = 1, \ldots, D-1,\ \ i = 1, \ldots, N.$$

This is in contrast to the new model, where Constraint (14) forces the completion dates to be accurate. It is not straightforward to force the completion dates in Nonås and Olsen (2005) to be accurate by including a similar constraint.

3.4. Valid inequalities

To reduce the times required to solve the new time indexed formulation, we also propose and test three different valid inequalities which rely on the following arguments:

1. A job i cannot end before it starts. If a job i is scheduled from or in day j, i.e. $s_{ij} = 1$, then the job has to be completed at a day k where $k \geq j$. We then have:
   $\sum_{j=1}^{d} e_{ij} - \sum_{j=1}^{d} s_{ij} \leq 0,\ i = 1, \ldots, N,\ d = 1, \ldots, D$ (end)

2. The number of propeller blades scheduled for job i in day j, $i \in J^b$, has to be lower than or equal to the capacity of box type b at day j. If $maxcap_b$ is the maximum allowed capacity for box-type b, the second valid inequality can be written:
   $\delta_{ij} \leq maxcap_b$, for all jobs $i \in J^b$ (box)
   where $maxcap_b = max_{k=1,...,K} COMB_b^k,\ b = 1, \ldots, B$.
3. If $mincap_b$ and $maxcap_b$ are, respectively, the minimum number and maximum number of boxes available per day for a job i of box type b, the total number of days necessary to produce job i is limited from below by $O_i/maxcap_b$ days and from above by $O_i/mincap_b$ days. This is formalized in the two following constraints:
   $\sum_{d=1}^{D} e_{id} - \sum_{d=1}^{D} s_{id} \leq \lceil O_i/mincap_b \rceil$, $\forall i \in J^b$, $b = 1, ...B$ (mm)
   $\sum_{d=1}^{D} e_{id} - \sum_{d=1}^{D} s_{id} \geq \lfloor O_i/maxcap_b \rfloor$, $\forall i \in J^b$, $b = 1, ...B$
   where $mincap_b = min_{k=1,...,K} COMB_b^k$ and $maxcap_b = max_{k=1,...,K} COMB_b^k$, $b = 1, \ldots, B$.

4. Multiobjective problem formulation

In addition to minimizing total tardiness, the manufacturer would like to avoid jobs with an extremely long tardiness, that would make customers unhappy and might impact future demands. Minimizing total tardiness and minimizing maximum tardiness can be conflicting objectives, so finding a formulation were both are optimized is not possible. Nonås and Olsen (2005) suggest to balance the two goals in their heuristic solution approach by either:

1. Penalizing jobs with a tardiness above an upper limit, or
2. Minimizing the sum of the square of the tardiness.

In the first approach, the foundry has to find a suitable upper limit on the tardiness and an appropriate penalty for jobs with a tardiness above this limit. There was no discussion, or advice given, on how to find these values. Here, we suggest to balance the two goals in the new time indexed formulation by implementing one of the following extensions to the basic formulation.

1) A new constraint that limits the tardiness for each job i:

$$\begin{gathered}\text{minimize } \sum_{i=1}^{N} t_i \\ \text{s.t. } t_i \leq kT,\ i = 1, \ldots, N \\ \text{Constraints (2) to (16),}\end{gathered}$$

where T is a parameter specific for the problem instance considered and k is a positive value larger than or equal to 1 that emphasizes the importance of a low maximum tardiness.

2) A new objective function that minimizes the weighted sum of the total tardiness and maximum tardiness:

$$\begin{gathered}\text{minimize } \textstyle\sum_{i=1}^{N} t_i + kt_{max}\\ \text{s.t. } t_i \leq t_{max},\ i = 1,\ldots,N\\ \text{Constraints (2) to (16),}\end{gathered}$$

where t_{max} is a decision variable and k is a positive parameter that illustrates the importance of a low maximum tardiness compared to a low total tardiness.

3) Implicitly include both criteria in the objective function and add upper bounds on the individual tardiness to reduce solution time:

$$\begin{gathered}\text{minimize } \textstyle\sum_{i=1}^{N} t_i^2\\ \text{s.t. } t_i \leq kT,\ i = 1,\ldots,N\\ \text{Constraints (2) to (16),}\end{gathered}$$

where T is a parameter specific for the problem instance considered and k is a positive value larger than or equal to 1 that emphasizes the importance of a low maximum tardiness. Note that most standard solvers, such as IBM ILOG CPLEX which is used in our computational experiments, can now handle quadratic objective functions if the constraints are linear.

The parameter $T = t_{max}$ where t_{max} is determined by solving the problem,

$$\begin{gathered}\text{minimize } t_{max}\\ \text{s.t. } t_i \leq t_{max},\ i = 1,\ldots,N\\ \text{Constraints (2) to (16).}\end{gathered}$$

In the next section, the performance of the three extensions are analyzed and compared to the basic case of minimizing total tardiness. Both solution times and values are reported. The values of k are chosen to illustrate how the solution values change when the different extensions are added to the basic formulation. Section 6 includes a more detailed discussion on how to find appropriate values for the parameter k depending on the priorities of the factory. In addition to the three extensions proposed above, we also briefly discuss the effect of including an upper bound on the tardiness in Extension 2 ($t_i \leq kT$, $i = 1,\ldots,N$).

An alternative to Extension 2 could be to minimize the sum of the relative deviations from the minimum total tardiness and the minimum maximum tardiness. However, that requires solving the problem both for maximum tardiness and for total tardiness, and would thus increase the solution time significantly, since finding the minimum total tardiness is time consuming. In addition, the relative deviation may not be the right measure as minimizing total tardiness and avoiding large maximum tardiness are the major goals.

5. Computational experiments and analysis

In this section, we first present the significant reduction in solution time obtained with the new time indexed formulation without the valid inequalities. Second, we study which valid inequalities to add to the new time indexed formulation to further improve the solution times. Finally, we present the performance of the different extensions related to the balance between the total tardiness and maximum tardiness.

We consider both the case where there are no restrictions on how early a job can be scheduled and the case where there are limitations on how many days an order can be scheduled ahead of its due date. For the last case, only results for the new model are reported. As pointed out in Section 3.3, it is not straightforward to include lower bounds on the completion dates in the old formulation.

We report results for:

- The basic mathematical model where the objective is to minimize total tardiness,
- The extension where the objective is to minimize total tardiness given restrictions on the maximum tardiness (Extension 1, $k = 1$ and $k = 1.5$),
- The extension where the objective is to minimize the sum of the maximum tardiness and total tardiness (Extension 2, $k = 1$),
- The extension where the objective is to minimize the sum of the square of the tardiness subject to an upper bound on the maximum tardiness (Extension 3, $k = 1.5$).

We consider two different sets of problem instances, both generated from the set of problem instances in Nonås and Olsen (2005). The problem instances in the two sets differ in the length of the time horizon, the usage rate of the capacity, and how balanced the demand is over the different box types. For the instances in the first problem set (Set 1), we have a time horizon of around 32 days and a very high usage rate of the available capacity. The number of jobs for an instance varies between 16 and 23, while the number of days varies between 30 and 34. Compared to the first set, the instances in the second set have a lower usage rate relative to the time horizon, a more unbalanced usage of the different box types, and a longer time horizon (around 57 days). The size of the problem instances ranges from 43 to 70 days and from 14 to 30 jobs. To give an idea of which type of problems instances we consider in the two problem sets, Tables B5 through B8 provide the specific data for one problem instance from each problem set. Tables B1 through B4 provide an overview of the usage rates for Sets 1 and 2, and the minimum and maximum order sizes for the different box types. The problem instances are solved using AMPL (IBM ILOG CPLEX 12.9) on a Dell computer with an Intel Core i5-8500T CPU @2.10GHz processor. We used a time limit of 2,500 seconds for each instance. "TL" indicates that optimality is not proven within the time limit of 2,500 seconds. The notation $\lceil 1.5T \rceil$

means that $1.5T$ is rounded up to the closest integer. We have 22 problem instances in Set 1 and 16 problem instances in Set 2. When comparing the new and the old mathematical models, we do not include the lower bound on the completion time (Constraint (18)) in the new model. However, Constraint (18) is later included when we analyze the effect of limited storage capacity on the solution time and solution values.

5.1. Improvements in solution times

We here present the average solution times for the two data sets for the basic case of minimizing total tardiness and for the three different extensions to the basic case. We also present the number of instances not solved to optimality within the time limit for each extension. The average solution times are calculated based on the time limit for these instances. The individual solution times are given in Appendix A. For Extension 1 and Extension 3, the average solution time to find the minimal maximum value T is not included in the solution time reported in the tables. The average solution time to find T is however given explicitly in the text above each table.

Note that, since they are calculated based on the time limit for one or more instances, the average solution times are lower bounds of the real average solution times which take the real solution times of all instances into account..

5.1.1. Instances from Set 1

The average solution times for the instances in Set 1 are given for the new and the old models in the first two rows in Table 3. The minimum and maximum solution times for the instances are given in the next two rows, while the two last rows show the number of problems not solved to optimality for each extension. Note that the average solution times for the new model are significantly lower than the average solution times for the old model. The difference in average solution times is particularly large for the Basic case and for Extension 2 (i.e., for the formulations with no bounds on the individual tardiness). For Extension 2, the old model only solves 4 out of 22 instances to optimality within the time limit, while the new model solves 21 of 22 instances to optimality. The average solution time to determine T is 191 seconds with the new formulation and 236 seconds with the old formulation, see Appendix A (Table A8) for the individual solution times.

		Basic case	Extension 1 k=1, k=1.5		Extension 2	Extension 3
	MILP model	min$\sum t_i$	min$\sum t_i$ $t_i \leq T$	min$\sum t_i$ $t_i \leq \lceil 1.5T \rceil$	min$\sum t_i + t_{max}$ $t_i \leq t_{max}$	min$\sum t_i^2$ $t_i \leq \lceil 1.5T \rceil$
Aver. time	New	398	160	213	474	2073
	Old	2143	238	659	2286	2314
Min, Max	New	[32, TL]	[2, 2495]	[5, TL]	[69, TL]	[2, TL]
	Old	[27, TL]	[1, TL]	[0, TL]	[93, TL]	[0, TL]
Not solved	New	1	0	1	1	16
within TL	Old	15	1	3	18	20

Table 3. Overview of solution times for Set 1.

We observe further that, for both models, the average solution time decreases from the Basic case to Extension 1 (k=1, k=1.5) as the upper bounds on $(t_i,\ i,\dots,N)$ get tighter, as expected due to a smaller feasible region. However, for the individual problem instances, we note that the solution times vary depending on how hard it is to get a schedule that satisfies the requirement on the maximum tardiness (see Appendix A for individual solution times). The average solution time significantly increases from Extension 1 (k=1.5), which minimizes a linear objective function, to Extension 3 (k=1.5), where a quadratic objective function is minimized, which is also expected due to a more complex objective function. Extending the Basic case by using both criteria in the objective function (Extension 2) did not result in any significant change in the average solution time. If we also limit the tardiness in Extension 2, the average solution time for the new model is reduced to 236 seconds by adding the constraint $(t_i \le \lceil 1.5T \rceil, i = 1,\dots,N)$ and to 350 seconds when adding a less tight constraint $(t_i \le 2T, i = 1,\dots,N)$. However, when adding the strongest constraint $(t_i \le \lceil 1.5T \rceil, i = 1,\dots,N)$ to Extension 2, the upper bound on the tardiness becomes binding in three instances. When the constraint is less tight, $(t_i \le 2T, i = 1,\dots,N)$, the upper bound becomes binding for at least one job in one instance. With a larger weight $k = 2$ on the maximum tardiness in Extension 2 $(\min \sum_{i=1}^{N} t_i + 2t_{max})$, the average solution time increases from 474 to 609 seconds. One instance is not solved to optimality within the time limit. If we further restrict the maximum tardiness to be lower than $\lceil 1.5T \rceil$, the average solution time is reduced to 262. No upper bounds are binding. The individual solutions and the solution times can be found in Appendix A, see Tables A1, A4 and A7. Adding the valid inequalities proposed in Section 3.4 to the Basic case, and the extensions to the Basic case, do not significant impact the average solution times for Set 1, see Section 5.4 for more details.

5.1.2. Instances from Set 2

The average solution times for the problem instances from Set 2 can be found in the first two rows of Table 4. The next two rows show the minimum and maximum solution times, while the two last rows show the number of instances not solved to optimality within the time limit. We see also here that the new time indexed formulation is superior compared to the formulation proposed by Nonås and Olsen (2005) for the Basic case, Extension 1 and Extension 2. For Extension 3, the old formulation performs best. However, if the valid inequality (end) is added, the new time indexed formulation also performs best for Extension 3, see Section 5.4. The average solution time to optimize T is respectively 114 seconds for the new formulation and 303 seconds for the old formulation. Using the old formulation, one instance is not solved to optimality within the time limit.

		Basic case	Extension 1 k=1, k=1.5		Extension 2	Extension 3
	MILP model	min$\sum t_i$	min$\sum t_i$ $t_i \leq T$	min$\sum t_i$ $t_i \leq \lceil 1.5T \rceil$	min$\sum t_i + t_{max}$ $t_i \leq t_{max}$	min$\sum t_i^2$ $t_i \leq \lceil 1.5T \rceil$
Aver. time	New	366	44	77	694	1551
	Old	1218	159	429	1473	1478
Min, Max	New	[9, TL]	[3, 366]	[3, 410]	[9, TL]	[4, TL]
	Old	[4, TL]	[0, 842]	[0, TL]	[5, TL]	[1, TL]
Not solved	New	2	0	0	2	9
within TL	Old	5	0	2	7	7

Table 4. Overview of solution times for Set 2.

For Set 2, as for Set 1, the average solution time for the new formulation is significantly reduced from the Basic case to the case where we have added upper bounds on how tardy a job can be (Extension 1), and is significantly increased from Extension 1 (k=1.5) to the case with a quadratic objective function (Extension 3, k=1.5). For Extension 1, note that the average solution time decreases as the upper bound on the maximum tardiness decreases. For the individual problem instances, however, we also observe that the solution times vary depending on how hard it is to get a schedule that satisfies the requirement on the maximum tardiness (see Appendix A for individual solution times). If, in Extension 2, the tardiness is constrained to be lower than or equal to 1.5 times the minimal maximum tardiness, the average solution time for the new formulation decreases from 694 to 308 seconds, the upper bound on t_i, $i = 1, \ldots, N$, is not binding for any of the problem instances. All instances are then solved to optimality within the time limit. If we have a larger weight on the maximum tardiness in Extension 2 (i.e, min $\sum_{i=1}^{N} t_i + 2t_{max}$), the average solution time increases from 694 to 718 seconds. However, if the constraint $t_i \leq \lceil 1.5T \rceil, \forall i$, is also added, the average solution time is reduced to 316 with only one instance not solved to optimality within the time limit. No upper bounds are binding. Individual solutions and solution times can be found in Appendix A.

If the valid inequality (end) is added to Extension 3, and the valid inequality (mm) is added to the Basic case and Extension 2, the average solution times for instances in Set 2 are significantly reduced. For the Basic case and Extension 2, all instances in Set 2 are solved to optimality within the time limit. For Extension 3, three more instances are solved to optimality within the time limit, i.e. one more instance than with the old model, see Section 5.4 and Appendix C for the individual results.

5.2. Performance of the different extensions

We now present the performance of the different extensions to our model with regards to the balance between total tardiness and maximum tardiness. We consider instances from Set 1 and Set 2 and report how much the total tardiness and maximum tardiness change if we implement one of the extensions instead of solving the basic model. An instance is marked with "TL" if optimality was not reached (or proved) within the time limit for either the basic model or the considered extension. An instance is marked with a "*" if there are multiple solutions to the basic model. The reduction in t_{max} may then be larger than or lower than the solution reported in the tables.

The actual values of the maximum tardiness and total tardiness are given in Appendix A.

5.2.1. Instances from Set 1

We first present results in Tables 5-8 that indicate how much the solution values change for problem instances in Set 1. The first row in the tables gives the reduction on the maximal tardiness, while the second row gives the increase of the total tardiness. If we extend the basic formulation (Basic case) to only accept solutions with tardiness lower than or equal to its minimal maximal value (Extension 1, $k = 1$), we see from Table 5 that the increase in total tardiness varies a lot, from the instances where the increase in total tardiness is significant relative to the decrease in maximum tardiness (instances 4, 7, 8, 11, 14), to the instances where the increase in total tardiness is minor relative to the decrease in maximum tardiness (instances 19, 20). The increase in total tardiness is huge for instance 14. A large order is here placed in front of many small orders with tight due dates in order to satisfy the tight upper bound on the individual tardiness.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
t_{max}	-16	-2	-7	-4	-7	-1	-4	-2	-6	0	-2	-2	-7	-13	-1*	TL	-2	-2	-4	-6	-2	-4
$\sum t_i$	12	2	6	10	8	2	8	10	7	0	6	2	5	40	0*	TL	1	4	1	2	1	4

Table 5. Changes in maximum tardiness and total tardiness, $\min \sum_{i=1}^{N} t_i$ s.t. $t_i \leq T$, $i = 1, \ldots, N$.

If the restriction on the maximum tardiness is less tight (Extension 1, $k = 1.5$), the deviation from the basic case will be lower both for the maximum tardiness and the total tardiness (see Table 6). In particular, the increase in total tardiness for instance 14 is lower due to a looser upper bound on the individual tardiness.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
t_{max}	-13	-1*	-4	-1	-3	0	0	0	-3	0	0	-1*	-6	-9	0	TL	0	0	-3	-6	0	-3*
$\sum t_i$	3	0*	1	2	2	0	0	0	5	0	0	0*	1	20	0	TL	0	0	1	2	0	0*

Table 6. Changes in maximum tardiness and total tardiness, $\min \sum_{i=1}^{N} t_i$, s.t. $t_i \leq \lceil 1.5T \rceil$, $i = 1, \ldots, N$.

If the basic formulation is extended to minimize the sum of the total tardiness and maximum tardiness (Extension 2, $k = 1$), we get a good balance between both criteria in general, and we always get the lowest maximum tardiness for equal value of the total tardiness (see Table 7). If we compare the results from Table 7 ($\min \sum_{i=1}^{N} t_i + t_{max}$), with the results from Table 6 ($\min \sum_{i=1}^{N} t_i$, $t_i \leq \lceil 1.5T \rceil$), we see that we get a larger reduction in maximum tardiness for the same increase in total tardiness for the same instances (5 ,15, 19). For instance 14, note that we get the same production plan as for the basic case (see Appendix A for the real values).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
t_{max}	-13	-1*	-4	0	-6	0	0	0	0	0	-1	-1	-6	0	-1*	TL	-2	-1*	-4	-6	-2	-3*
$\sum t_i$	3	0*	1	0	2	0	0	0	0	0	1	0	1	0	0*	TL	1	0*	1	2	1	0*

Table 7. Changes in maximum tardiness and total tardiness, $\min \sum_{i=1}^{N} t_i + t_{max}$.

If the basic formulation is extended by considering a quadratic objective function with upper bounds on the tardiness (Extension 3, $k = 1.5$), we may get a relatively large increase of the total tardiness for the reduction of the maximum tardiness (see Table 8, instances 7 and 9). The reason is that the quadratic objective function focuses on reducing the tardiness for all jobs with a relatively large tardiness at the expense of a larger total tardiness. The quadratic objective may also result in solutions with a really low maximum tardiness, for instance in Set 1, $t_{max} = T$ for four instances (3, 9, 10, 20).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
t_{max}	TL	TL	-7	TL	-6	TL	-2	0	-6	0	TL	TL	TL	TL	TL	TL	TL	TL	-3	-6	TL	TL
$\sum t_i$	TL	TL	9	TL	2	TL	9	0	10	0	TL	TL	TL	TL	TL	TL	TL	TL	1	2	TL	TL

Table 8. Changes in maximum tardiness and total tardiness, $\min \sum_{i=1}^{N} t_i^2$, $t_i \leq \lceil 1.5T \rceil$, $i = 1, \dots, N$.

The performance of the different extensions can be summarized very well by looking at instance 1. If the basic model is extended to include tight upper bounds on the maximum tardiness (Extension 1, $t_i \leq T$) we get a large decrease in maximum tardiness and a corresponding large increase in total tardiness. When the upper bound on the tardiness is less tight (Extension 1, $t_i \leq \lceil 1.5T \rceil$), the changes in the maximum tardiness and the total tardiness get smaller. By taking into account both the maximum tardiness and total tardiness in the objective function (Extension 2, k=1), we get a solution where the maximum tardiness is reduced compared to (Extension 1, $t_i \leq \lceil 1.5T \rceil$) while the total tardiness is unchanged. And at last, using a quadratic objective function (Extension 3, $t_i \leq \lceil 1.5T \rceil$), we get a maximum tardiness that is equal to its minimal value and a total tardiness that is larger than the total tardiness for all the other extensions.

5.2.2. Instances from Set 2

We now look at how the total tardiness and maximum tardiness change when we consider problem instances from Set 2, that is for problem instances that generate schedules with relatively few tardy jobs and with spare capacity.

Table 9 presents how much the total tardiness increases from the Basic case when the maximum tardiness is constrained by its minimum value (Extension 1, $k = 1$). Note that the increase in total tardiness relative to the decrease in maximum tardiness varies from a relatively small increase in total tardiness (instance 8), to a relatively large increase in total tardiness (instance 14). The increase of the total tardiness is largest for instance 14. A large order is here scheduled before three small orders with tight due dates in order to satisfy the tight upper bound on the individual tardiness.

For seven of the instances, we obtain the same total tardiness as for the basic case. This is due to excess capacity in the optimal schedules.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
t_{max}	0	0	-12	-1	0	0	-4	-8	-3	-3	0	-3	0	-4	-3	-3*
$\sum t_i$	0	0	13	3	0	0	4	2	4	19	0	9	0	19	2	0*

Table 9. Changes in maximum tardiness and total tardiness, $\min \sum_{i=1}^{N} t_i$, $t_i \leq T$, $i = 1, \ldots, N$.

If we accept a larger value of maximum tardiness (Extension 1, $k = 1.5$), both the reduction in maximum tardiness and the increase in total tardiness can be smaller (instances 3, 9, 10). Note from Table 10 that, for most of the instances, the same solution as for the basic case is obtained. This is mainly due to a small number of tardy jobs and more capacity than needed for the current set of jobs. The upper bound on the maximum tardiness is only binding for a few of the instances.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
t_{max}	0	0	-9	0	0	0	0	0	-2*	-1*	0	0	0	0	0	-3*
$\sum t_i$	0	0	4	0	0	0	0	0	0*	0*	0	0	0	0	0	0*

Table 10. Changes in maximum tardiness and total tardiness, $\min \sum_{i=1}^{N} t_i$, $t_i \leq \lceil 1.5T \rceil$, $i = 1, \ldots, N$.

If the basic case is extended to minimize the sum of the total tardiness and maximum tardiness (Extension 2), we will in general get a lower maximum tardiness at the expence of an increased total tardiness. However, note that, from Table 11, most of the instances terminate with the same total tardiness and maximum tardiness as for the basic case of minimizing total tardiness. The exceptions are Instances 3, 7 and 8. Instances 9, 10, 15 and 16 terminate with the same total tardiness but with a lower maximum tardiness (multiple optimal solutions in the basic case). As for Extension 1 ($k = 1.5$), the results are the same as the ones obtained in the Basic case, mainly due to a small number of tardy jobs and more capacity than needed for the current set of jobs. Note that Extension 2 ($\min \sum_{i=1}^{N} t_i + t_{max}$) will, in contrast to Extension 1 ($\min \sum_{i=1}^{N} t_i$, $t_i \leq kT$, $i = 1, \ldots, N$),

always terminate with the lowest maximum tardiness for a given total tardiness, since the maximum tardiness is included in the objective function.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
t_{max}	0	0	-9	0	0	0	-4	-7	-2*	-1*	0	0	0	0	-2*	-3*
$\sum t_i$	0	0	4	0	0	0	4	1	0*	0*	0	0	0	0	0*	0*

Table 11. Changes in maximum tardiness and total tardiness, $\min \sum_{i=1}^{N} t_i + t_{max}$.

If the Basic case is extended to include a quadratic objective function (Extension 2, $k = 1.5$), note that, again, the same solution as for the Basic case is obtained for most of the instances, see Table 12. For some instances, there is a small increase of the total tardiness for a larger decrease of the maximum tardiness. The upper bound on the tardiness is not binding for any of the instances.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
t_{max}	0	0	TL	0	0	TL	-1	TL	-2	TL	0	0	0	TL	-3	-3
$\sum t_i$	0	0	TL	1	0	TL	1	TL	0	TL	0	3	0	TL	2	0

Table 12. Changes in maximum tardiness and total tardiness, $\min \sum_{i=1}^{N} t_i^2$, $t_i \leq \lceil 1.5T \rceil$.

For most of the problem instances in Set 2, the values of the total tardiness and maximum tardiness are approximately equal regardless of which extension is used. This is mainly due to a small number of tardy jobs and more capacity than needed for the current set of jobs.

5.3. Avoid jobs scheduled too early

In this section, we analyze how the solution time and the solution quality (total tardiness and maximum tardiness) change when we include restrictions on how many days a job can be scheduled ahead of its due date. We only report results for the new mathematical model. As discussed earlier, the completion dates in the old formulation are only accurate for tardy jobs, and Constraint (18) cannot easily be included in the old formulation. Since the effect of adding restrictions on completion dates has a larger impact on the solution values for instances with tight schedules, we present data only for the first problem set (Set 1). The average solution times when we include Constraint (18) with parameter $early_i = 3$, $i = 1, \dots, 22$, in the Basic case and the three suggested extensions are presented in the first row in Table 13. In the second row, the minimum and maximum solution times for the instances are given, while the last row shows the number of instances not solved to optimality within the time limit.

	Basic case	Extension 1 k=1, k=1.5		Extension 2	Extension 3
MILP model	$\min \sum t_i$	$\min \sum t_i$ $t_i \leq T$	$\min \sum t_i$ $t_i \leq \lceil 1.5T \rceil$	$\min \sum t_i + t_{max}$ $t_i \leq t_{max}$	$\min \sum t_i^2$ $t_i \leq \lceil 1.5T \rceil$
Avg. solution time	124	13	32	227	663
Min, Max	[4, 1529]	[1, 83]	[2, 297]	[2, TL]	[2, TL]
Not solved within TL	0	0	0	1	3

Table 13. Average solution times for the instances in Set 1, $c_i \geq DD_i - 3, i = 1, \dots, 22$.

Note that when constraint $c_i \geq DD_i - 3, \forall i$, is added, the average solution times decrease for all the extensions and more instances are solved to optimality within the time limit (see Appendix A for individual solution times). This is expected due to a smaller feasible region. Note also that, when Constraint $c_i \geq DD_i - 3$, $\forall i$, is added, the total tardiness increases for all but three instances (Instances 11 and 22, Extension 1 (k=1), and Instance 7, Extension 3), and the maximum tardiness increases for all but seven instances (Instances 1 and 9, Extension 2, and Instances 5, 9, 13, 19 and 22, Base case), see Appendix A, Table A11. The maximum tardiness may decrease since the opportunity to push small jobs in front of larger jobs to minimize total tardiness is reduced. The change in total tardiness and maximum tardiness when constraint $c_i \geq DD_i - 3, \forall i$, is added to (Extension 2, $k = 1$) is given in Table 14. We see that the total tardiness increases for all instances, while the change in the maximum tardiness is negative for some instances and positive for other instances. For the real values of the maximal and total tardiness, see the 6th column in Tables A4 and A5.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
t_{max}	-2	1	1	1	0	0	0	0	-6	0	0	3	1	1	1	TL	1	4	0	1	3	0
$\sum t_i$	5	3	4	2	4	5	1	1	8	0	2	8	1	9	3	TL	5	15	1	2	8	3

Table 14: Changes in the solution values. Set 1, Extension 2, adding $c_i \geq DD_i - 3, \forall i$.

The performance of the three extensions relative to each other follows the same pattern as for the case where no restrictions on completion dates are considered. In Appendix A, we present how much the total tardiness and maximum tardiness change from the basic case for each of the three extensions when constraint $c_i \geq DD_i - 3, \forall i$, is added.

If the valid inequality (end) is added to Extension 3, and the valid inequality (mm) is added to the Basic case and Extension 2, the average solution times for the instances in Set 1 are significantly reduced. For Extension 2, all the instances are solved to optimality within the time limit and, for Extension 3, one additional instance is solved to optimality within the time limit, see Section 5.4 and Appendix C for individual results.

5.4. Solution times with valid inequalities

This section presents and discusses, for both data sets, the average solution times when the valid inequalities presented in Section 3.4 are added to the Basic case and the three extensions to the Basic case. The minimum and maximum solution times are also presented, and the number of instances not solved to optimality within the time limit. The average solution times are again calculated based on the time limit for these instances.

Tables 15 and 16 show the average solution times for the instances in Set 1, for respectively the case without and the case with constraints on how early a job can be scheduled. Table 17 shows the average solution times for the instances in Set 2. The individual solution times can be found in Appendix C. The first three rows in each table present the average solution times for the specified data set when the valid inequality specified in the second column is added to the Basic case and the three extensions to the Basic case. The next three rows specify the minimum and maximum solution times. The last three rows give the number of instances not solved to optimality within the time limit for each valid inequality.

		Basic case	Extension 1 k=1, k=1.5		Extension 2	Extension 3
	MILP	$\min\sum t_i$	$\min\sum t_i$	$\min\sum t_i$	$\min\sum t_i + t_{max}$	$\min\sum t_i^2$
	add on		$t_i \leq T$	$t_i \leq \lceil 1.5T \rceil$	$t_i \leq t_{max}$	$t_i \leq \lceil 1.5T \rceil$
Aver. Time	(end)	416	193	202	584	2014
	(box)	377	158	218	482	2073
	(mm)	413	87	210	488	2128
Min, Max	(end)	[54, TL]	[2, TL]	[3, TL]	[65, TL]	[2, TL]
	(box)	[33, TL]	[2, 2360]	[1, TL]	[69, TL]	[2, TL]
	(mm)	[14, TL]	[2, 823]	[2, TL]	[61, TL]	[2, TL]
Not solved	(end)	1	1	1	2	16
within TL	(box)	1	0	1	1	16
	(mm)	1	0	1	1	18

Table 15. Overview of solution times for Set 1, when including valid inequalities

		Basic case	Extension 1 k=1, k=1.5		Extension 2	Extension 3
	MILP	$\min\sum t_i$	$\min\sum t_i$	$\min\sum t_i$	$\min\sum t_i + t_{max}$	$\min\sum t_i^2$
	add on		$t_i \leq T$	$t_i \leq \lceil 1.5T \rceil$	$t_i \leq t_{max}$	$t_i \leq \lceil 1.5T \rceil$
Aver. Time	(end)	171	11	35	214	668
	(box)	126	13	33	227	664
	(mm)	68	10	30	154	652
Min, Max	(end)	[3, 2372]	[1, 55]	[2, 311]	[3, TL]	[2, TL]
	(box)	[5, 1539]	[1, 82]	[2, 297]	[2, TL]	[2, TL]
	(mm)	[5, 589]	[1, 39]	[2, 206]	[5, 1658]	[1, TL]
Not solved	(end)	0	0	0	1	2
within TL	(box)	0	0	0	1	3
	(mm)	0	0	0	0	2

Table 16. Overview of solution times for Set 1, when including $c_i \geq DD_i - 3$, $\forall i$ and valid inequalities

		Basic case	Extension 1 k=1, k=1.5		Extension 2	Extension 3
	MILP	$\min\sum t_i$	$\min\sum t_i$	$\min\sum t_i$	$\min\sum t_i + t_{max}$	$\min\sum t_i^2$
	add on		$t_i \leq T$	$t_i \leq \lceil 1.5T \rceil$	$t_i \leq t_{max}$	$t_i \leq \lceil 1.5T \rceil$
Aver. Time	(end)	612	38	107	714	1268
	(box)	615	31	77	715	1554
	(mm)	259	46	72	313	1665
Min, Max	(end)	[7, TL]	[4, 176]	[4, 543]	[8, TL]	[5, TL]
	(box)	[8, TL]	[4, 135]	[4, 470]	[9, TL]	[3, TL]
	(mm)	[9, 898]	[5, 281]	[4, 300]	[7, 1472]	[4, TL]
Not solved	(end)	2	0	0	4	6
within TL	(box)	2	0	0	3	9
	(mm)	0	0	0	0	10

Table 17. Overview of solution times for Set 2, when including valid inequalities.

The results in Tables 15, 16 and 17 suggest to add the valid inequality (end) to Extension 3 and the valid inequality (mm) to both the Basic case and to Extension 2. We see from the three tables that adding the valid inequality (end) to Extension 3 have, in general, a positive effect on the average solution times for both Sets 1 and 2. For Set 2, the decrease of the average solution time is significant when the valid inequality (end) is added to Extension 3, from 1551 seconds to 1268 seconds. Moreover, three additional instances are solved to optimality within the time limit. For Set 1, adding the valid inequality (end) to Extension 3 (without and with $c_i \geq DD_i - 3$, $\forall i$) have no impact on the average solution time, but one additional instance is solved to optimality within the time limit for the case with bounds on how early a job can be scheduled (i.e., $c_i \geq DD_i - 3$, $\forall i$).

We see further that adding the valid inequality (mm) to the Base case and Extension 2 have a positive effect on the average solution times for both Sets 1 and 2. For Set 2, the reduction of the average solution time is significant and all instances are also solved to optimality within the time limit. The same apply for Set 1 for the case with bounds on how early a job can be scheduled (i.e., $c_i \geq DD_i - 3, \forall i$). Adding the valid inequality (mm) to the Basic case and Extension 2 results in significantly reduced average solution times, and one additional instance is also solved to optimality within the time limit for Extension 2. For Set 1 when there is no bound on how early a job can be scheduled, adding the inequality (mm) to the Base case and Extension 2 have no effect on the average solution times.

Except for the cases discussed above, the valid inequalities (end), (box) and (mm) have a minor or negative effect on the average solution times for the new time indexed formulations and for both data sets.

6. Suggested scheduling procedure

We have outlined a new time indexed formulation that can solve an industrial scheduling problem within a reasonable time frame. From the numerical studies, we see that the solution time for the new formulation is significantly lower than the solution time for an older formulation. When limited storage capacity is included (restrictions on how many days an order can be completed before its due date) the solution time for the new model decreases further. The huge improvement in solution time makes it possible to solve problem instances with up to around 30 days and around 23 individual jobs. Three different variants of the model are proposed to find a production plan with a good balance between the total tardiness and maximum tardiness in a reasonable time. Extension 1 balances the maximum tardiness and total tardiness by imposing an upper bound on the tardiness. A large upper bound focuses on minimizing the total tardiness, while a smaller upper bound focuses on keeping a small maximum tardiness. Extension 2 balances the maximum tardiness and the total tardiness by minimizing a weighted sum of both criteria. The lower the weight on the maximum tardiness, the higher the total tardiness and the larger the maximum tardiness. Extension 3 balances the maximum tardiness and total tardiness by minimizing the sum of the square of the tardiness. The quadratic objective function reduces the tardiness for all jobs with a relatively large tardiness at the expense of a larger total tardiness.

For Extension 1, we propose that $t_i \leq kT$ where T is equal to $\{\min\, t_{max},\ \text{s.t. } t_i \leq t_{max}\}$. To find the appropriate value of k, the foundry should consider the value of T, how important it is to reduce the total tardiness from Extension 1 ($k = 1$), and the maximum delay the customer can accept in the current market. For instances where we observe small values for T and for Extension 1 ($k = 1$), the foundry can iteratively increase k from 1 until a reasonable balance is achieved. An alternative is to solve Extension 1 for $k = 1$ and $k = 1.5$ and, based on the results, decide on a new k. To find the right weight k for the maximum tardiness in Extension 2 (min $\sum_{i=1}^{N} t_i + kT_{max}$) the foundry could take into account the value of T, the total tardiness from Extension 1 ($k = 1$), and what increase in maximum tardiness the foundry is ready to accept for a unit decrease in total tardiness. If it is appropriate that a one unit increase in the maximal tardiness is offset by at least a one unit decrease in the total tardiness, the weight of the maximum tardiness should at least be equal to 1. If a larger decrease in total tardiness is expected, the weight should be larger than 1. For Extension 3, we propose that the upper bound on the tardiness is based on the parameter T ($t_i \leq kT$). The upper bound on the tardiness in Extension 3 should be as tight as possible to reduce solution times, but large enough to avoid that the constraint becomes binding. If the main focus of the foundry is to minimize the total tardiness while keeping the tardiness below a well defined upper limit, we suggest that the foundry uses Extension 1 (min $\sum_{i=1}^{N} t_i$, s.t. $t_i \leq kT$) to find a suitable production schedule. If it is difficult to find a well defined upper limit and if it is important not only to minimize the total tardiness but also the tardiness for the most tardy job, we suggest that the foundry uses Extension 2 (min $\sum_{i=1}^{N} t_i + kt_{max}$). A higher relative weight on the maximum tardiness will specify the importance of keeping the maximum tardiness small. To reduce solution times, the tardiness should also here be constrained by an upper bound ($t_i \leq 2T$). If the main focus of the company is to reduce the tardiness for the most tardy jobs, we suggest that the company uses Extension 3 (min $\sum_{i=1}^{N} t_i^2$, s.t. $t_i \leq kT$), where k is chosen based on T in order to reduce solution times. This approach should be used when the company is more concerned about the duration of the tardiness than the total tardiness. That is when they would like to avoid very tardy jobs and prefer a larger total tardiness generated from more jobs but less tardy jobs.

From the numerical examples, we see that we generally find a good balance between the maximum tardiness and the total tardiness by letting $k = 1.5$ in Extension 1 and Extension 3, and by letting $k = 1$ in Extension 2. The schedules obtained with Extension 1 ($k = 1.5$) and Extension 2 ($k = 1$) will, compared to the schedules obtained with Extension 1 ($k = 1$) and Extension 3 ($k = 1.5$), have fewer tardy jobs but with a larger tardiness. Compared to the other alternatives, the schedules obtained with Extension 3 ($k = 1.5$) will have more jobs that are tardy, but fewer jobs with a large tardiness. To find the right balance between the maximum tardiness and the total tardiness, we recommend that the company compares the production schedule from Extension 1 ($k = 1$ and $k = 1.5$) and Extension 2 (s.t. $t_i \leq 2T$) and implement the schedule which best fits the market conditions or customer preferences. The schedule from Extension 3 ($k = 1.5$), should not be considered due to its large solution time, and because the proposed schedule is often similar to the schedule generated for one of the other extensions. An exception would be if there is an instance where the most important objective is to reduce the number of jobs with a large tardiness.

Based on the different characteristics of the schedules obtained with Extension 1 ($k = 1$ and $k = 1.5$) and Extension 2 ($k = 1$), we recommend the following solution procedure for the foundry.

1. Solve the basic model with the objective of finding the parameter T ($T = min\ t_{max}, st\ t_i \leq t_{max}$),
2. Solve Extension 1 ($k = 1$) to find the production schedule with the minimum total tardiness considering that $t_i \leq T$,
3. Solve Extension 1 ($k = 1.5$) and Extension 2 (s.t. $t_i \leq 2T$) to find alternative production schedules,
4. Compare the production schedules from Step 2 and Step 3 and implement the production schedule that best fits the current market conditions or customer needs.

The willingness of the customers to wait will in general change according to the market conditions, i.e. it will be higher when the capacity in the market is lower than the market demand.

We propose to use a rolling horizon process where the solution procedure is run once a week taking into account a planning horizon of four weeks. A detailed scheduling horizon of four weeks is suitable due to the uncertainty the company faces, such as emergency orders and the update of due dates.

For Set 1, with a high capacity usage rate and many late jobs, the new formulation can solve scheduling problems on a horizon of about one month. For Set 2, where the usage rate is lower than in Set 1, the new formulation can solve scheduling problems on a longer time horizon. The maximum time horizon in which the new formulation can be solved, within a reasonable computational time, depends on the number and types of jobs to be scheduled, the distribution of due dates and the capacity usage rate. In general, the solution time increases with the number of days, the number of jobs, the number of box types that are used, the usage rate and the number of tardy jobs.

7. Operational considerations

For the jobs scheduled to start in the first week, job specific materials and equipment have to be available at the right time for the jobs to start. The foundry will from time to time meet challenges as how to deal with orders that cannot be processed due to unavailable equipment and scheduling of emergency orders in a currently fully booked schedule. The first problem can be solved by adding a new constraint to the mathematical model that specifies the earliest start date for a job. The second problem can be solved by setting a high priority and an early due date for the order. The emergency order should be scheduled at the first available day given the set of orders currently being processed and the set of resources (i.e. combinations) currently available, since the foundry does not allow interruptions in the current processing of the orders. In the following subsections, we extend the mathematical formulation to take these issues into account. First, we allow for jobs to have different weights, then we make it possible to specify an earliest starting date of a job.

7.1. Take into account jobs with different weights

Both in the short term and more long term, the foundry would like to have the opportunity to specify that it is more important for some jobs to be scheduled according to their due dates than others. To specify that some jobs are more important than others, as it usually done in scheduling problems, job specific weights can be added to the tardiness in the objective function:

$$\text{minimize } \textstyle\sum_{i=1}^{N} w_i t_i \qquad (19)$$

The higher the value of w_i, the higher the priority of the order. If the company wants to insert an express job in the production plan, the priority of the job should be set very high.

7.2. Postpone the start date of a job

To postpone production of an order, the start date of the job can be ensured using the following constraint:

$$s_i \geq r_i,\ i = 1, \ldots, N \qquad (20)$$

where r_i specifies the point in time (usually called release dates in scheduling problems) where drawings and other equipment are available to process job i.

8. Conclusions

We proposed a novel time indexed formulation for the scheduling problem of a foundry producing propellers. The objective for the foundry is to find a production plan that minimizes both total tardiness and maximum tardiness, while taking into account capacity restrictions on labor, tools and space. Nonås and Olsen (2005) have previously proposed a mathematical model for this problem considering the objective of minimizing total tardiness. The formulation could, however, not solve realistic sized scheduling problems to optimality. We have extended the work of Nonås and Olsen (2005) in several ways. First, we have proposed a new and more efficient mathematical formulation that makes it possible to solve industrial sized instances within a reasonable time. Numerical results are presented that show the huge decrease in the solution time obtained with the new time indexed formulation. Further, we have extended the mathematical model to take into account not only the total tardiness but also the maximum tardiness. Different types of objectives and constraints are suggested based on how the company values a low total tardiness compared to a low maximum tardiness. We have also taken into account that there is limited storage capacity at the foundry and at their customers. Finally, we suggest extensions to handle emergency orders and forced deviations in the plan due for instance to missing equipment.

Further, we illustrate how the solution values change for different types of objectives and constraints. If the main focus of the foundry is to minimize the total tardiness while keeping the tardiness for the most tardy job below a given upper limit, we suggest to use a mathematical model that minimizes total tardiness given an upper bound on the maximum tardiness. If it is important not only to minimize the total tardiness but also the tardiness for the most tardy job, we suggest that the foundry finds a production plan that minimizes a weighted sum of total tardiness and maximum tardiness. If the main focus of the company is to reduce the tardiness for the most tardy jobs, we suggest that the company solves the given problem instance using a quadratic objective function. This solution should be used when the company is more concerned about the duration of the tardiness then the total tardiness. That is when they would like to avoid very tardy jobs and prefer a larger total tardiness generated from more but less tardy jobs. To keep the solution time low, the mathematical model should have upper bounds both on total tardiness and maximum tardiness.

Artigues, C., 2017. On the strength of time-indexed formulations for the resource-constrained project scheduling problem. Operations Research Letters 45 (2), 154–159.

Ballestín, F., Mallor, F., Mateo, P. M., 2012. Production scheduling in a market-driven foundry: a mathematical programming approach versus a project scheduling metaheuristic algorithm. Optimization and Engineering 13, 663–687.

Bigras, L.-P., Gamache, M., Savard, G., 2008. Time-indexed formulations and the total weighted tardiness problem. INFORMS Journal on Computing 20 (1), 133–142.

Boland, N., Clement, R., Waterer, H., 2016. A bucket indexed formulation for nonpreemptive single machine scheduling problems. INFORMS Journal on Computing 28 (1), 14–30.

Chen, Z.-L., Powell, W. B., 1999a. A column generation based decomposition algorithm for a parallel machine just-in-time scheduling problem. European Journal of Operational Research 116 (1), 220–232.

Chen, Z.-L., Powell, W. B., 1999b. Solving parallel machine scheduling problems by column generation. INFORMS Journal on Computing 11 (1), 78–94.

Dauzère-Pérès, S., 1995. Minimizing late jobs in the general one machine scheduling problem. European Journal of Operational Research 81 (1), 134–142.

Deghdak, K., T'kindt, V., Bouquard, J.-L., 2016. Scheduling evacuation operations. Journal of Scheduling 19 (4), 467–478.

Detienne, B., Dauzere-Péres, S., Yugma, C., 2011. Scheduling jobs on parallel machines to minimize a regular step total cost function. Journal of Scheduling 14 (6), 523–538.

Dyer, M. E., Wolsey, L. A., 1990. Formulating the single machine sequencing problem with release dates as a mixed integer program. Discrete Applied Mathematics 26 (2-3), 255–270.

Gauri, S. K., 2009. Modeling product-mix planning for batches of melt under multiple objectives in a small scale iron foundry. Production Engineering 3 (2), 189–196.

Hans, E., van de Velde, S., 2011. The lot sizing and scheduling of sand casting operations. International Journal of Production Research 49 (9), 2481–2499.

Ku, W.-Y., Beck, J. C., 2016. Mixed integer programming models for job shop scheduling: A computational analysis. Computers & Operations Research 73, 165–173.

Manne, A. S., 1960. On the job-shop scheduling problem. Operations Research 8 (2), 219–223.

Matibevic, G., Majdandzic, N., Lovrc, T., 2008. Production scheduling model in aluminium foundry. Strojniški vestnik - Journal of Mechanical Engineering 54 (1), 37–48.

Nonås, S. L., Olsen, K. A., 2005. Optimal and heuristic solutions for a scheduling problem arising in a foundry. Computers & Operations Research 32 (9), 2351–2382.

Obeid, A., Dauzère-Pérès, S., Yugma, C., 2014. Scheduling job families on non-identical parallel machines with time constraints. Annals of Operations Research 213 (1), 221–234.

Pan, Y., Liang, Z., 2017. Dual relaxations of the time-indexed ILP formulation for min–sum scheduling problems. Annals of Operations Research 249 (1-2), 197–213.

Queyranne, M., Schulz, A. S., 1994. Polyhedral approaches to machine scheduling. TU Berlin, Preprint 408.

Samà, M., D'Ariano, A., D'Ariano, P., Pacciarelli, D., 2017. Scheduling models for optimal aircraft traffic control at busy airports: Tardiness, priorities, equity and violation considerations. Omega 67, 81–98.

Sousa, J. P., Wolsey, L. A., 1992. A time indexed formulation of non-preemptive single machine scheduling problems. Mathematical programming 54 (1), 353–367.

Teixeira, R. F., Fernandes, F. C. F., Pereira, N. A., 2010. Binary integer programming formulations for scheduling in market-driven foundries. Computers & Industrial Engineering 59 (3), 425–435.

van den Akker, J., Hurkens, C. A., Savelsbergh, M. W., 2000. Time-indexed formulations for machine scheduling problems: Column generation. INFORMS Journal on Computing 12 (2), 111–124.

van den Akker, J. M., Hoogeveen, J. A., van de Velde, S. L., 1999. Parallel machine scheduling by column generation. Operations Research 47 (6), 862–872.

Velez, S., Dong, Y., Maravelias, C. T., 2017. Changeover formulations for discrete-time mixed-integer programming scheduling models. European Journal of Operational Research 260 (3), 949–963.

Appendix A

Tables A1, A2 and A3 specify the individual solution times for Set 1 (without and with $c_i \geq DD_i - 3$) and Set 2. The instance number and the problem size are given in the first three columns. An instance not solved to optimality within the time limit of 2,500 seconds is indicated by "TL". If CPLEX returns a non-optimal solution for an instance within the time limit of 2,500 seconds, this is also indicated by "TL". The notation $\lceil 1.5T \rceil$ means that $1.5T$ is rounded up to the closest integer. Both the new and old models are solved using AMPL (IBM ILOG CPLEX 12.9).

			Basic case		Extension 1 k=1, k=1.5				Extension 2		Extension 3	
			min$\sum t_i$		min$\sum t_i$		min$\sum t_i$		min$\sum t_i + t_{max}$		min$\sum t_i^2$	
	Size				$t_i \leq T$		$t_i \leq \lceil 1.5T \rceil$		$t_i \leq t_{max}$		$t_i \leq \lceil 1.5T \rceil$	
	N	D	New	Old	New	Old	New	Old	New	Old	New	Old
1	19	34	121	TL	28	231	30	604	246	TL	TL	TL
2	17	34	53	1535	26	15	11	44	120	1530	TL	TL
3	18	33	179	TL	23	27	48	528	213	TL	1008	TL
4	19	32	83	TL	149	168	70	333	350	TL	TL	TL
5	16	33	240	TL	26	110	95	2323	168	1203	1970	TL
6	23	33	907	TL	77	107	146	463	283	TL	TL	TL
7	18	33	185	TL	50	210	58	808	529	TL	1966	TL
8	20	34	116	822	27	12	51	121	97	TL	TL	TL
9	19	34	125	2127	5	2	50	29	144	TL	TL	TL
10	23	33	32	27	2	1	2	0	69	93	2	0
11	20	32	856	TL	47	431	130	618	790	TL	TL	TL
12	20	32	100	621	77	79	85	73	338	TL	TL	TL
13	20	30	136	TL	44	14	51	104	436	2001	TL	TL
14	21	34	426	2495	23	91	324	TL	449	TL	TL	TL
15	20	34	285	1639	28	11	43	122	708	TL	TL	TL
16	23	34	TL	TL	138	TL	TL	TL	TL	TL	TL	TL
17	20	34	213	TL	40	30	63	186	420	TL	TL	TL
18	21	34	469	TL	2495	632	546	TL	511	TL	TL	TL
19	23	32	544	TL	23	6	53	82	422	TL	566	TL
20	18	32	342	TL	5	5	31	58	232	TL	87	896
21	18	33	241	TL	114	106	111	360	318	TL	TL	TL
22	23	30	593	TL	81	446	186	72	1053	TL	TL	TL
	Average		398	2143	160	238	213	659	474	2286	2073	2314

Table A1. Individual solution times for Set 1, for the new and old models.

			Basic case	Extension 1 k=1, k=1.5		Extension 2	Extension 3
	Size		$\min\sum t_i$	$\min\sum t_i$	$\min\sum t_i$	$\min\sum t_i + t_{max}$	$\min\sum t_i^2$
				$t_i \leq T$	$t_i \leq \lceil 1.5T \rceil$	$t_i \leq t_{max}$	$t_i \leq \lceil 1.5T \rceil$
	N	D	New	New	New	New	New
1	19	34	47	17	27	133	620
2	17	34	7	1	3	7	9
3	18	33	57	4	27	243	146
4	19	32	38	34	28	100	558
5	16	33	77	7	21	87	65
6	23	33	65	16	22	179	815
7	18	33	28	13	24	70	402
8	20	34	28	2	6	27	43
9	19	34	35	3	13	48	204
10	23	33	4	2	2	2	2
11	20	32	160	22	49	251	579
12	20	32	27	4	7	41	258
13	20	30	69	3	12	115	416
14	21	34	43	6	42	210	1866
15	20	34	50	3	18	108	341
16	23	34	1529	83	297	TL	TL
17	20	34	66	5	19	86	443
18	21	34	56	27	39	155	TL
19	23	32	63	3	14	114	27
20	18	32	40	3	6	52	74
21	18	33	15	6	7	44	224
22	23	30	221	18	28	423	TL
	Average		124	13	32	227	663

Table A2. Individual solution times for Set 1, for the new model, $c_i \geq DD_i - 3$.

			Basic case		Extension 1 k=1, k=1.5				Extension 2		Extension 3	
	Size		$\min\sum t_i$		$\min\sum t_i$		$\min\sum t_i$		$\min\sum t_i + t_{max}$		$\min\sum t_i^2$	
					$t_i \leq T$		$t_i \leq \lceil 1.5T \rceil$		$t_i \leq t_{max}$		$t_i \leq \lceil 1.5T \rceil$	
	N	D	New	Old	New	Old	New	Old	New	Old	New	Old
1	20	50	22	287	7	7	14	9	50	263	71	221
2	15	45	12	12	4	0	3	1	12	5	4	1
3	26	60	TL	TL	366	391	140	1189	TL	TL	TL	TL
4	25	53	21	71	11	6	9	11	27	2345	TL	1834
5	21	70	155	284	9	3	10	6	153	393	886	373
6	27	47	336	2468	24	26	39	105	263	TL	TL	TL
7	25	70	679	1679	22	36	45	228	785	TL	TL	2413
8	24	62	1649	TL	87	484	410	TL	2162	TL	TL	TL
9	14	55	53	64	3	4	8	26	69	79	71	504
10	23	65	TL	TL	57	842	290	TL	TL	TL	TL	TL
11	22	55	57	150	7	2	9	6	67	106	1118	243
12	25	62	652	TL	14	9	32	149	981	TL	TL	TL
13	16	43	9	4	4	1	4	0	9	5	17	2
14	30	70	1121	TL	67	727	189	99	1248	TL	TL	TL
15	19	65	86	80	6	5	8	11	69	696	125	TL
16	20	70	273	1852	13	1	19	14	206	2144	TL	563
	Average		3666	1218	44	159	77	429	694	1473	1551	1478

Table A3. Individual solution times for Set 2, for the new and old models.

Tables A4, A5 and A6 present the total tardiness ($\sum t_i$) and the maximum tardiness (t_{max}) for the different extensions of the new time indexed formulation. "TL" indicates that optimality is not proved within the time limit of 2,500 seconds. t_i and t_{max} are marked with *end*, respectively *mm*, if an optimal solution is found within the time limit by including the valid inequality (*end*), respectively (*mm*). The notation $\lceil 1.5T \rceil$ means that $1.5T$ is rounded up to the closest integer.

			Basic case		Extension 1 k=1, k=1.5				Extension 2		Extension 3	
	Size		Min $\sum t_i$		Min $\sum t_i$ $t_i \leq T$		Min $\sum t_i$ $t_i \leq \lceil 1.5T \rceil$		Min $\sum t_i + t_{max}$ s.t. $t_i \leq t_{max}$		Min $\sum T_i^2$ $t_i \leq \lceil 1.5T \rceil$	
	N	D	$\sum t_i$	t_{max}	$\sum t_i$	t_{max}	$\sum t_i$	t_{max}	$\sum t_i$	t_{max}	$\sum t_i$	t_{max}
1	19	34	34	24	46	8	37	11	37	11	TL	TL
2	17	34	4	4	6	2	4	3	4	3	TL	TL
3	18	33	42	15	48	8	43	11	43	11	51	8
4	19	32	14	9	24	5	16	8	14	9	TL	TL
5	16	33	29	15	37	8	31	12	31	9	31	9
6	23	33	10	4	12	3	10	4	10	4	TL	TL
7	18	33	21	11	29	7	21	11	21	11	30	9
8	20	34	12	8	22	6	12	8	12	8	12^{end}	8^{end}
9	19	34	16	12	23	6	21	9	16	12	26^{end}	6^{end}
10	23	33	0	0	0	0	0	0	0	0	0	0
11	20	32	18	5	24	3	18	5	19	4	TL	TL
12	20	32	4	4	6	2	4	3	4	3	TL	TL
13	20	30	23	12	28	5	24	6	24	6	TL	TL
14	21	34	20	20	60	7	40	11	20	20	TL	TL
15	20	34	26	6	26	5	26	6	26	5	TL	TL
16	23	34	TL	TL	59	7	TL	TL	TL	TL	TL	TL
17	20	34	18	6	19	4	18	6	19	4	TL	TL
18	21	34	11	5	15	3	11	5	11	4	TL	TL
19	23	32	13	7	14	3	14	4	14	3	14	4
20	18	32	25	10	27	4	27	4	27	4	27	4
21	18	33	12	6	13	4	12	6	13	4	TL	TL
22	23	30	10	6	14	2	10	3	10	3	TL	TL

Table A4. Individual solution values for Set 1.

	Size		Basic case		Extension 1 k=1, k=1.5				Extension 2		Extension 3	
			$\min\sum t_i$		$\min\sum t_i$ $t_i \leq T$		$\min\sum t_i$ $t_i \leq \lceil 1.5T \rceil$		$\min\sum t_i + t_{max}$ $t_i \leq t_{max}$		$\min\sum t_i^2$ $t_i \leq \lceil 1.5T \rceil$	
	N	D	$\sum t_i$	t_{max}	$\sum t_i$	t_{max}	$\sum t_i$	t_{max}	$\sum t_i$	t_{max}	$\sum t_i$	t_{max}
1	19	34	35	24	48	8	39	12	42	9	49	9
2	17	34	7	4	7	4	7	4	7	4	7	4
3	18	33	44	15	53	8	47	12	47	12	54	8
4	19	32	16	10	25	6	18	9	16	10	23	7
5	16	33	32	13	40	8	33	12	35	9	36	9
6	23	33	14	5	17	3	14	5	15	4	17	4
7	18	33	22	11	30	7	22	11	22	11	29	10
8	20	34	13	8	22	6	13	8	13	8	13	8
9	19	34	20	11	24	6	23	9	24	6	28	6
10	23	33	0	0	0	0	0	0	0	0	0	0
11	20	32	20	5	21	4	20	5	21	4	21	4
12	20	32	12	6	21	5	12	6	12	6	12	6
13	20	30	25	7	28	5	25	7	25	7	28	6
14	21	34	29	21	64	7	44	11	29	21	49	10
15	20	34	29	6	30	5	29	6	29	6	31	5
16	23	34	49	17	69	7	55	10	50^{mm}	14^{mm}	TL	TL
17	20	34	23	9	24	5	24	8	24	5	25	5
18	21	34	26	10	27	7	26	9	26	8	TL	TL
19	23	32	14	4	15	3	14	4	15	3	14	4
20	18	32	26	10	29	5	28	8	29	5	29	5
21	18	33	20	9	22	6	20	9	21	7	22	6
22	23	30	13	3	13	3	13	3	13	3	14^{end}	3^{end}

Table A5. Individual solution values for Set 1, with limits on storage capacity, $c_i \geq DD_i - 3$.

	Size		Basic case		Extension 1 k=1, k=1.5				Extension 2		Extension 3	
			$\min\sum t_i$		$\min\sum t_i$ $t_i \leq T$		$\min\sum t_i$ $t_i \leq \lceil 1.5T \rceil$		$\min\sum t_i + t_{max}$ $t_i \leq t_{max}$		$\min\sum t_i^2$ $t_i \leq \lceil 1.5T \rceil$	
	N	D	$\sum t_i$	t_{max}	$\sum t_i$	t_{max}	$\sum t_i$	t_{max}	$\sum t_i$	t_{max}	$\sum t_i$	t_{max}
1	20	50	2	2	2	2	2	2	2	2	2	2
2	15	45	0	0	0	0	0	0	0	0	0	0
3	26	60	33^{mm}	19^{mm}	46	7	37	10	37^{mm}	10^{mm}	TL	TL
4	25	53	13	7	16	6	13	7	13	7	14	7
5	21	70	4	2	4	2	4	2	4	2	4	2
6	27	47	5	4	5	4	5	4	5	4	TL	TL
7	25	70	28	14	32	10	28	14	32	10	29	13
8	24	62	67	23	69	15	67	23	68	16	TL	TL
9	14	55	25	11	29	8	25	9	25	9	25	9
10	23	65	40^{mm}	13^{mm}	59	10	40	12	40^{mm}	12^{mm}	TL	TL
11	22	55	7	5	7	5	7	5	7	5	7	5
12	25	62	40	13	49	10	40	13	40	13	43^{end}	13^{end}
13	16	43	10	9	10	9	10	9	10	9	10	9
14	30	70	16	16	35	12	16	16	16	16	TL	TL
15	19	65	14	10	16	7	14	10	14	8	16	7
16	20	70	7	6	7	3	7	3	7	3	7^{end}	3^{end}

Table A6. Individual solution values for Set 2.

Table A7 presents the individual solutions for Set 1 for the new mathematical model with Extension 2 ($k = 1$ and $k = 2$) and an upper bound on the tardiness. The first columns show the instance number and problem size, and the next two columns the individual solutions when Extension 2 ($k = 1$) is extended with upper bounds on the tardiness ($t_i \leq 1.5T$ and $t_i \leq 2T$). The two last columns present the individual solutions for Extension 2 ($k = 2$) with and without upper bounds on the tardiness. The columns "*new*″ indicate the computational times for the new mathematical model. "TL" indicates that optimality is not proved within the time limit of 2,500 seconds.

	Extension 2													
	Size		$\min\sum t_i + t_{max}$ $t_i \leq t_{max}$ $t_i \leq \lceil 1.5T \rceil$			$\min\sum t_i + t_{max}$ $t_i \leq t_{max}$ $t_i \leq 2T$			$\min\sum t_i + 2t_{max}$ $t_i \leq t_{max}$			$\min\sum t_i + 2t_{max}$ $t_i \leq t_{max}$ $t_i \leq \lceil 1.5T \rceil$		
	N	D	$\sum t_i$	t_{max}	new	$\sum t_i$	t_{max}	new	$\sum t_i$	t_{max}	new	$\sum t_i$	t_{max}	new
1	19	34	37	11	40	37	11	160	41	9	810	37	11	118
2	17	34	4	3	27	4	3	50	4	3	86	6	2	29
3	18	33	43	11	58	43	11	106	48	8	324	48	8	69
4	19	32	16	8	200	14	9	132	20	6	622	20	6	191
5	16	33	31	9	104	31	9	424	31	9	241	31	9	125
6	23	33	10	4	301	10	4	455	10	4	561	10	4	288
7	18	33	21	11	72	21	11	370	27	8	359	29	7	117
8	20	34	12	8	47	12	8	90	12	8	175	12	8	68
9	19	34	22	7	51	16	12	124	23	6	82	23	6	41
10	23	33	0	0	2	0	0	2	0	0	48	0	0	2
11	20	32	18	5	341	18	5	500	19	4	1875	19	4	425
12	20	32	4	3	102	4	3	112	6	2	226	6	2	97
13	20	30	24	6	66	24	6	90	24	6	697	24	6	48
14	21	34	40	11	182	34	14	665	20	20	1809	41	10	487
15	20	34	26	5	61	26	5	123	26	5	325	26	5	59
16	23	34	TL	TL	TL	TL	TL	TL	TL	TL	TL	TL	TL	TL
17	20	34	19	4	136	19	4	166	19	4	642	19	4	104
18	21	34	11	4	497	11	4	851	11	4	640	11	4	612
19	23	32	14	3	60	14	3	95	14	3	267	14	3	40
20	18	32	27	4	41	27	4	66	27	4	231	27	4	48
21	18	33	13	4	175	13	4	382	13	4	309	13	4	144
22	23	30	10	3	102	10	3	210	10	3	557	10	3	156
	Average				236			350			609			262

Table A7. Individual solutions for Set 1, Extension 2: Upper bounds on t_i and different weights on t_{max}.

Table A8 presents the total tardiness and the maximum tardiness obtained when the maximum tardiness is minimized for each instance in Set 1, without and with limited storage capacity ($c_i \leq DD_i - 3$). The column new and old represent the solution times. Again, "TL" indicates that optimality is not proved within the time limit of 2,500 seconds.

	Size		min t_{max} $t_i \leq t_{max}$				min t_{max} $t_i \leq t_{max}$ $c_i \leq DD_i - 3$		
	N	D	$\sum t_i$	T	new	old	$\sum t_i$	T	new
1	19	34	152	8	37	14	152	8	21
2	17	34	34	2	46	57	68	4	15
3	18	33	144	8	53	27	144	8	15
4	19	32	95	5	63	106	114	6	40
5	16	33	128	8	54	18	128	8	307
6	23	33	63	3	135	463	63	3	27
7	18	33	126	7	22	37	126	7	15
8	20	34	120	6	48	78	120	6	27
9	19	34	114	6	16	15	114	6	6
10	23	33	0	0	25	3	0	0	5
11	20	32	60	3	575	794	80	4	39
12	20	32	40	2	33	221	100	5	18
13	20	30	100	5	80	45	100	5	29
14	21	34	147	7	51	91	147	7	28
15	20	34	100	5	41	25	100	5	21
16	23	34	161	7	196	579	161	7	136
17	20	34	80	4	87	34	100	5	42
18	21	34	63	3	TL	TL	147	7	58
19	23	32	69	3	36	11	69	3	35
20	18	32	72	4	12	11	90	5	12
21	18	33	72	4	36	54	108	6	38
22	23	30	46	2	64	11	69	3	75
	Average				191	236			46

Table A8. Individual solutions for Set 1, when solving for T.

Table A9 shows the solutions and the solution times first for Set 2, Extension 2 (k=1) with upper bounds on t_i, and then for Extension 2 (k=2) with and without upper bounds on t_i, $t_i \leq \lceil 1.5T \rceil$, $i = 1, \ldots, N$, using the new formulation. For Extension 2 (k=2), with and without upper bounds on t_i, Column "mm" refers to the solution time for the new time indexed formulation when the valid inequality (mm) is added.

Extension 2

	Size		$\min\sum t_i + t_{max}$ $t_i \leq t_{max}$ $t_i \leq \lceil 1.5T \rceil$			$\min\sum t_i + 2t_{max}$ $t_i \leq t_{max}$				$\min\sum t_i + 2t_{max}$ $t_i \leq t_{max}$ $t_i \leq \lceil 1.5T \rceil$			
	N	D	$\sum t_i$	t_{max}	new	$\sum t_i$	t_{max}	new	mm	$\sum t_i$	t_{max}	new	mm
1	20	50	2	2	8	2	2	47	15	2	2	7	15
2	15	45	0	0	4	0	0	13	9	0	0	4	5
3	26	60	37	10	393	37	10	TL	2372	37	10	992	1438
4	25	53	13	7	13	13	7	28	55	13	7	14	16
5	21	70	4	2	12	4	2	278	47	4	2	13	16
6	27	47	5	4	21	5	4	265	166	5	4	38	38
7	25	70	32	10	97	32	10	437	249	32	10	141	113
8	24	62	68	16	1510	69	15	TL	879	69	15	TL	734
9	14	55	25	9	17	25	9	50	32	25	9	14	14
10	23	65	40	12	2432	40	12	TL	1184	40	12	1016	369
11	22	55	7	5	14	7	5	54	15	7	5	12	11
12	25	62	40	13	158	42	12	1428	273	42	12	106	102
13	16	43	10	9	6	10	9	7	7	10	9	5	5
14	30	70	16	16	205	16	16	1161	496	16	16	138	82
15	19	65	14	8	9	14	8	56	105	14	8	13	15
16	20	70	7	3	24	7	3	141	71	7	3	29	14
	Average				308			718	373			316	187

Table A9. Individual solutions for Set 2, Extension 2 (k=1, k=2) with and without upper bounds on t_i, and without and with the valid inequality (mm).

Table A10 shows the value of T and the solution times in seconds used to find T for each instance in Set 2 using both the new and the old formulations.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
T	2	0	7	6	2	4	10	15	8	10	5	10	9	12	7	3
New sec	24	13	182	30	119	219	137	470	18	183	21	83	7	185	27	107
Old sec	26	19	313	30	143	203	79	TL	2	243	53	442	3	610	24	151

Table A10. Individual solutions for Set 2.

Table A11 reports, for each instance in Set 1, how much the total tardiness and the maximum tardiness increase when the basic mathematical model is extended to include limited storage capacity ($c_i \geq DD_i - 3$). The results are reported for the Basic case and the four extensions.

			Basic case		Extension 1 k=1, k=1.5				Extension 2		Extension 3	
			min$\sum t_i$		min$\sum t_i$ $t_i \leq T$		min$\sum t_i$ $t_i \leq \lceil 1.5T \rceil$		min$\sum t_i + t_{max}$ $t_i \leq t_{max}$		min$\sum t_i^2$ $t_i \leq \lceil 1.5T \rceil$	
	Size N	D	$\sum t_i$	t_{max}	$\sum t_i$	t_{max}	$\sum t_i$	t_{max}	$\sum t_i$	t_{max}	$\sum t_i$	t_{max}
1	19	34	1	0	2	0	2	1	5	-2	TL	TL
2	17	34	3	0	1	2	3	1	3	1	TL	TL
3	18	33	2	0	5	0	4	1	4	1	3	0
4	19	32	2	1	1	1	2	1	2	1	TL	TL
5	16	33	3	-2	3	0	2	0	4	0	5	0
6	23	33	4	1	5	0	4	1	5	0	TL	TL
7	18	33	1	0	1	0	1	0	1	0	-1	1
8	20	34	1	0	0	0	1	0	1	0	1	0
9	19	34	4	-1	1	0	2	0	8	-6	2	0
10	23	33	0	0	0	0	0	0	0	0	0	0
11	20	32	2	0	-3	1	2	0	2	0	TL	TL
12	20	32	8	2	15	3	8	3	8	3	TL	TL
13	20	30	2	-5	0	0	1	1	1	1	TL	TL
14	21	34	9	1	4	0	4	0	9	1	TL	TL
15	20	34	3	0	4	0	3	0	3	1	TL	TL
16	23	34	TL	TL	10	0	TL	TL	TL	TL	TL	TL
17	20	34	5	3	5	1	6	2	5	1	TL	TL
18	21	34	15	5	12	4	15	4	15	4	TL	TL
19	23	32	1	-3	1	0	0	0	1	0	0	0
20	18	32	1	0	2	1	1	4	2	1	2	1
21	18	33	8	3	9	2	8	3	8	3	TL	TL
22	23	30	3	-3	-1	1	3	0	3	0	TL	TL

Table A11: Changes in the solution values, Set1. The mathematical model extended with $c_i \geq DD_i - 3, \forall i$.

Appendix B

Tables B1 and B2 provide an overview of the production capacity and production requirements (in days) of the instances in Sets 1 and 2. The first column shows the problem size. Then, the second column gives the number of empty days in the production schedule (using Extension 1, $k = 1.5$), followed by the estimates (in days) of the production capacity requirements for each box type. The last columns show the average estimated capacity requirement and the smallest and largest estimated capacity requirements. The estimated number of production days required for box type b is calculated based on the box type combination $abc = (2, 0.9, 1, 0.7, 0.4)$, which is the average of all the box type combinations in Table 2. For a given problem instance, the estimated capacity for box type b is equal to the total demand for box type b divided by $abc[b]$.

	Size		Ext. 1 ($k = 1.5$)	Estimate of necessary production capacity (in days)							
			Days with	Box	Box	Box	Box	Box	Summary		
	N	D	no production	type 1	type 2	type 3	type 4	type 5	avg	min	max
1	19	34	3	31	34	30	0	45	28	0	45
2	17	34	2	19	43	35	20	45	32	19	45
3	18	34	3	17	20	48	21	50	31	17	50
4	19	33	3	23	33	17	29	35	27	17	35
5	16	32	2	20	23	24	29	45	28	20	45
6	23	33	3	16	37	39	29	33	30	16	39
7	18	33	2	19	42	14	13	48	27	13	48
8	20	34	1	31	22	34	29	40	31	22	40
9	19	34	4	9	28	28	30	55	30	9	55
10	23	33	3	15	40	29	20	38	28	15	40
11	20	32	3	12	33	13	37	43	28	12	43
12	20	32	4	11	40	42	47	0	28	0	47
13	20	30	4	16	23	27	39	23	25	16	39
14	21	34	2	31	48	14	0	35	25	0	48
15	20	34	2	26	29	0	64	30	30	0	64
16	23	34	1	17	43	16	33	43	30	16	43
17	20	34	5	27	20	33	26	43	30	20	43
18	21	34	4	14	38	26	39	33	30	14	39
19	23	32	3	22	54	35	0	20	26	0	54
20	18	32	3	17	20	9	57	28	26	9	57
21	18	33	3	15	23	38	40	33	30	15	40
22	23	30	4	25	20	42	24	23	27	20	42

Table B1. Set 1: Days with no production (Extension 1, $k = 1.5$), and estimates of necessary capacity for each box type.

	Size		Ext. 1 ($k = 1.5$)	Estimate of necessary production capacity (in days)							
			Days with	Box	Box	Box	Box	Box	Summary		
	N	D	no production	type 1	type 2	type 3	type 4	type 5	avg	min	max
1	20	50	1	4	24	33	64	70	39	4	70
2	15	45	6	0	69	48	27	23	33	0	69
3	26	60	10	25	13	50	96	0	37	0	96
4	25	53	0	39	14	16	40	60	34	14	60
5	21	70	12	24	0	105	11	63	41	0	105
6	27	47	0	18	76	55	27	43	44	18	76
7	25	70	16	40	36	0	96	78	50	0	96
8	24	62	5	17	20	86	31	120	55	17	120
9	14	55	9	20	6	5	24	110	33	5	110
10	23	65	5	26	52	14	30	108	46	14	108
11	22	55	5	45	54	14	0	48	32	0	54
12	25	62	6	26	53	0	96	30	41	0	96
13	16	43	6	27	0	43	9	28	21	0	43
14	30	70	20	27	106	72	0	38	48	0	106
15	19	65	9	15	18	9	67	65	35	9	67
16	20	70	12	3	32	57	89	33	43	3	89

Table B2. Set 2: Days with no production (Extension 1, $k = 1.5$), and estimates of necessary capacity for each box type.

Tables B3 and B4 present, for each problem instance in both sets, the minimum and maximum order sizes (number of propeller blades) for each box type.

	Size		Type 1	Type 2	Type 3	Type 4	Type 5
1	19	34	[5, 14]	[5, 12]	[5,13]	[0, 0]	[5, 8]
2	17	34	[3, 17]	[5, 21]	[5,17]	[7, 7]	[4, 14]
3	18	34	[4, 25]	[5, 7]	[6,12]	[6, 9]	[3, 7]
4	19	33	[3, 13]	[3, 16]	[5,12]	[5, 5]	[5, 9]
5	16	32	[5, 24]	[4, 12]	[5,14]	[6, 14]	[3, 5]
6	23	33	[5, 13]	[4, 8]	[5,12]	[5, 8]	[3, 5]
7	18	33	[5, 12]	[5, 14]	[3,6]	[4, 5]	[5, 8]
8	20	34	[5, 18]	[5, 9]	[5,12]	[8, 12]	[3, 8]
9	19	34	[3, 1]	[3, 8]	[5,7]	[5, 16]	[4, 8]
10	23	33	[5, 8]	[3, 7]	[3,6]	[5, 9]	[5, 5]
11	20	32	[4, 5]	[3, 8]	[5,8]	[5, 6]	[5, 7]
12	20	32	[5, 9]	[3, 10]	[5,15]	[5, 11]	[0, 0]
13	20	30	[5, 8]	[3, 8]	[5,8]	[3, 8]	[4, 5]
14	21	34	[3, 16]	[3, 12]	[3,6]	[0, 0]]	[4, 10]
15	20	34	[4, 15]	[4, 10]	[0,0]	[4, 11]	[4, 8]
16	23	34	[3, 11]	[3, 11]	[2,10]	[3, 9]	[3, 5]
17	20	34	[3, 17]	[6, 12]	[5,12]	[2, 12]	[6, 11]
18	21	34	[5, 14]	[3, 12]	[4,8]	[3, 7]	[3, 5]
19	23	32	[3, 9]	[4, 12]	[3,1]	[0, 0]]	[8, 8]
20	18	32	[5, 11]	[4, 5]	[4,5]	[4, 9]	[5, 6]
21	18	33	[5, 12]	[4, 12]	[5,14]	[4, 8]	[6, 7]
22	23	30	[5, 19]	[3, 5]	[3,6]	[7, 10]	[4, 5]

Table B3. For each instance in Set 1, minimum and maximum order sizes for each box type.

	Size		Type 1	Type 2	Type 3	Type 4	Type 5
	N	D	[min,max]	[min,max]	[min,max]	[min,max]	[min,max]
1	20	50	[7, 7]	[3, 19]	[4, 17]	[2, 11]	[2, 11]
2	15	45	[0, 0]	[5, 17]	[8, 21]	[8, 11]	[4, 5]
3	26	60	[3, 14]	[5, 7]	[4, 21]	[2, 13]	[0, 0]
4	25	53	[3, 19]	[5, 8]	[2, 5]	[3, 8]	[7, 17]
5	21	70	[2, 18]	[0, 0]	[5, 25]	[8, 8]	[3, 17]
6	27	47	[5, 13]	[4, 20]	[4, 19]	[6, 13]	[3, 9]
7	25	70	[2, 22]	[5, 21]	[0, 0]	[5, 15]	[5, 13]
8	24	62	[4, 25]	[5, 7]	[5, 23]	[5, 11]	[5, 17]
9	14	55	[5, 24]	[5, 5]	[5, 5]	[3, 14]	[3, 14]
10	23	65	[4, 22]	[5, 16]	[3, 6]	[4, 12]	[4, 17]
11	22	55	[3, 24]	[3, 18]	[3, 6]	[0, 0]	[4, 15]
12	25	62	[4, 15]	[4, 22]	[0, 0]	[4, 11]	[12, 12]
13	16	43	[3, 17]	[0, 0]	[5, 22]	[2, 4]	[11, 11]
14	30	70	[3, 9]	[4, 21]	[3, 20]	[0, 0]	[15, 15]
15	19	65	[5, 11]	[2, 5]	[4, 5]	[4, 12]	[5, 15]
16	20	70	[5, 5]	[4, 20]	[5, 19]	[3, 16]	[6, 7]

Table B4. For each instance in Set 2, minimum and maximum order sizes for each box type.

Tables B5 and B7 present the parameters and the completion dates for respectively instance 3 in Set 1 and instance 12 in Set 2 (using Extension 1, $k = 1.5$). Table B5 also presents the completion dates for (Extension 1, $k = 1.5$, $c_i <= DD_i - 3$). Tables B6 and B8 provide the schedule for instance 3 in Set 1 and for instance 12 in Set 2 (using Extension 1, $k = 1.5$). For each day d we present which box combination to use, which jobs to produced and the unused box capacity for day d.

Set 1, Instance 3: Parameters

	Box type 1			Box type 2			Box type 3						Box type 4		Box type 5			
Job i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
O_i	4	5	25	7	5	6	6	8	12	8	6	8	9	6	5	5	7	3
DD_i	2	8	27	5	12	25	5	5	9	12	18	28	6	14	11	15	20	25
Extension 1, $k = 1.5$: Completion dates, tardiness and days too early																		
c_i	1	6	27	5	11	17	9	5	19	13	24	28	7	21	12	17	31	24
t_i	0	0	0	0	0	0	4	0	10	1	6	0	1	7	1	2	11	0
c_i-DD_i	1	2	0	0	1	8	0	0	0	0	0	0	0	0	0	0	0	1
Extension 1, $k = 1.5$, $c_i \leq DD_i - 3$: Completion dates, tardiness and days too early																		
c_i	1	6	25	5	12	28	9	5	19	13	24	28	7	21	12	17	32	24
t_i	0	0	0	0	0	3	4	0	10	1	6	0	1	7	1	2	12	0
c_i-DD_i	1	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Table B5. Instance 3, Set 1: Parameters and output data for each job i (Extension 1, k=1.5 and Extension 1, k=1.5, $c_i \leq DD_i$-3).

Day	Box combination					Scheduled jobs					Spare capacity,				
d	b=1	b=2	b=3	b=4	b=5	b=1	b=2	b=3	b=4	b=5	b=1	b=2	b=3	b=4	b=5
1	5	0	0	2	0	1			13		1	0	0	0	0
2	0	2	2	1	0		4	8	13		0	1	0	0	0
3	0	2	2	1	0		4	8	13		0	0	0	0	0
4	0	2	2	1	0		4	8	13		0	0	0	0	0
5	0	2	2	1	0		4	8	13		0	0	0	0	0
6	5	0	0	2	0	2			13		0	0	0	0	0
7	0	2	2	1	0		5	7	13		0	1	0	0	0
8	0	1	2	0	1		5	7		15	0	0	0	0	0
9	0	1	2	0	1		5	7		15	0	0	0	0	0
10	0	1	2	0	1		5	10		15	0	0	0	0	0
11	0	1	2	0	1		5	10		15	0	0	0	0	0
12	0	1	2	0	1		6	10		15	0	0	0	0	0
13	0	1	2	0	1		6	10		16	0	0	0	0	0
14	0	1	2	0	1		6	9		16	0	0	0	0	0
15	0	1	2	0	1		6	9		16	0	0	0	0	0
16	0	1	2	0	1		6	9		16	0	0	0	0	0
17	0	1	2	0	1		6	9		16	0	0	0	0	0
18	2	0	2	1	0	3		9	14		1	0	0	0	0
19	2	0	2	1	0	3		9	14		0	0	0	0	0
20	5	0	0	2	0	3			14		0	0	0	0	0
21	5	0	0	2	0	3			14		0	0	0	0	0
22	2	0	2	0	1	3		11		18	0	0	0	0	0
23	2	0	2	0	1	3		11		18	0	0	0	0	0
24	2	0	2	0	1	3		11		18	0	0	0	0	0
25	2	0	2	0	1	3		12		17	0	0	0	0	0
26	2	0	2	0	1	3		12		17	0	0	0	0	0
27	2	0	2	0	1	3		12		17	0	0	0	0	0
28	0	1	2	0	1			12		17	0	1	0	0	0
29	2	0	2	0	1					17	2	0	2	0	0
30	2	0	2	0	1					17	2	0	2	0	0
31	0	1	2	0	1					17	0	1	2	0	0
32	0	1	2	0	1						0	1	2	0	1
33	0	1	2	0	1						0	1	2	0	1
34	7	0	0	0	0						7	0	0	0	0

Table B6. Set 1, Instance 3 (Extension 1, k=1.5): For each day, we present which box combination to use, which jobs to produce, and the unused capacity of each box type.

Set 2, Instance 12: Parameters

	Box type 1							Box type 2						Box type 4								
Job i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
O_i	9	5	5	5	15	4	8	4	22	8	4	5	5	7	5	11	7	4	6	7	6	5
DD_i	8	23	30	41	41	51	59	4	8	31	36	46	60	3	5	12	19	35	46	47	50	55
Extension 1, $k = 1.5$: Completion dates and tardiness																						
c_i	2	20	30	34	41	24	54	6	20	29	35	33	60	5	10	25	17	29	38	47	41	43
t_i	0	0	0	0	0	0	0	2	12	0	0	0	0	2	5	13	0	0	0	0	0	0

Table B7. Set 2, Instance 12: Parameters and output data for each job i (Extension 1, k=1.5)

Day	Box combination used,					Scheduled jobs					Spare capacity,				
d	b=1	b=2	b=3	b=4	b=5	b=1	b=2	b=3	b=4	b=5	b=1	b=2	b=3	b=4	b=5
1	5	0	0	2	0	1		14			0	0	0	0	0
2	4	0	0	2	0	1		14			1	0	0	0	0
3	0	1	0	1	1		8	14		25	0	0	0	0	0
4	0	1	0	1	1		8	14		25	0	0	0	0	0
5	0	1	0	1	1		8	14		25	0	0	0	0	0
6	0	1	0	1	1		8	15		25	0	0	0	0	0
7	0	1	0	1	1		9	15		25	0	0	0	0	0
8	0	1	0	1	1		9	15		25	0	0	0	0	0
9	0	1	0	1	1		9	15		25	0	0	0	0	0
10	0	1	0	1	1		9	15		25	0	0	0	0	0
11	0	1	0	1	1		9	17		25	0	0	0	0	0
12	0	1	0	1	1		9	17		25	0	0	0	0	0
13	0	1	0	1	1		9	17		25	0	0	0	0	0
14	0	1	0	1	1		9	17		25	0	0	0	0	0
15	0	2	0	1	0		9	17			2	0	0	0	0
16	0	2	0	1	0		9	17			2	0	0	0	0
17	1	2	0	1	0	2	9	17			1	0	0	0	0
18	1	3	0	0	0	2	9				0	0	2	0	0
19	1	3	0	0	0	2	9				0	0	2	0	0
20	2	2	0	1	0	2	9	16			0	0	0	0	0
21	0	0	0	2	0			16			5	0	0	0	0
22	1	0	0	2	0	6		16			4	0	0	0	0
23	1	0	0	2	0	6		16			4	0	0	0	0
24	2	0	0	2	0	6		16			3	0	0	0	0
25	0	0	0	2	0			16			5	0	0	0	0
26	0	2	0	1	0		10	18			2	0	0	0	0
27	0	2	0	1	0		10	18			2	0	0	0	0
28	0	2	0	1	0		10	18			2	0	0	0	0
29	0	2	0	1	0		10	18			2	0	0	0	0
30	5	0	0	0	0	3					0	0	0	2	0
31	0	0	0	0	0						5	0	0	2	0
32	2	2	0	0	0	4	12				0	0	0	1	0
33	1	3	0	0	0	4	12				0	0	2	0	0
34	2	2	0	0	0	4	11				0	0	0	1	0
35	0	2	0	1	0		11	19			2	0	0	0	0
36	0	0	0	2	0			19			5	0	0	0	0
37	0	0	0	2	0			19			5	0	0	0	0
38	0	0	0	2	0			19/21			5	0	0	0	0
39	5	0	0	2	0	5		21			0	0	0	0	0
40	5	0	0	2	0	5		21			0	0	0	0	0
41	5	0	0	2	0	5		21/22			0	0	0	0	0
42	0	0	0	2	0			22			5	0	0	0	0
43	0	0	0	2	0			22			5	0	0	0	0
44	0	0	0	2	0			20			5	0	0	0	0
45	0	0	0	1	0			20			5	0	0	1	0
46	0	0	0	2	0			20			5	0	0	0	0
47	0	0	0	2	0			20			5	0	0	0	0
48	0	0	0	2	0			23			5	0	0	0	0
49	0	0	0	2	0			23			5	0	0	0	0
50	0	0	0	1	0			23			5	0	0	1	0
51	0	0	0	0	0						5	0	0	2	0
52	0	0	0	0	0						5	0	0	2	0
53	3	0	0	0	0	7					2	0	0	2	0
54	5	0	0	0	0	7					0	0	0	2	0
55	0	0	0	0	0						1	3	2	0	0
56	0	0	0	2	0			24			5	0	0	0	0
57	0	0	0	1	0			24			5	0	0	1	0
58	0	0	0	1	0			24			5	0	0	1	0
59	0	3	0	0	0		13				1	0	2	0	0
60	0	2	0	0	0		13				1	1	2	0	0
61	0	0	0	0	0						5	0	0	2	0
62	0	0	0	0	0						5	0	0	2	0

Table B8 Set 2, Instance 12 (Extension 1 k=1.5): For each day, we present which box combination to use, which jobs to produce, and the unused capacity of each box type.

Appendix C

Tables C1, C2 and C3 specify the individual solution times for Set 1 (without and with $c_i \geq DD_i - 3$) and Set 2 (without $c_i \geq DD_i - 3$), for the new time indexed formulation when the different valid inequalities are added to the Basic case and the extensions to the Basic case.

Tables C1, C2 and C3 give for, each problem instance, the average solution time for the Basic case and for each of the extensions to the Basic case, when the valid inequalities (end), (box) and (mm) are added. An instance not solved to optimality within the time limit of 2,500 seconds is indicated by "TL". When Extension 3 returns a non optimal solution within the time limit of 2,500 seconds, this is also indicated by "TL". The notation $\lceil 1.5T \rceil$ means that $1.5T$ is rounded up to the closest integer.

	Base case			Extension 1 k=1, k=1.5						Extension 2			Extension 3		
	$\min \sum t_i$			$\min \sum t_i$			$\min \sum t_i$			$\min \sum t_i + t_{max}$			$\min \sum t_i^2$		
				$t_i \leq T$			$t_i \leq \lceil 1.5T \rceil$			$t_i \leq t_{max}$			$t_i \leq \lceil 1.5T \rceil$		
Nr	end	box	mm	end	box	mm	end	box	mm	end	box	mm	end	box	mm
1	113	124	72	28	43	25	27	34	22	275	246	162	TL	TL	TL
2	65	96	51	12	26	15	44	44	28	124	113	65	TL	TL	TL
3	126	178	188	23	26	23	36	44	48	185	214	243	1274	1012	1097
4	103	88	111	184	149	93	70	86	114	468	352	293	TL	TL	TL
5	251	212	306	32	24	49	105	72	74	88	237	165	TL	1971	TL
6	1027	606	668	95	75	89	85	211	99	1230	718	553	TL	TL	TL
7	158	184	100	26	54	50	58	68	65	241	532	164	TL	1963	TL
8	111	118	130	22	20	19	46	64	28	118	97	125	1213	TL	TL
9	96	125	163	7	6	9	44	50	45	210	144	109	1251	TL	TL
10	54	33	14	2	2	2	3	1	2	65	69	61	2	2	2
11	612	859	496	146	46	84	201	132	129	2502	780	1434	TL	TL	TL
12	109	100	155	42	77	42	79	86	121	307	338	388	TL	TL	TL
13	149	181	187	32	43	15	60	57	92	173	282	328	TL	TL	TL
14	136	426	221	28	38	44	95	115	132	226	451	458	TL	TL	TL
15	175	287	406	17	30	27	57	62	45	218	706	141	TL	TL	TL
16	2518	2510	2514	829	136	265	TL	TL	TL	TL	TL	TL	TL	TL	TL
17	485	283	448	52	39	43	57	63	93	673	521	891	TL	TL	TL
18	1344	468	1268	2503	2360	823	589	663	602	1775	505	801	TL	TL	TL
19	694	534	398	11	24	9	56	54	67	386	473	295	389	564	606
20	132	339	173	9	4	4	36	31	39	140	233	111	170	88	102
21	237	240	317	73	169	109	83	172	117	338	320	455	TL	TL	TL
22	452	292	691	73	82	65	92	186	158	575	748	993	TL	TL	TL
Average	416	377	413	193	158	87	202	218	210	584	482	488	2014	2073	2128

Table C1. Individual solution times for Set 1, when the valid inequalities are added to the Basic case and its extensions

	Base case			Extension 1 k=1, k=1.5						Extension 2			Extension 3		
	min$\sum t_i$			min$\sum t_i$			min$\sum t_i$			min$\sum t_i + t_{max}$			min$\sum t_i^2$		
				$t_i \leq T$			$t_i \leq \lceil 1.5T \rceil$			$t_i \leq t_{max}$			$t_i \leq \lceil 1.5T \rceil$		
Nr	end	box	mm	end	box	mm	end	box	mm	end	box	mm	end	box	mm
1	40	49	23	17	16	12	23	27	24	149	134	49	568	617	559
2	4	8	5	2	2	1	3	3	2	9	7	7	8	7	6
3	41	58	58	3	3	4	32	29	27	176	243	152	312	145	248
4	51	41	36	26	34	23	31	28	20	117	101	88	743	558	531
5	79	83	51	4	7	6	26	21	35	88	87	57	45	64	251
6	112	69	69	22	15	26	33	21	19	207	179	130	1073	816	598
7	36	29	34	18	13	14	24	25	12	94	70	80	807	401	412
8	12	29	21	4	3	3	4	6	4	14	27	20	40	43	47
9	27	36	12	3	3	3	9	13	7	53	48	32	161	204	181
10	3	5	5	1	1	2	2	2	2	3	2	5	2	2	1
11	179	165	113	24	23	13	53	48	91	275	250	352	1291	579	622
12	23	28	15	6	3	4	10	13	8	46	42	24	242	257	165
13	44	72	51	3	3	3	19	12	12	135	115	35	709	414	656
14	97	44	46	6	7	5	43	42	40	201	209	125	1988	1875	2118
15	136	52	38	3	3	4	17	18	20	72	108	68	347	350	152
16	2372	1539	589	55	82	39	311	297	206	TL	TL	1658	TL	TL	TL
17	107	65	46	4	5	5	22	19	14	111	87	64	637	441	1245
18	52	57	53	26	27	24	38	39	50	93	155	114	TL	TL	TL
19	79	62	91	3	3	2	16	13	17	81	114	67	33	26	31
20	51	40	37	3	3	4	17	7	14	57	52	53	111	74	85
21	26	16	13	5	6	6	7	6	11	34	44	19	160	224	141
22	182	219	83	13	19	16	39	28	31	182	423	179	863	TL	1282
Average	171	126	68	11	13	10	35	33	30	214	227	154	668	664	652

Table C2. Individual solution times for Set 1, when the valid inequalities are added to the Basic case and its extensions, $c_i \geq DD_i - 3$, $\forall i$.

	Base case			Extension 1 k=1, k=1.5						Extension 2			Extension 3		
	min$\sum t_i$			min$\sum t_i$			min$\sum t_i$			min$\sum t_i + t_{max}$			min$\sum t_i^2$		
				$t_i \leq T$			$t_i \leq \lceil 1.5T \rceil$			$t_i \leq t_{max}$			$t_i \leq \lceil 1.5T \rceil$		
Nr	end	box	mm	end	box	mm	end	box	mm	end	box	mm	end	box	mm
1	34	21	15	5	7	7	10	10	11	67	57	15	61	71	565
2	20	11	9	4	4	5	5	4	4	34	12	9	5	3	4
3	982	TL	754	176	135	281	342	100	178	TL	TL	1472	TL	TL	TL
4	48	22	32	9	11	13	10	10	11	38	27	45	1147	TL	513
5	157	151	138	10	14	18	11	13	35	295	152	44	806	903	TL
6	260	323	44	31	15	26	39	30	28	326	260	97	TL	TL	TL
7	590	646	144	46	20	35	108	46	41	909	779	246	TL	TL	TL
8	TL	1468	894	74	82	94	444	315	300	TL	TL	1139	TL	TL	TL
9	55	51	23	5	4	5	7	7	6	75	69	47	175	71	191
10	TL	TL	898	116	106	117	543	470	231	TL	TL	1091	TL	TL	TL
11	140	56	27	15	9	12	12	18	11	151	65	53	506	1145	TL
12	1192	654	803	17	17	16	34	32	55	TL	978	277	1588	TL	TL
13	7	8	9	4	4	5	4	5	4	8	9	7	32	19	46
14	1006	1083	214	103	52	83	115	153	216	1230	1239	350	TL	TL	TL
15	68	81	60	8	9	10	10	8	6	63	69	74	195	127	286
16	219	261	84	12	14	10	15	14	14	213	204	41	745	TL	TL
Average	612	615	259	38	31	46	107	77	72	714	715	313	1268	1554	1665

Table C3. Individual solution times for Set 2, when the valid inequalities are added to the Basic case and its extensions.