Computers & Operations Research 65 (2016) 53–63

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

# Bi-criteria sequencing of courses and formation of classes for a bottleneck classroom

Oliver G. Czibula [a,b,*], Hanyu Gu [a], Feng-Jang Hwang [a], Mikhail Y. Kovalyov [c], Yakov Zinder [a]

[a] *University of Technology Sydney, Australia*
[b] *Ausgrid, Australia*
[c] *United Institute of Informatics Problems, National Academy of Sciences of Belarus, Belarus*

ARTICLE INFO

Available online 26 June 2015

*Keywords:*
Batch scheduling
Bi-criterion scheduling
Job families
Equal processing times

ABSTRACT

In this paper, the problem of class formation and sequencing for multiple courses subject to a bottleneck classroom with an ordered bi-criteria objective is studied. The problem can be modelled as a single-machine batch scheduling problem with incompatible job families and parallel job processing in batches, where the batch size is family-dependent. For the minimisation of the number of tardy jobs, the strong NP-hardness is proven. For the performance measure of the maximum cost, we consider single criterion and bi-criteria cases. We present an $O(n^2 \log n)$ algorithm, $n$ is the number of jobs, for both cases. An Integer Programming model as well as Simulated Annealing and Genetic Algorithm matheuristics to solve a fairly general case of the bi-criteria problem is presented and computationally tested.


## 1. Introduction

This study addresses a scheduling problem that is inspired by the problem of scheduling training courses for trainees working for large organisations with scarce teaching resources. One of such organisations is Ausgrid, Australia's largest electricity distributor, which is required by Australian law under the Work Health and Safety Act 2011 – Part 2, division 4, section 27, to deliver regular safety and technical training to all its employees who work on or near the electricity network. The problem of scheduling thousands of workers (students), hundreds of classes of multiple courses with dedicated trainers and classrooms across a small number of training facilities is highly complex in practice. It is useful, however, to develop efficient algorithms for solving smaller or special cases as it provides insight into the underlying problem.

This paper investigates the case of a single classroom with a limited capacity. In the studied problem, there are a set of students and a set of courses. Each student is required to undertake one or more courses by certain due dates. It is assumed that we have a single classroom where multiple students can be taught simultaneously, provided they are all undertaking the same course. The classroom retains a limited capacity, and each course is characterised by a maximum class size in students, in accordance with its distinct training content. Without loss of generality, the maximum class size is no greater than the classroom capacity. The class duration is also course-dependent. The classroom is not allowed to be used for more than one class at any time. The objective is to determine an optimal formation and sequence of classes for courses and students so as to minimise the considered performance metrics. As with many industrial problems, it is often difficult to quantify the needs of the organisation with a single objective function. We investigate the case where two ordered objectives, a primary and a secondary objective, are considered. An optimal schedule is one that minimises the secondary objective over the set of schedules that minimise the primary objective.

The studied course scheduling problem with a bottleneck classroom can be modelled as a single-machine batch scheduling problem with incompatible job families, where the batch size is family-dependent. Before providing the comprehensive mapping between these two problems, we describe the single machine batch scheduling problem with incompatible job families. There is a set of $n$ jobs, which are categorised into $m$ distinct job families. The number of jobs in family $f$ is denoted by $n_f$, and $\sum_{f=1}^{m} n_f = n$. Here and below, we assume that each summation and maximum is taken over all families or jobs, unless otherwise specified. The job $j$ of family $f$ is denoted by $(f, j)$, $f \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n_f\}$. The jobs belonging to the same job family $f$ have a common processing time $p_f$. Each job $(f, j)$ is also characterised by a due date $d_{f,j}$, a weight $w_{f,j}$ and two non-decreasing cost functions $g_{f,j}(C_{f,j})$ and $h_{f,j}(C_{f,j})$, where $C_{f,j}$ is the completion time of the job. The batching machine, where the parallel-batch mode is applied, can process simultaneously multiple jobs belonging to the same family

* Corresponding author. Tel.: +61 2 8745 1559.
*E-mail addresses:* oczibula@ausgrid.com.au (O.G. Czibula), hanyu.gu@uts.edu.au (H. Gu), feng-jang.hwang@uts.edu.au (F.-J. Hwang), kovalyov_my@newman.bas-net.by (M.Y. Kovalyov), yakov.zinder@uts.edu.au (Y. Zinder).

http://dx.doi.org/10.1016/j.cor.2015.06.010
0305-0548/

as a batch [2]. The maximum number of jobs allowed to be processed concurrently, i.e. the batch size, which is clearly no greater than the machine capacity, depends on the family of those jobs inside the batch and is denoted by $b_f$. The processing time of a batch belonging to family $f$ does not depend on the number of jobs in the batch and is always $p_f$. There is no machine idle time between the batches. A schedule is defined to be the sequence and composition of all batches. The objective is to find a batch schedule for jobs such that the considered performance measures dependent on job completion times are minimised.

The mapping is given as follows. The bottleneck classroom is represented as a single batching machine. Each course maps to a job family, and each course requirement for a student is depicted as a job. This means that if a student is required to complete, for example, three different courses, these three requirements will be represented by three jobs which belong to three distinct families. The due date imposed upon some course requirement for some student is exactly the due date of the corresponding job. The course-dependent class duration is represented as the family-dependent job processing time. Then a class of students formed for some course corresponds to a batch of jobs for some family. The maximum class size for some course is represented as the batch size for the corresponding job family. Hereafter, the studied problem will be described as the single-machine batching scheduling problem with incompatible job families subject to family-dependent batch sizes.

There has been an effort in studying the prototype of the considered problem, i.e. the single-machine batching scheduling problem with incompatible job families, where the batch size is constant rather than family-dependent. For this same prototype problem but with a different application, Uzsoy [24] showed that the minimisation of the total weighted completion time, or maximum lateness can be solved in polynomial time. Later, Mehta et al. [20] proved that the problem for the minimisation of total tardiness is NP-hard in the strong sense. As for the minimisation of the number of tardy jobs, Jolai [14] showed that the problem is NP-hard and presented a polynomial-time dynamic programming algorithm for the special case where the number of job families and machine capacity are fixed. Jolai [15] studied the problem of minimising earliness–tardiness criteria subject to a common unrestricted job due date, proving the problem is NP-hard. Considering a more generalised problem where each job requires a different amount of machine capacity, Dobson and Nambimadom [6] considered the problem of minimising the total weighted completion time, presenting a polynomial-time algorithm for special cases of the problem, and some heuristics for the general case. Azizoglu and Webster [1] designed a branch-and-bound procedure for the same problem which can optimally solve cases up to about 25 jobs in acceptable time. The problem of scheduling jobs with equal processing time on a single batching machine to minimise a primary and secondary criterion, as well as a linear weighted criterion, is studied by Lie et al. [19]. The authors present polynomial time algorithms for various combinations of the primary and secondary criteria. Janiak et al. [12], Cheng et al. [4] and Janiak et al. [13] developed polynomial time algorithms for *group technology* single machine scheduling problems, including problems with ordered criteria, in which the jobs are processed sequentially in a batch, a single batch is made for each family and each batch is preceded by a set-up time.

The remainder of this paper is organised as follows. Section 2 provides analytical results of the considered problem, including a proof of NP-hardness in the strong sense for the number of tardy jobs criterion, a polynomial time algorithm for the maximal primary and maximal secondary criterion case, and a polynomial algorithm for the case with a given sequence of batches. Section 3 discusses computational approaches to solving the considered problem, including Integer Programming (IP) approaches for exact solutions, and Simulated Annealing (SA) and Genetic Algorithm (GA) approaches for heuristic solutions. Section 5 contains concluding remarks.

## 2. Analytical results

In this section we discuss the complexity results for four ordered bi-criteria cases and the polynomial solvability of one of the cases under the assumption that the sequence of batches is fixed a priori. We use the notation $(f_1, f_2)$ to represent the ordered relationship between two objectives in the ordered bi-criteria problem. For example, $(\sum g_{f,j}, \sum h_{f,j})$ is an ordered bi-criteria objective whose optimal solution minimises the secondary criterion $\sum h_{f,j}$ over the set of solutions that minimise the primary criterion $\sum g_{f,j}$. Note that the objective functions $\sum h_{f,j}$, $\sum g_{f,j}$, $\max h_{f,j}$ and $\max g_{f,j}$ are *regular*, that is, they are non-decreasing in each argument.

### 2.1. Complexity of ordered bi-criteria cases

There are four cases of the bi-criteria problem with the sum and maximum objectives depending on their type and order.

We will first strengthen the result of Jolai [14] and show that the problem of minimising the number of tardy jobs, $\sum U_{f,j}$,is NP-hard in the strong sense. Given a batch schedule,function $U_{f,j}(t)$ is defined such that $U_{f,j}(t) = 0$ if $t \le d_{f,j}$ (for $t = C_{f,j}$ job $(f,j)$ is on-time) and $U_{f,j}(t) = 1$ if $t > d_{f,j}$ (for $t = C_{f,j}$ job $(f, j)$ is tardy).

**Theorem 1.** *The problem of minimising $\sum U_{f,j}$ is NP-hard in the strong sense irrespective of whether the batch sizes are bounded or unbounded.*

**Proof.** We use a reduction from the decision version of the NP-hard in the strong sense problem of minimising total weighted tardiness on a single machine discussed in Lawler [18]. Given an instance $\mathtt{A}$ of the single-machine total weighted tardiness minimisation problem,we can construct an instance $\mathtt{B}$ of the studied problem in pseudo-polynomial time in the following way. For instance $\mathtt{A}$, we let $\overline{n}$ be the number of jobs,and denote by $\overline{p}_j$, $\overline{d}_j$,and $\overline{w}_j$ the processing time,due date,and weight,respectively,for job $j = 1, \ldots, \overline{n}$. For each job $j$ in instance $\mathtt{A}$, construct a corresponding family $j$ consisting of $\overline{w}_j$ jobs with due date $\overline{d}_j$, $\overline{w}_j$ jobs with due date $\overline{d}_j + 1$, and so on until $\overline{w}_j$ jobs with due date $\overline{d}_j + UB$, where the upper bound $UB = \sum_{j=1}^{\overline{n}} \overline{p}_j$, for instance $\mathtt{B}$. This gives instance $\mathtt{B}$ a total of $\overline{w}_j(UB+1)$ jobs for each family $j$, whose job processing time is $\overline{p}_j$, $j = 1, \ldots, \overline{n}$. The batch size for each family is assumed to be sufficiently large such that all jobs of each family can be processed simultaneously in one batch.

Consider the decision problem of determining whether a schedule with the total weighted tardiness no greater than some positive integer $\kappa$ exists for instance $\mathtt{A}$. If such a schedule exists for instance $\mathtt{A}$, then a schedule whose number of tardy jobs is no greater than $\kappa$ for instance $\mathtt{B}$ also exists. In the corresponding schedule of instance $\mathtt{B}$, all jobs of the same family are processed in one batch, and the batch sequence, i.e. the family sequence, of instance $\mathtt{B}$ is identical to the job sequence of instance $\mathtt{A}$. See Fig. 1 for an illustration.

Conversely, if a batch schedule with the number of tardy jobs no greater than $\kappa$ exists for instance $\mathtt{B}$, then it can be converted into a schedule of the same or better quality in which all jobs of the same family are assigned in the same batch. Then the sequence of jobs for instance $\mathtt{A}$ defined by the sequence of families in the latter batch schedule has the total weighted tardiness no greater than $\kappa$. □