## Discrete Optimization

# Scheduling under linear constraints 

Kameng Nip ${ }^{\text {a }}$, Zhenbo Wang ${ }^{\text {a,*, }}$, Zizhuo Wang ${ }^{\text {b }}$<br>a Department of Mathematical Sciences, Tsinghua University, Beijing, China<br>${ }^{\mathrm{b}}$ Department of Industrial and Systems Engineering, University of Minnesota, MN, USA

## A R T I C L E I N F O

## Article history:

Received 17 June 2015
Accepted 17 February 2016
Available online 26 February 2016

## Keywords:

Parallel machine scheduling
Linear programming
Computational complexity
Approximation algorithm


#### Abstract

We introduce a parallel machine scheduling problem in which the processing times of jobs are not given in advance but are determined by a system of linear constraints. The objective is to minimize the makespan, i.e., the maximum job completion time among all feasible choices. This novel problem is motivated by various real-world application scenarios. We discuss the computational complexity and algorithms for various settings of this problem. In particular, we show that if there is only one machine with an arbitrary number of linear constraints, or there is an arbitrary number of machines with no more than two linear constraints, or both the number of machines and the number of linear constraints are fixed constants, then the problem is polynomial-time solvable via solving a series of linear programming problems. If both the number of machines and the number of constraints are inputs of the problem instance, then the problem is NP-Hard. We further propose several approximation algorithms for the latter case.


© 2016 Elsevier B.V. All rights reserved.

## 1. Introduction

A scheduling problem aims to allocate resources to jobs, so as to meet a specific objective, e.g., to minimize the makespan or the total completion time. One common assumption in the classical scheduling problem is that the processing times of jobs are deterministic and are given in advance. However, in practice, the processing times are usually uncertain/unknown or could be part of the decisions. A number of works in the literature have proposed various scheduling models in which the processing times are uncertain/unknown, such as the stochastic scheduling problem (Dean, 2005; Möhring, Radermacher, \& Weiss, 1984; Möhring, Schulz, \& Uetz, 1999) and the robust scheduling problem (Daniels \& Kouvelis, 1995; Kasperski, 2005; Kasperski \& Zielinski, 2008). In the stochastic scheduling problem, it is assumed that the processing times are random variables and the expected makespan is considered. In the robust scheduling problem, it is assumed that the processing time of each job belongs to a certain set and the objective is to find a robust schedule under some performance criterion (e.g., minimize the maximum absolute deviation of total completion time, or the total lateness). Note that in either the stochastic or the robust scheduling problems, the processing times are still exogenously given.

[^0]In the presented paper, we introduce a new scheduling model. In our model, the processing times of jobs are not exogenously given, instead they can be chosen as part of the decisions, but they must satisfy a set of linear constraints. We call this problem the "scheduling under linear constraints" (SLC) problem. Note that the SLC problem reduces to the classical parallel machine scheduling problem $P \| C_{\max }$ when the processing times of jobs are given (or equivalently, when the linear constraints have a unique solution). This problem is related to the scheduling problem with controllable processing times studied in the literature (Nowicki \& Zdrzalka, 1988, 1990; Shabtay \& Steiner, 2007). In the latter problem, the processing times of jobs are controlled by factors such as the starting times and the sequence of the jobs, while in our problem, the processing times are part of the decision variables.

The SLC problem is also related to the lot sizing and scheduling problem in production planning, which decides the type and amount of jobs to process at each time period over a time horizon (Drexl \& Haase, 1995; Drexl \& Kimms, 1997; Haase, 1994). However, although these two problems may share some similar backgrounds, they are different in many ways: (1) In the SLC problem, each task must be completed in a consecutive time interval and can only be chosen once, while in the lot sizing and scheduling problem, an activity (e.g., the production of certain type of products) can be scheduled in multiple non-consecutive periods; (2) The objective of the lot sizing and scheduling problem is to minimize the total costs, including the setup costs, the inventory holding costs, etc, which is significantly different from the

Table 1
Example for the industrial production problem.

| Composition | Alloy |  |  |  |  | Demand |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
|  | 1 | 2 | 3 | $\ldots$ | $n$ |  |
| Iron | 24 | 8 | 3 | $\ldots$ | 2 | $\geq 56$ |
| Copper | 3 | 3 | 3 | $\ldots$ | 1 | $\geq 30$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| Aluminium | 4 | 33 | 137 | $\ldots$ | 100 | $\geq 1000$ |
| Max. of alloy |  |  |  |  |  | Quantity |
| 1 | 1 | 0 | 0 | $\ldots$ | 0 | $\leq 10$ |
| 2 | 0 | 1 | 0 | $\cdots$ | 0 | $\leq 7$ |
| 3 | 0 | 0 | 1 | $\cdots$ | 0 | $\leq 20$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| $n$ | 0 | 0 | 0 | $\cdots$ | 1 | $\leq 15$ |

objective of the SLC problem (Kreipl \& Pinedo, 2004), which is to minimize the makespan of the schedule; (3) Due to the difference in the objective, the key tradeoff in these two problems are different. In the SLC problem, the main consideration is how to balance the workload of each machine, and assign the jobs evenly across machines. In contrast, the key consideration in the lot sizing and scheduling problem is how to divide the jobs and schedule them (e.g., how many units of products to produce in each use of the machine), which is more similar to that in an EOQ model (see Snyder \& Shen, 2011); (4) As we will see later, the mathematical programming formulation for the SLC problem is a mixed integer quadratic program, while the common formulation for the lot sizing and scheduling problem is a mixed integer program (e.g., see Drexl and Haase, 1995, page 75). Therefore, the methodologies and research approaches are also different.

In the following, we provide a few examples that motivate the study of the SLC problem.

1. Industrial production problem. Perhaps the earliest motivation for the scheduling problem came from manufacturing (e.g., see Pinedo, 2009, 2012). Suppose a manufacturer requires certain amounts of different raw metals, and he needs to extract them from several alloys. There are several machines that can extract the alloys in parallel. We focus on the procedure of extracting the alloys, of which the goal is to finish as early as possible. In this problem, the processing times of extracting each alloy depend on the processing quantities, and traditionally, they are predetermined in advance. However, in practice, those quantities are determined by the demands of the raw metals and can be solved as a feasible solution to a blending problem (Danø, 1960; Eiselt \& Sandblom, 2007). Sometimes, each alloy also has its own maximum quantity. An example of such a scenario is given in Table 1.
In the example shown in Table 1, the demand of iron is 56 , and each unit of alloy 1 contains 24 units of iron, each unit of alloy 2 contains 8 units of iron, etc. Let $x_{i}$ be the quantity of alloy $i$ to be extracted. Then the requirement on the demand of iron can be represented as a linear inequality $24 x_{1}+8 x_{2}+$ $3 x_{3}+\cdots+2 x_{n} \geq 56$. Furthermore, the maximum amount of alloy 1 available is 10 , which can be represented as a linear inequality $x_{1} \leq 10$. Similarly, we can write linear constraints for the demand of other metals and the quantity for other alloys. In this problem, the decision maker needs to determine the nonnegative job quantities $x_{1}, \ldots, x_{n}$ satisfying the above linear constraints, and then assign these jobs to the parallel machines such that the last completion time is minimized. This problem can be viewed as a minimum makespan parallel machine scheduling problem, where the processing times of jobs satisfy some linear constraints.

Table 2
Example for the advertising media selection Problem.

| Sum of | Each unit time broadcast provides |  |  |  |  |  |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
|  | ad 1 | ad 2 | ad 3 | $\ldots$ | ad $n$ |  |
| Attractions to women | 20 | 100 | 100 | $\ldots$ | 10 | $\geq 500$ |
| Attractions to men | 15 | 10 | 0 | $\ldots$ | 80 | $\geq 500$ |
| Attractions to teens | 30 | 0 | 30 | $\ldots$ | 100 | $\geq 200$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| Max time for ad 1 | 1 | 0 | 0 | $\ldots$ | 0 | $\leq 20$ |
| Min time for ad 1 | 1 | 0 | 0 | $\ldots$ | 0 | $\geq 10$ |
| Max time for ad 2 | 0 | 1 | 0 | $\ldots$ | 0 | $\leq 35$ |
| $:$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |

2. Advertising media selection problem. A company has several parallel broadcast platforms which can broadcast advertisements simultaneously, such as multiple screens in a shopping mall or different spots on a website. There is a customer who wants to broadcast his advertisements (ad $1, \ldots, n$ ) on these platforms. ${ }^{1}$ It is required that each individual advertisement must be broadcast without interruption and the running time of each advertisement has to satisfy some linear constraints. The company needs to decide the running times $x_{i}$ allocated to each advertisement $i$, and also which advertisement should be released on which platform as well as the releasing order. The objective is to minimize the completion time. An example of such a problem is given in Table 2.
Similar to the first example, the above-described problem can be naturally formulated as a minimum makespan parallel machine scheduling problem in which the parameters (running times of the advertisements) are determined by a system of linear constraints.
3. Transportation problem. Both linear programming and machine scheduling problems have extensive applications in the field of transportation management (Eiselt \& Sandblom, 2007; Pinedo, 2009, 2012). The parallel machine scheduling problem has many similarities with the transportation scheduling models. For example, a fleet of tankers or a number of workers can be considered as a parallel machine environment, and transporting or handling cargo is analogous to processing a job (Pinedo, 2009). Meanwhile, the transportation problem can be formulated as a linear program. Let $x_{i j}$ be the capacity of cargo that needs to be transported from origin $i$ to destination $j$. They often have to satisfy certain supply and demand constraints, which are usually linear constraints.
In practice, the decision maker decides how to assign cargo (jobs) to tankers or workers (parallel processors), so as to finish the handling as quickly as possible. This is a parallel machine scheduling problem. And the processing times usually depend on $x_{i j} \mathrm{~s}$, which have to satisfy some linear constraints as mentioned above. This also leads to a parallel machine scheduling problem with linear constraints.

In this paper, we study the SLC problem, discussing the computational complexity and algorithms for this problem under various settings. In particular, we show that if there is only one machine with an arbitrary number of linear constraints, or there is an arbitrary number of machines with no more than two linear constraints, or both the number of machines and the number of linear constraints are fixed constants, then the problem is polynomialtime solvable via solving a series of linear programming problems. If both the number of machines and the number of constraints are inputs of the problem instance, then the problem is

[^1]
# https://daneshyari.com/en/article/479244 

Download Persian Version:

## https://daneshyari.com/article/479244

## Daneshyari.com


[^0]:    * Corresponding author. Tel.: +861062772796; fax: +861062787945.

    E-mail addresses: njm13@mails.tsinghua.edu.cn (K. Nip), zwang@math.tsinghua. edu.cn (Z. Wang), zwang@umn.edu (Z. Wang).

[^1]:    ${ }^{1}$ This example can be easily extended to cases with multiple customers.

