Computers & Industrial Engineering 115 (2018) 220–225

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Minimizing the maximum lateness on a single machine with raw material constraints by branch-and-cut

Péter Györgyi, Tamás Kis*

Institute for Computer Science and Control, H1111 Budapest, Kende str. 13-17, Hungary

ARTICLE INFO

Keywords:
Scheduling
Single machine
Non-renewable resources
Branch-and-cut
Integer programming

ABSTRACT

Machine scheduling with raw material constraints has a great practical potential, as it is solved by ad hoc methods in practice in several manufacturing and logistic environments. In this paper we propose an exact method for solving this problem with the maximum lateness objective based on mathematical programming, our main contribution being a set of new cutting planes that can be used to accelerate a MIP solver. We report on computational results on a wide set of instances.

1. Introduction

Counting with raw materials (or non-renewable resources more generally) in the course of planning and scheduling of manufacturing processes is inevitable in order to obtain feasible production plans and schedules (see e.g., Stadtler & Kilger, 2008). The following case occurs frequently in practice and constitutes the main motivation of this paper. We have to schedule the production of some parts on a production line over the next week, and we have an initial stock and expect some additional shipments from the suppliers over the week. Our goal is to minimize the maximum of the late deliveries, or in other words, the lateness. Since the parts may require common raw materials for their production, it is not obvious how to allocate the supplies to the parts to produce. The arising optimization problem is precisely the topic of this paper.

More formally, we focus on scheduling a single machine subject to raw material constraints. That is, in addition to the machine, there are some raw materials with an initial stock and some additional replenishments over time with a priori known dates and quantities. Jobs may require various quantities from these resources, and a job can be started only if the required amount is on stock. Upon starting a job, the stock level of all the resources are decreased by the quantities needed by the job. Each job has a due-date and the objective is to minimize the maximum lateness. As an illustration, consider Fig. 1 in which a schedule of two jobs is shown on a single machine, and notice that job J_2 must wait until the replenishment of the raw-material, because the first scheduled job decreases the stock level below its requirement.

The above model has been first studied by Carlier (1984), and by Slowinski (1984). In particular, Carlier has shown that minimizing the maximum job completion time (makespan) is NP-hard in the strong sense in general. This implies that our problem is NP-hard in the strong sense as well. Over the years, a number of papers appeared dealing with some variants and proposing either complexity results (Gafarov, Lazarev, & Werner, 2011; Toker, Kondakci, & Erkip, 1991; Xie, 1997), or approximation algorithms (Grigoriev, Holthuijsen, & van de Klundert, 2005; Györgyi & Kis, 2015a, 2015b; Györgyi & Kis, 2017). However, there are only sporadic computational results on this problem. Grigoriev et al. (2005) have provided some test results for one of their approximation algorithms. Belkaid, Maliki, Boudahri, and Sari (2012) propose lower bounds and heuristics for minimizing the makespan in a parallel machine environment with non-renewable resource constraints.

To our best knowledge, no exact method has been described for our problem in the literature. However, for a related problem, where some of the jobs produce, while other jobs consume some non-renewable resources (and there are no replenishments from external sources) Briskorn, Jaehn, and Pesch (2013) propose an exact method for minimizing the total weighted completion time of the jobs. In the more general project scheduling setting, Neumann and Schwindt (2003) study the makespan minimization problem with inventory constraints, and describe a branch-and-bound method for solving it.

Single machine scheduling with the maximum lateness objective is polynomially solvable by ordering the jobs in earliest due-date order, see Jackson (1955). In spite of the existence of a polynomial time algorithm, we are not aware of any linear programming based method of polynomial time complexity in which the coefficients of the variables are determined by polynomial functions of the problem data. That is, we require that from any input with n jobs we should be able to get the

* Corresponding author.
E-mail addresses: gyorgyi.peter@sztaki.mta.hu (P. Györgyi), kis.tamas@sztaki.mta.hu (T. Kis).

https://doi.org/10.1016/j.cie.2017.11.016
Received 28 June 2017; Received in revised form 13 November 2017; Accepted 15 November 2017
Available online 16 November 2017
0360-8352/

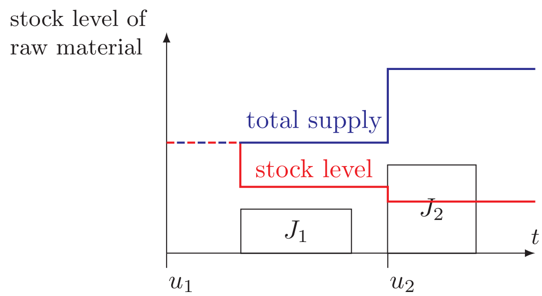

Fig. 1. Illustration of the problem. The height of a job indicates the amount of the required raw material.

LP formulation by plugging the problem data into multivariate polynomials that yield the coefficients of the decision variables. However, it is not allowed to insert new constraints, or do some sorting and then fill in the coefficients of the variables and the right-hand-sides in the LP. In fact, Blazewicz, Dror, and Weglarz (1991) propose a MIP formulation for $1||L_{\max}$ using positional variables. Moreover, a number of alternative formulations are compared and evaluated for single machine scheduling problems with various objective function by Keha, Khowala, and Fowler (2009). Some of the models of Keha et al. find their roots in the MIP model of Manne (1960) for the job-shop scheduling problem using completion time variables and ordering variables for each pair of distinct jobs requiring the same machine. In contrast, for single machine scheduling with the sum of (weighted) job completion times objective, $1||\sum w_j C_j$, an LP formulation is developed by Queyranne (1993) in which the coefficients of the constraints are linear functions of the problem data, while the right-hand-sides are determined by quadratic polynomials of the data. Although the number of inequalities is exponential in the number of jobs, but they can be separated efficiently, so the LP can be solved in polynomial time. We will adapt the inequalities of Queyranne in Section 3.2 to our MIP model.

Main results and structure of the paper. Firstly, we will elaborate upon the modeling of the problem by a mixed-integer linear program (MIP). Since we have to compute the maximum lateness objective, choosing the right MIP model is a non-trivial issue (Section 2). Second, we will devise new inequalities valid for the feasible solutions of the MIP formulation, and also two which may cut off feasible solutions, but they keep at least one optimal solution (Section 3.2). The new inequalities will be used in a branch-and-cut method to strengthen the LP-relaxation of the MIP formulation (Section 3.1). We will also sketch a heuristic method for getting an initial feasible solution as well as an upper bound on the optimum value in Section 3.3. We emphasize that in most papers mentioned above, mathematical programs are used only for modeling the problem, while the methods devised are based on some other representations. In contrast, our branch-and-cut method uses the MIP model as the representation of the problem, and we do not use the solver as a black-box, instead, we generate cutting planes in the course of the solution process in order to speed up the optimization algorithm. Thirdly, we summarize our computational results on a large set of benchmark instances. The goal of the experiments is to determine the limitation of the method, and also to assess the benefit of using cutting planes to strengthen the MIP formulation (Section 4). Finally, we conclude the paper in Section 5.

2. Problem formulation

In this section first we define our problem more formally, then describe our MIP formulation in several steps.

In our scheduling problem there is a single machine, a set of n jobs $\mathscr{J}$, and a set of ρ non-renewable resources $\mathscr{R}$. Each job j has a processing time $p_j > 0$, a due-date $d_j \geqslant 0$, and resource requirements $a_{ij} \geqslant 0$ for $i \in \mathscr{R}$. The non-renewable resources are supplied at dates $0 = u_1 < u_2 < \cdots < u_q$, and the amount supplied from resource $i \in \mathscr{R}$ at date u_ℓ is $\tilde{b}_{i,\ell} \geqslant 0$. All problem data are non-negative and integer.

A *schedule* specifies the starting time S_j of each job $j \in \mathscr{J}$; it is feasible if (i) the jobs are not preempted, (ii) no two distinct jobs overlap in time, i.e., $S_{j_1} + p_{j_1} \leqslant S_{j_2}$ or $S_{j_2} + p_{j_2} \leqslant S_{j_1}$ for each pair of distinct jobs j_1 and j_2, and (iii) for each resource $i \in \mathscr{R}$, and for each time point t, the total supply until time t is not less than the total consumption of those jobs starting not later than t, i.e., if $u_\ell \leqslant t$ is the last supply date no later than t, then $\sum_{j\in J:S_j\leqslant t} a_{ij} \leqslant \sum_{k=1}^{\ell} \tilde{b}_{ik}$ for each resource $i \in \mathscr{R}$. We aim at finding a feasible schedule S minimizing the maximum lateness $L_{\max}(S) := \max_{j\in\mathscr{J}} C_j(S) - d_j$, where $C_j(S) = S_j + p_j$ is the completion time of job j in schedule S.

We may assume that for each $i \in \mathscr{R}$, the total demand does not exceed the total supply, i.e., $\sum_{j\in\mathscr{J}} a_{ij} \leqslant \sum_{\ell=1}^{q} \tilde{b}_{i\ell}$, otherwise no feasible solution exists. The cumulative supply of resource i up to supply date u_ℓ is $b_{i\ell} := \sum_{k=1}^{\ell} \tilde{b}_{ik}$.

Keha et al. (2009) describe 4 distinct MIP formulations for single machine scheduling with the maximum lateness objective ($1||L_{\max}$). None of these formulations take non-renewable resources into account, but any of them could be further developed to model our problem. We have ruled out the time-indexed formulation, since in that model the number of variables linearly depends on the magnitude of the job processing times, and should we extended that model by non-renewable resource constraints, also on the magnitude of the supply dates. After some preliminary tests (we extended each model of Keha et al. by modeling the non-renewable resource constraints), we have chosen the model with *completion time variables*, and we describe it in detail subsequently.

We use three main types of variables in our formulation. Variable C_j denotes the completion time of job $j \in \mathscr{J}$ and for each ordered pair of jobs $j_1 j_2 \in \mathscr{J}$ with $j_1 < j_2$, the binary variable $ord_{j_1 j_2}$ has value 1 if and only if j_1 precedes j_2 in the schedule. Finally, there are $q\cdot|\mathscr{J}|$ binary decision variables $z_{j\ell} j \in \mathscr{J}$, $\ell = 1,\ldots,q$, to assign jobs to supplies, i.e., $z_{j\ell} = 1$ if and only if job j can be started before $u_{\ell+1}$ ($u_{q+1} = \infty$), i.e., the first ℓ supplies can cover its resource requirements along with all other jobs $j' \neq j$ with $z_{j'\ell} = 1$. Then $z_{j\ell} \geqslant z_{j,\ell-1}$ must hold and if $z_{j\ell} - z_{j,\ell-1} = 1$ then job j must not start before u_ℓ. The MIP formulation is

$$\text{minimize} \quad L_{\max} \tag{1}$$

subject to

$$C_j \geqslant p_j, \quad j \in \mathscr{J} \tag{2}$$

$$C_{j_1} + p_{j_2} \leqslant C_{j_2} + M\cdot(1 - ord_{j_1 j_2}), \quad j_1 j_2 \in \mathscr{J}, j_1 < j_2 \tag{3}$$

$$C_{j_2} + p_{j_1} \leqslant C_{j_1} + M\cdot ord_{j_1 j_2}, \quad j_1 j_2 \in \mathscr{J}, j_1 < j_2 \tag{4}$$

$$L_{\max} \geqslant C_j - d_j, \quad j \in \mathscr{J} \tag{5}$$

$$C_j - p_j \geqslant \sum_{\ell=2}^{q} u_\ell\cdot(z_{j,\ell} - z_{j,\ell-1}), \quad j \in \mathscr{J} \tag{6}$$

$$\sum_{j\in\mathscr{J}} a_{ij} z_{j\ell} \leqslant b_{i\ell}, \quad \ell = 1,\ldots,q-1,\ i \in \mathscr{R} \tag{7}$$

$$z_{j,\ell-1} \leqslant z_{j,\ell}, \quad j \in \mathscr{J}, \ \ell = 2,\ldots,q \tag{8}$$

$$z_{j,q} = 1, \quad j \in \mathscr{J} \tag{9}$$

$$ord_{j_1 j_2} \in \{0,1\}, \quad j_1 j_2 \in \mathscr{J}, j_1 < j_2 \tag{10}$$

$$z_{j\ell} \in \{0,1\}, \quad j \in \mathscr{J}, \ \ell = 1,\ldots,q. \tag{11}$$

The objective is to minimize $L_{\max}$. Constraints (2)–(4) ensure that the jobs do not overlap in time. Inequalities (5) express that $L_{\max}$ is at least $\max_{j\in\mathscr{J}} C_j - d_j$. By (6) for each job j, the starting time $C_j - p_j$ is at least the u_ℓ provided that $z_{j\ell} - z_{j,\ell-1} = 1$. The resource constraints are encoded by (7), since if $z_{j\ell} = 1$ then job j can be started before $u_{\ell+1}$, hence, its resource consumption must be satisfied from the cumulative supply $b_{i\ell}$, for each $i \in \mathscr{R}$. The rest of the constraints order the $z_{j\ell}$, set $z_{jq} = 1$, since