European Journal of Operational Research 206 (2010) 301–312

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

A unified approach for scheduling with convex resource consumption functions using positional penalties

Yaron Leyvand [a], Dvir Shabtay [a], George Steiner [b,*]

[a] Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Beer-Sheva, Israel
[b] Operations Management Area, DeGroote School of Business, McMaster University, Hamilton, Ontario, Canada

ARTICLE INFO

Article history:
Received 29 May 2009
Accepted 18 February 2010
Available online 21 February 2010

Keywords:
Single machine scheduling
Controllable processing times
Resource allocation
Due date assignment
Positional penalties
Polynomial-time algorithm

ABSTRACT

We provide a unified model for solving single machine scheduling problems with controllable processing times in polynomial time using positional penalties. We show how this unified model can be useful in solving three different groups of scheduling problems. The first group includes four different due date assignment problems to minimize an objective function which includes costs for earliness, tardiness, due date assignment, makespan and total resource consumption. The second group includes three different due date assignment problems to minimize an objective function which includes the weighted number of tardy jobs, due date assignment costs, makespan and total resource consumption costs. The third group includes various scheduling problems which do not involve due date assignment decisions. We show that each of the problems from the first and the third groups can be reduced to a special case of our unified model and thus can be solved in $O(n^3)$ time. Furthermore, we show how the unified model can be used repeatedly as a subroutine to solve all problems from the second group in $O(n^4)$ time. In addition, we also show that faster algorithms exist for several special cases.

1. Introduction

For the majority of deterministic scheduling problems in the literature, job processing times are considered to be fixed. In various real-life systems, however, processing times may be controllable by allocating resources, such as additional money, overtime, energy, fuel, catalysts, subcontracting, or additional manpower, to the job operations. In such systems, job scheduling and resource allocation decisions should be coordinated carefully to achieve the most efficient system performance.

Janiak [20] described an interesting application of a scheduling problem with controllable processing times in steel mills, where batches of ingots have to be preheated before being hot-rolled in a blooming mill, and both the preheating time and the rolling time are decreasing functions of the gas flow intensity. Another interesting application arises from scheduling in a machine-tooling environment, where the job processing time is a function of the feed rate and the spindle speed used for each operation (see [42]). Subcontracting can also be modeled by controllable processing times [40], where the compressed portion of the processing time corresponds to the subcontracted part of each job. Due to the large variety of applications, there is extensive literature on the subject of scheduling with controllable processing times (e.g., [2,22,16,8, 26,34,35,27]). A survey of results up to 1990 is provided by Nowicki and Zdrzalka [28] and a more recent one is given by Shabtay and Steiner [39].

In most of the above-mentioned studies on scheduling with controllable processing times, it was assumed that the job processing time is a bounded linear function of the amount of resource allocated to the processing of the job, i.e., the *resource consumption function* takes the form

$$p_j(u_j) = \overline{p}_j - a_j u_j, \qquad j = 1, \ldots, n, \quad 0 \leqslant u_j \leqslant \overline{u}_j < \overline{p}_j / a_j, \tag{1}$$

where n is the number of non-preemptive jobs, u_j is the amount of resource allocated to job j, $\overline{p}_j$ is the non-compressed processing time for job j, $\overline{u}_j$ is the upper bound on the amount of resource that can be allocated to job j and a_j is the positive compression rate of job j. However, for many resource allocation problems in physical or economic systems, a linear resource consumption function has only limited application, since it fails to reflect the *law of diminishing returns.* This law states that productivity increases at a decreasing rate with the amount of resource employed. One class of models reflecting this law (e.g., [25,34,35,41]) uses a convex resource consumption function described by the following equation

$$p_j(u_j) = \left(\frac{w_j}{u_j}\right)^k, \tag{2}$$

where w_j is a positive parameter that represents the *workload* of the processing operation for job j, and k is a positive constant. This

* Corresponding author.
E-mail addresses: leyvand@bgu.ac.il (Y. Leyvand), dvirs@bgu.ac.il (D. Shabtay), steiner@mcmaster.ca (G. Steiner).

0377-2217/$ - see front matter Crown Copyright © 2010 Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2010.02.026

resource consumption function has been used extensively in continuous resource allocation (e.g., [25,32,3,4,34,35]). In fact, Monma et al. [25] pointed out that $k = 1$ corresponds to many actual government and industrial operations and that the $k = 0.5$ arises from VLSI (very large scale integration) circuit design, where the product of the silicon area (resource) and the square of time spent equals a constant value (the workload) for an individual job.

There are some studies [9,46], however, which use more general resource consumption functions, as even Eq. (2) may fail to accurately model the resource consumption for certain applications. The relationship between the required processing time and the resources employed is more complex in these applications. For example, the time required to preheat steel ingots to the appropriate temperature before rolling them in a steel mill is a convex, decreasing (usually exponential) function of the energy (gas flow intensity) applied [45]. Another example is when the processing time initially decreases proportionally (linearly) with the amount of resource allocated to the job, but after a certain point, it becomes much more expensive to further reduce the job processing time because it starts to follow the law of diminishing returns. Furthermore, each job may have a different type of resource consumption function depending on its characteristics (e.g. some linear or linear in part, others different convex, etc.). Thus, there is also a need to develop algorithms for solving scheduling problems with convex, decreasing resource consumption functions which are more general than the families described by Eq. (1) or Eq. (2). In this paper, we present polynomial-time optimization algorithms for many different scheduling problems with general convex, decreasing resource consumption functions, which *can vary* even between or even within jobs and only have to satisfy the following not very restrictive properties (we refer by $g - conv$ to the set of functions satisfying Properties 1–4):

Property 1. *Each job's processing time, $p_j(u_j)$, is a bounded, differentiable function of the amount of resource u_j allocated to the job, where $p_j(u_j) : \Re^+ \rightarrow \Re^+$, and $\underline{u}_j$ and $\overline{u}_j$ are the lower and upper bounds, respectively, for the amount of resource that can be allocated to job j for $j = 1, \ldots, n$.*

Property 2. *Each job's processing time is a non-increasing function of the amount of resource allocated to the job, i.e., $dp_j(u_j)/du_j \leqslant 0$ for $u_j \in [\underline{u}_j, \overline{u}_j]$. (We will also use the notation $p'_j(u_j)$ for the derivative $dp_j(u_j)/du_j$, where convenient.)*

Property 3. *Each $p_j(u_j)$ is convex, i.e., $d^2p_j(u_j)/(du_j)^2 \geqslant 0$ for $u_j \in [\underline{u}_j, \overline{u}_j]$.*

Property 4. *For any $y \in [p'_j(\underline{u}_j), p'_j(\overline{u}_j)]$, a unique point u_j exists such that $p'_j(u_j) = y$ and it can be determined in constant time.*

The general problem we study in this paper may be stated as follows: n independent, non-preemptive jobs, $J = \{1, 2, \ldots, n\}$, are available for processing at time zero and are to be processed on a single machine. Each job processing time, $p_j(u_j)$, is a function of the resource allocated to the job, u_j, where $p_j(u_j)$ satisfies Properties 1–4, and v_j is the cost of one unit of resource allocated to job j. A schedule is defined by a job sequence $\pi = ([1], [2], \ldots, [n])$ and a resource allocation vector $\mathbf{u} = (u_1, u_2, \ldots, u_n)$, where $[j]$ represents the job that is in the jth position in π for $j = 1, 2, \ldots, n$. Our objective is to determine a schedule which minimizes a general unified cost function that is the sum of scheduling costs, expressed by using positional penalties, and the resource consumption costs. This unified cost function can be formulated as follows:

$$Z(\pi, \mathbf{u}) = \sum_{j=1}^{n} \xi_j p_{[j]}(u_{[j]}) + \sum_{j=l+1}^{n} \psi_{[j]} + \sum_{j=1}^{n} v_{[j]} u_{[j]} = \sum_{j=1}^{n} z_{[j]}(u_{[j]}), \quad (3)$$

where ξ_j is a *positional, job-independent penalty* for *any* job scheduled in the jth position, $\psi_{[j]}$ is a *constant job-dependent penalty* for jobs that are sequenced after the lth position in π (the index l can always be determined from the more specific form of the scheduling cost), and

$$z_{[j]}(u_{[j]}) = \begin{cases} \xi_j p_{[j]}(u_{[j]}) + v_{[j]} u_{[j]} & \text{for } j = 1, \ldots, l, \\ \xi_j p_{[j]}(u_{[j]}) + \psi_{[j]} + v_{[j]} u_{[j]} & \text{for } j = l+1, \ldots, n, \end{cases}$$

by definition. The term

$$f(\mathbf{u}) = \sum_{j=1}^{n} v_{[j]} u_{[j]} = \sum_{j=1}^{n} v_j u_j, \quad (4)$$

in (3) is the *total resource consumption cost*, and the rest of the objective,

$$g(\pi, \mathbf{u}) = \sum_{j=1}^{n} \xi_j p_{[j]}(u_{[j]}) + \sum_{j=l+1}^{n} \psi_{[j]}, \quad (5)$$

will be referred to as the *scheduling cost*.

We will use and extend the standard three field notation $A|B|C$ introduced by Graham et al. [15] for scheduling problems. The A field describes the machine environment. For example, if 1 appears in the A field, it means that we deal with a single machine scheduling problem. The B field defines the job processing characteristics and constraints. We also include in the B field the information needed about the type of resource consumption function. For example, if *lin* appears in this field, it means that a linear resource consumption function given by Eq. (1) is assumed, *conv* means that we use Eq. (2) for describing the resource consumption function, and *g–conv* means that we assume a general convex resource consumption function that only has to satisfy Properties 1–4. The C field contains the optimizing criteria (for ease of presentation, shown in a symbolic fashion without including the summation indices, for example, $\sum_{j=l+1}^{n} \psi_{[j]}$ is replaced by $\sum \psi_{[j]}$).

The paper is organized as follows. In Section 2, we present an $O(n^3)$ optimization algorithm to determine the optimal schedule for the $1|g{-}conv|\sum \xi_j p_{[j]} + \sum \psi_{[j]} + \sum v_{[j]} u_{[j]}$ unified problem. In the next section, we present two groups of scheduling problems which involve due date assignment and resource allocation decisions. In the first group, the objective function includes costs for earliness, tardiness, makespan, due date assignment and resource allocation. We show that, for four different due date assignment methods, the objective can be reformulated as a special case of (3), which enables us to solve this set of problems in $O(n^3)$ time. The second group contains three problems involving due date assignment and resource allocation decisions in which the objective function includes costs for the number of tardy jobs, makespan, due date assignment and resource allocation. We show how we can solve this set of problems by solving at most n problems with an objective that is again a special case of (3). This yields an $O(n^4)$ time solution for this set of problems. In addition, we show how the complexity can be reduced from $O(n^4)$ to $O(n)$ for two of the due date assignment methods. In Section 4, we show how the unified approach can be used to solve some additional classical scheduling problems with controllable processing times. These include minimizing the makespan, the sum of completion times, the variation of job completion times and the variation of waiting times with controllable processing times. The last section contains a summary and our concluding remarks.

2. The unified problem

In the following crucial lemma, we present the optimal resource allocations as a function of the assignment of jobs to positions in the sequence. It is important to emphasize that the lemma allows