Computers & Operations Research 70 (2016) 115–126

Contents lists available at ScienceDirect

Computers & Operations Research

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

Metaheuristics for the single machine weighted quadratic tardiness scheduling problem

Tomás C. Gonçalves [a], Jorge M.S. Valente [b,*], Jeffrey E. Schaller [c]

[a] Faculdade de Economia da Universidade do Porto, C/O Jorge Valente, Rua Dr. Roberto Frias, s/n, 4200-464 Porto, Portugal
[b] LIAAD – INESCTEC LA, Faculdade de Economia da Universidade do Porto, Rua Dr. Roberto Frias, s/n, 4200-464 Porto, Portugal
[c] Department of Business Administration, Eastern Connecticut State University, 83 Windham St., Willimantic, CT 06226-2295, USA

ARTICLE INFO

Available online 12 January 2016

Keywords:
Scheduling
Single machine
Weighted quadratic tardiness
Metaheuristics
Iterated local search
Variable greedy
Genetic algorithm

ABSTRACT

This paper considers the single machine scheduling problem with weighted quadratic tardiness costs. Three metaheuristics are presented, namely iterated local search, variable greedy and steady-state genetic algorithm procedures. These address a gap in the existing literature, which includes branch-and-bound algorithms (which can provide optimal solutions for small problems only) and dispatching rules (which are efficient and capable of providing adequate solutions for even quite large instances). A simple local search procedure which incorporates problem specific information is also proposed.

The computational results show that the proposed metaheuristics clearly outperform the best of the existing procedures. Also, they provide an optimal solution for all (or nearly all, in the case of the variable greedy heuristic) the smaller size problems. The metaheuristics are quite close in what regards solution quality. Nevertheless, the iterated local search method provides the best solution, though at the expense of additional computational time. The exact opposite is true for the variable greedy procedure, while the genetic algorithm is a good all-around performer.

1. Introduction

This paper considers a single machine scheduling problem with weighted quadratic tardiness costs. Formally, the problem can be stated as follows. A set of n independent jobs $\{1, 2, \cdots, n\}$ is to be scheduled on a single machine that can handle only one job at a time. The machine is continuously available from time zero onwards, and preemptions are not allowed. Job $j, j = 1, 2, \cdots, n$, requires a processing time p_j, has a weight w_j and should ideally be completed by its due date d_j. For a given schedule, the tardiness of job j is defined as $T_j = \max\{C_j - d_j; 0\}$, where C_j is the completion time of job j. The objective is then to find a schedule that minimizes the sum of the weighted squared tardiness values $\sum_{j=1}^{n} w_j T_j^2$.

Single machine scheduling environments may appear to arise infrequently in practice. However, they actually occur in several practical settings. A specific example, arising in the chemical industry, is given in [1]. Scheduling models with a single machine are also useful for problems with multiple processors. Indeed, a single bottleneck machine is often the source of inefficiency in many production systems. Therefore, the performance of these systems will then depend mainly on the quality of the schedules generated for this single bottleneck processor. Moreover, the study of single machine problems provides results and insights that prove valuable for scheduling more complex settings, such as parallel machines, flow shops or even job shops.

The objective function considers squared tardiness costs. Tardiness is a widely used performance measure in scheduling, since tardy deliveries can result in contractual penalties, lost sales and loss of customer goodwill. A squared tardiness is used in this paper, instead of the more usual (in the literature) linear tardiness or maximum tardiness alternatives. Each of these three measures has its merits, and neither is intrinsically better. Indeed, every one of these criteria can be appropriate, depending on the setting and the goals and preferences of the decision maker.

A maximum tardiness criterion is adequate when the main objective is to prevent a quite large delay. As detailed in [2], however, maximum tardiness focuses on the job with the largest delay, and disregards the tardiness that might be incurred in all the other jobs. Thus, if the decision maker wishes to take into account all delays and all customers, measures such as linear or squared tardiness are preferable. The choice between linear or quadratic again depends on setting and preferences.

Under a linear tardiness, the distribution of the overall total tardiness is irrelevant. That is, a sequence in which only one or a few jobs are quite tardy is equivalent to another sequence where

* Corresponding author. Tel.: +351 225 571 100; fax: +351 225 505 050.
E-mail addresses: tomcabrita@hotmail.com (T.C. Gonçalves), jvalente@fep.up.pt (J.M.S. Valente), schallerj@easternct.edu (J.E. Schaller).

http://dx.doi.org/10.1016/j.cor.2016.01.004
0305-0548/

multiple jobs are only a little tardy, as long as the sum of the tardiness values is the same. A quadratic tardiness objective function, however, severely penalizes larges values of the tardiness, and will usually avoid schedules in which a single or only a few jobs contribute the majority of the cost, as described in more detail in [2].

In the same line, and as highlighted in [3,4], in linear tardiness the incremental penalty of a job does not change as the tardiness increases, so two jobs each one time unit late are equivalent to one job two units late. In contrast, under a squared tardiness measure, the incremental penalty of a job does increase as the tardiness increases, so one job two time units late incurs a larger cost than two jobs each one unit late. Furthermore, the loss function of Taguchi [5] proposes that a customer's dissatisfaction tends to increase quadratically with the tardiness, instead of linearly. Thus, a squared tardiness objective is relevant in practice; indeed, the scheduling methodology developed by Hoitomt et al. [3] used the quadratic tardiness objective and was actually implemented as part of a knowledge-based scheduling system at Pratt and Whitney.

The considered problem has previously been studied in [6,7]. Schaller and Valente [6] developed several dominance rules, as well as branch-and-bound procedures which incorporated these rules. Valente and Schaller [7], on the other hand, proposed and analyzed several efficient dispatching rules. To the best of our knowledge, only a limited number of other papers have considered a weighted quadratic tardiness objective function. Hoitomt et al. [3] developed a solution procedure based on lagrangean relaxation for parallel machines problems with simple precedence constraints, and demonstrated this procedure using three examples. Sun et al. [2] analyzed several heuristics for the problem with a single machine, release dates and sequence dependent setup times. Finally, a job shop scheduling problem with alternative processing plans was studied by Thomalla [4], who compared, on three small examples, a lagrangean relaxation based lower bound and heuristics with other methods.

The complexity of the single machine weighted quadratic tardiness problem is, again to the best of our knowledge, still open. However, and given existing complexity results, it seems most likely that the problem is hard. Indeed, the corresponding linear problem, i.e. the total weighted tardiness problem, is strongly NP-hard [8,9].

Two streams of research on single machine scheduling that are related to the considered problem are models with a quadratic performance measure and the total weighted tardiness problem. Among tardiness-related quadratic performance measures, the quadratic lateness problem has been studied by Gupta and Sen [10], Sen et al. [11], Su and Chang [12], Schaller [13] and Soroush [14,15]. Also, the linear earliness and squared tardiness problem was considered by Schaller [16], Valente [17–19], Valente and Schaller [20], and Behnamian and Zandieh [21]. The problem with both quadratic earliness and quadratic tardiness costs was studied by Valente and Alves [22], Valente and Moreira [23], Valente [24], Valente et al. [25], Singh et al. [26], Kianfar and Moslehi [27], and Vilà and Pereira [28]. A large number of papers have been published on the total weighted tardiness problem. Exact methods have been surveyed and compared in [29], and several heuristic methods were analyzed in [30]. Sen et al. [31] provide a more recent literature review of both exact and heuristic procedures for this linear problem.

This paper presents three metaheuristics, namely iterated local search, variable greedy and steady-state genetic algorithm procedures. These heuristics address a gap in the existing literature. Indeed, and as previously mentioned, the existing procedures consist of branch-and-bound algorithms, which can provide an optimal solution for small instances, and efficient dispatching rules, which are often the only heuristic approach capable of providing solutions, in reasonable time, for large problems. Metaheuristics are often quite valuable for medium sized instances, since they are usually able to provide high quality solutions (superior to those of dispatching rules) within acceptable computational times. A local search procedure, which is used in the metaheuristics, is also presented. This proposed local search is essentially an adjacent pairwise interchange procedure, which incorporates problem specific information.

The remainder of this paper is organized as follows. The local search procedure is described in Section 2. Section 3 presents the three proposed metaheuristics. The computational results are reported in Section 4. Finally, some concluding remarks are provided in Section 5.

2. Local search procedure

In this section, the proposed local search procedure is described. As previously mentioned, the local search is essentially an adjacent pairwise interchange improvement procedure. Therefore, after the application of the local search, no further improvement in the sequence is possible by swapping any pair of adjacent jobs. However, the procedure incorporates problem specific information. The pseudo-code for the proposed local search is given in Procedure 1. In this context, let i be a position in a sequence and $[i]$ be the job in position i.

Procedure 1. Local search procedure

1. Set $i=1$.
2. While $i<n$:
   - 2.1. If jobs $[i]$ and $[i+1]$ are early:
     - 2.1.1. If $d_{[i]} > d_{[i+1]}$:
       - 2.1.1.1. Swap jobs $[i]$ and $[i+1]$.
       - 2.1.1.2. If $i>1$, set $i=i-1$.
     - 2.1.2. Otherwise, set $i=i+1$.
   - 2.2. Else if jobs $[i]$ and $[i+1]$ are tardy:
     - 2.2.1. If $w_{[i]}(2T_{[i]}+1)p_{[i+1]} < w_{[i+1]}(2T_{[i+1]}+1)p_{[i]}$:
       - 2.2.1.1. If the objective function value is improved by swapping jobs $[i]$ and $[i+1]$:
         - 2.2.1.1.1. Swap jobs $[i]$ and $[i+1]$.
         - 2.2.1.1.2. If $i>1$, set $i=i-1$.
       - 2.2.1.2. Otherwise, set $i=i+1$.
     - 2.2.2. Otherwise, set $i=i+1$.
   - 2.3. Else if job $[i]$ is early and job $[i+1]$ is tardy:
     - 2.3.1. If $d_{[i]} \ge C_{[i+1]}$:
       - 2.3.1.1. Swap jobs $[i]$ and $[i+1]$.
       - 2.3.1.2. If $i>1$, set $i=i-1$.
     - 2.3.2. Else if $w_{[i]}(C_{[i+1]}-d_{[i]})^2 < w_{[i+1]}(2T_{[i+1]}+1)p_{[i]}$:
       - 2.3.2.1. If the objective function value is improved by swapping jobs $[i]$ and $[i+1]$:
         - 2.3.2.1.1. Swap jobs $[i]$ and $[i+1]$.
         - 2.3.2.1.2. If $i>1$, set $i=i-1$.
       - 2.3.2.2. Otherwise, set $i=i+1$.
     - 2.3.3. Else, set $i=i+1$.
   - 2.4. Else, set $i=i+1$.

The procedure starts at the first position in the sequence, and stops when the final position is reached. At each iteration, the jobs at the current and next positions are analyzed. If the two jobs are swapped, the procedure backtracks one position when possible, since a new comparison can now be made. Otherwise, the procedure moves forward by one position.