Accepted Manuscript

On the complexity of the single machine scheduling problem minimizing total weighted delay penalty

Oscar C. Vasquez

PII: S0167-6377(14)00078-9
DOI: http://dx.doi.org/10.1016/j.orl.2014.05.009
Reference: OPERES 5832

To appear in: *Operations Research Letters*

Received date: 17 September 2013
Revised date: 20 May 2014
Accepted date: 21 May 2014

On the complexity of the single machine scheduling problem minimizing total weighted delay penalty

Oscar C. Vasquez[a]

[a]LIP6, Univ. Pierre et Marie Curie, Paris and Industrial Eng. Dep., Univ. of Santiago of Chile. supported by ANR-11-BS02-0015

Abstract

We consider a single machine scheduling problem for a fixed penalty function f. For this problem every job j has a priority w_j, a processing time p_j and the goal is to find an order on the given jobs that minimizes $\sum w_j f(C_j)$, where C_j is the completion time of the job j. This paper studies the complexity of this problem for various classes of penalty functions.

Keywords: Scheduling, penalty function, complexity

1. Introduction

Typically in a scheduling problem, we are given a number of jobs, each of which has some processing time and some priority weight, which have to be scheduled on a single machine in order to minimize a particular objective function. In this paper we consider objective functions of the form $\sum_j w_j f(C_j)$, where w_j is the priority weight of job j, C_j the completion time of job j and f is a problem specific penalty function. Extending the three-field notation [8] our problem could be denoted as $1||\sum w_j f(C_j)$.

Most of the penalty functions considered in literature are monotone increasing, which forces optimal schedules to be *left-shifted*, meaning that all executions happen without idle time between time 0 and the total job processing time. In this case the problem is said to have the *left-shifted property*. The study of different penalty functions has several motivations. The first one concerns machines with speed varying over time. If we denote by s the speed function, then after t_0 time units, the machine processed workload $\int_0^{t_0} s(t)dt$. Minimizing the weighted sum of the completion times in this setting reduces to minimizing the value $\sum w_j f(C_j)$ for a constant speed machine, where f is the inverse of the integral of s. Varying speed can be the consequence of a learning effect, upgrades, or even varying available energy or tear and wear of the machine. The second motivation was mentioned in [29, 21], where the authors considered a particular dynamic voltage scheduling problem with the goal of optimizing a linear combination between total weighted completion time and overall energy consumption. It was shown that this problem reduces to constant speed scheduling problem with the objective function $\sum w_j C_j^\beta$ for some constant $0 < \beta < 1$.

In this paper we survey penalty functions with several elementary properties. We consider convex and concave functions, increasing and decreasing functions, strictly monotone functions and non monotone functions. Among those classes we consider piecewise linear functions with two slopes for simplicity and restrict ourself to left-shifted schedules, even when the penalty functions are not monotone increasing.

Most of the previous work focused on convex functions. In this class of functions, NP-hardness proofs consist generally in reductions from P - and involve instances, where all jobs j have the same Smith-ratio w_j/p_j, with the exception of a single job. Our contribution is to show that for any increasing concave penalty function these so-called *almost equal ratio instances* are easy, ruling out a trivial adaption of the NP-hardness proofs. In addition we model the scheduling problem for a particular concave penalty function as the minimization of *half-products*, inheriting therefore of pseudopolynomial time algorithms and FPTAS for this problem.

2. Literature Review

We list a few scheduling problems for several penalties functions, whose complexity has been studied in the literature.