

# Parameter Learning Online Algorithm for Multiprocessor Scheduling with Rejection\*

Tamás Németh<sup>†</sup> and Csanád Imreh<sup>†</sup>

## Abstract

In multiprocessor scheduling with rejection the jobs are characterized by a processing time and a penalty and it is possible to reject the jobs. The goal is to minimize the makespan of the schedule for the accepted jobs plus the sum of the penalties of the rejected jobs. In this paper we present a new online algorithm for the problem. Our algorithm is a parameter learning extension of the total reject penalty algorithm. The efficiency of the algorithm is investigated by an experimental analysis.

**Keywords:** online algorithms, scheduling, experimental analysis

## 1 Introduction

In this paper we develop a new algorithm for the solution of the online scheduling with rejection problem on identical machines. The algorithm is based on the idea of learning the parameter of the Reject Total Penalty (RTP) algorithm. We measure the efficiency of the new algorithm by an experimental analysis.

The problem of scheduling with rejection is defined in [2]. In this model, it is possible to reject the jobs. The jobs are characterized by a processing time and a penalty. The goal is to minimize the makespan of the schedule for the accepted jobs plus the sum of the penalties of the rejected jobs. In the online case a 2.618-competitive algorithm is given for arbitrary number of machines. This algorithm is called Reject Total Penalty (RTP). One basic idea in scheduling with rejection is to compare the penalty and the load (processing time divided by the number of machines) of the job, and reject the job in the case when the penalty is smaller. This greedy algorithm can make a bad decision when the number of machines is large and this makes possible to appear large jobs with small loads. RTP handles these jobs more carefully. We give the detailed definition in the next section. In [2] a further, 1.618-competitive algorithm is presented in the case of 2 machines. Matching lower bounds are also given. In the offline case an FPTAS is presented for fixed number

---

\*This research has been supported by the Hungarian National Foundation for Scientific Research, Grant F048587.

<sup>†</sup>Institute of Informatics, University of Szeged E-mail: {tnemeth,cimreh}@inf.u-szeged.hu