Proceedings of the 13th IFAC Symposium on
Information Control Problems in Manufacturing
Moscow, Russia, June 3-5, 2009

Rescheduling for new orders on a single machine with setup times

H. Hoogeveen * **C. Lenté** ** **V. T'kindt** ***

* *Utrecht University, PO Box 80.089, 3508TB Utrecht, The Netherlands (e-mail: slam@cs.uu.nl).*
** *Université Francois Rabelais Tours, Laboratory of Computer Sciences, 64 avenue Jean Portalis, 37230 Tours, France (e-mail: lente@univ-tours.fr)*
*** *Université Francois Rabelais Tours, Laboratory of Computer Sciences, 64 avenue Jean Portalis, 37230 Tours, France (e-mail: tkindt@univ-tours.fr)*

Abstract: We focus on some single machine scheduling problems for which a set of new jobs have to be scheduled after a schedule of old jobs has been set. Each new and old job belongs to a family and changing the production from one family to another requires a setup. The initial schedule of old jobs is assumed to minimize the sum of setup times. The new jobs can be either scheduled after the old jobs or inserted within the existing schedule, which results in a disruption cost that has to be minimized together with the sum of setup times of the overall schedule. In this paper we tackle several simple setup time configurations yielding different scheduling problems for which we propose optimal polynomial time algorithms or provide $\mathcal{NP}$-hardness proofs. In the former case we consider the problem of enumerating the set of strict Pareto optima for the sum of setup times and disruption cost criteria.

Keywords: Scheduling, New orders, Multicriteria, Exact algorithms, Complexity proofs.

1. INTRODUCTION AND PROBLEM FORMULATION

Consider a short-term scheduling problem for which a set of jobs has been scheduled for execution, but has not been processed yet by the production resources. A new set of jobs enters the shop and has to be scheduled, taking into account the scheduling decisions that have already been taken. This kind of problem is referred to as a *Rescheduling problem for new orders* (Hall and Potts (2004)) and has appeared in the literature only quite recently. This situation may occur in industries for which the short-term schedule is meant for a longer period than the actual real production period. This is the case, for instance, in a shampoo packing system in which a short-term schedule is calculated for a period of 36 hours, knowing that it is daily updated. When doing so, the scheduler has 12 hours of remaining unproduced but scheduled jobs. New jobs, corresponding to 24 hours of production, have to be scheduled taking this remaining schedule into account. Therefore, the aim of the scheduler is to schedule the new jobs such that the objective function is minimized, together with the disruption cost induced by inserting new jobs into the remaining schedule.

The problem tackled in this paper can be formally stated as follows. A single machine is available for processing jobs, from time 0 onwards. Let J_0 be the set of the n_0 jobs already scheduled, the so-called 'old' jobs, and J_N be the set of the n_N new incoming jobs. Each job j is defined by a processing time p_j and belongs to a single family f_k. Switching processing from family k to family ℓ requires a setup time, which we assume to depend on family ℓ only and not on family k: this setup time is denoted by s_ℓ. The objective, for both sets of jobs, is to minimize the sum of setup times which is equivalent to minimize the makespan of the overall schedule, referred to as C_{max}.

The input of the problem consists of the sets of jobs (old and new) and the initial schedule (sequence) α for the jobs in J_0, which is assumed to be optimal for the makespan criterion. We are asked to find a schedule for all jobs. We are allowed to disturb the original schedule α, but this will incur a disruption cost, which is measured either by a change in the scheduling positions of the old jobs, or by a change in the completion times of the old jobs. Let π denote a schedule for the old and new jobs. Then we define $P_j(\pi)$ as the position of job j in schedule π and $C_j(\pi)$ as its completion time in schedule π. We consider in this paper three disruption measures:

- $D_j(\alpha, \pi)$: the absolute positional disruption. For each job $j \in J_0$ we define $D_j(\alpha, \pi)$ as the absolute difference between its position in α and its position in π. We have $D_j(\alpha, \pi) = |P_j(\pi) - P_j(\alpha)|$.
- $P_j(\alpha, \pi)$: the positional disruption. For each job $j \in J_0$ we define $P_j(\alpha, \pi)$ as the difference between its position in α and its position in π. We have $P_j(\alpha, \pi) = P_j(\pi) - P_j(\alpha)$.
- $\Delta_j(\alpha, \pi)$: the absolute completion time disruption. For each job $j \in J_0$ we define $\Delta_j(\alpha, \pi)$ as the absolute difference between its completion time in α and its

completion time in π. We have $\Delta_j(\alpha, \pi) = |C_j(\pi) - C_j(\alpha)|$.

Henceforth, the disruption cost Z can be measured by $D_{max} = \max_{j \in J_0} D_j$, $\sum_j D_j$, $P_{max} = \max_{j \in J_0} P_j$, $\sum_j P_j$, $\Delta_{max} = \max_{j \in J_0} \Delta_j$ or $\sum_j \Delta_j$.
We focus on the calculation of strict Pareto optima for the makespan and a disruption cost Z. A schedule s is a strict Pareto optimum iff there does not exist another schedule s' such that $C_{max}(s') \leq C_{max}(s)$ and $Z(s') \leq Z(s)$ with at least one strict inequality. We say that s is a weak Pareto optimum if, in the above definition, the inequalities are strict inequalities.

Table 1 presents a summary of the notations used throughout this paper.

Table 1. The used notations

Notation	Meaning
n_0	The number of old jobs
J_0	The set of old jobs
α	The initial schedule (sequence) of old jobs
n_N	The number of new jobs
J_N	The set of new jobs
p_j	Processing time of job j
f_k	Set of jobs of family number k
s_k	Setup time changing the production to family k
n_0^f	the number of distinct families in J_0
f_k^0	Family number k restricted to the old jobs
n_N^f	The number of distinct families in J_N
f_k^N	Family number k restricted to the new jobs
$\lvert f_k^0 \rvert$	The number of jobs in family f_k^0
$\lvert f_k^N \rvert$	The number of jobs in family f_k^N
$n_{0 \cap N}^f$	The number of families in J_N that are also in J_0
$k \rightarrow \ell/\alpha$	The families that precede family f_ℓ in schedule α

Research on this topic is related to *rescheduling problems* which have been the matter of numerous studies (see Vieira et al. (2003) for a review). However, the approach consider here is relatively new, and only a few papers on this subject have appeared so far. The first one is by Unal et al. (1997). They consider the single machine problem with sequence dependent family setup times. The aim is to obey deadlines for old jobs and minimize the makespan of the new jobs. They propose a polynomial procedure to check whether there exists a solution in which no additional setups are required, while obeying the deadlines and the order of the old jobs in the initial schedule. They further show that the problem of minimizing total weighted completion time is $\mathcal{NP}$-hard in general, but polynomial when the processing times are unit.
A major contribution is due to Hall and Potts (2004), who proposed a seminal paper in the field. They define the disruption measures D_j and Δ_j, and study several single machine scheduling problems. In each of these problems, the objective function contains a component that is equal to the total completion time or the maximum completion time, which has to be minimized in combination with or subject to a disruption cost, which is modelled by either D_{max}, $\sum_j D_j$, Δ_{max} or $\sum_j \Delta_j$. Among the tackled problems, they consider those for which the scheduling criterion is minimized under the constraint that the disruption cost is bounded by a threshold value ϵ. This is equivalent to compute a weak Pareto optimum for these two criteria. Hall and Potts proposed either polynomial time algorithms or showed $\mathcal{NP}$-hardness. It is important to notice that for almost all of the studied problems, the initial sequence α is assumed to be optimal for the scheduling criterion. Nevertheless, at the end of their paper they briefly discuss the multiple disruptions situation, which means that α is the result of multiple job insertions and is thus no longer optimal for the scheduling criterion. Hall et al. (2007) consider a rescheduling problem with a single machine and multiple disruptions. They proposed a branch-and-bound algorithm which solves instances with up to 1000 jobs in size.
Furthermore, Yuan and Mu (2007) study four single machine rescheduling problems with the assumption that sequence α is optimal for the makespan criterion and that each job has a release date. They consider the problem of minimizing this objective subject to the constraint that the disruption criterion is bounded by a value ϵ. The two problems for which the disruption criterion is either Δ_{max} or $\sum_j \Delta_j$ are shown to be $\mathcal{NP}$-hard. This is not the case of the problem with D_{max} criterion which can be solved in polynomial time.

In this paper we provide results and algorithms for various single machine scheduling problems. We will use the classical three-field notation scheme introduced by Graham et al. (1979), where we use the notation introduced by T'kindt and Billaut (2006) to denote bicriteria problems. In Section 2 we focus on the $1|s_f = s|P_{max}, C_{max}$ problem. The setup times between jobs from different families are all equal and the objective is to find the set of Pareto optimal points for the objectives of minimizing the makespan and minimizing the disruption cost, which is computed as the maximum of the relative positional disruptions. We propose a polynomial time algorithm for enumerating the set of strict Pareto optima for these two criteria. The problem remains polynomially solvable even in the case of the maximum absolute completion time disruption. This problem, referred to as $1|s_f = s|\Delta_{max}, C_{max}$, is tackled in Section 3. We next consider in Section 4 the $1|s_f|P_{max}, C_{max}$ problem with family dependent, but not sequence dependent, setup times for which we propose a polynomial time dynamic programming algorithm for enumerating the set of strict Pareto optima. In Section 5 we show that, for the Δ_{max} criterion, this enumeration cannot be achieved in polynomial time since the $1|s_f|\Delta_{max}, C_{max}$ problem is shown to be $\mathcal{NP}$-hard in the weak sense. We show, in Section 6, that this result also holds for the $1|s_f| \sum_j \Delta_j, C_{max}$ problem. Table 2 summarizes the tackled enumeration problems and the corresponding complexity.

For conciseness purposes, all proofs of the proposed results are omitted.

2. THE $1|S_F = S|P_{MAX}, C_{MAX}$ PROBLEM

We assume in this section that the setup times are family independent and that the disruption criterion is P_{max}. The goal is to enumerate the set of strict Pareto optima for the C_{max} and P_{max} criteria. Before giving the main solution