Cost Minimizing Scheduling of Work and Rework Processes on a Single Facility under Deterioration of Reworkables

Karl Inderfurth^a, Mikhail Y. Kovalyov^b, C.T. Daniel Ng^c, Frank Werner^a

^{a}*Otto-von-Guericke-Universität, Magdeburg, Germany, E-mail:*
inderfurth@ww.uni-magdeburg.de, frank.werner@mathematik.uni-magdeburg.de

^{b}*Faculty of Economics, Belarus State University, Minsk, Belarus, E-mail:*
koval@newman.bas-net.by

^{c}*Department of Logistics, The Hong Kong Polytechnic University*
Hung Hom, Kowloon, Hong Kong, E-mail: lgtctng@polyu.edu.hk

Abstract

The problem of scheduling the production of new and recovering defective items of the same product manufactured on the same facility is studied. The items are produced in batches. The processing of a batch includes two stages. In the first stage, all items of a batch are manufactured and good quality items go to the inventory to satisfy given demands. In the second stage, defective items of the same batch are reworked. After rework each item has the required good quality. During waiting for rework defective items are assumed to deteriorate. This results in an increase in time and cost for performing rework processes. It is assumed that the percentage of defective items is the same in each batch, and that they are uniformly distributed in each batch. A setup time as well as a setup cost is required to start batch processing and to switch from production to rework. The objective is to find batch sizes such that all demands are satisfied and total setup, rework and inventory holding cost is minimized. Polynomial time algorithms are presented to solve two realistic special cases of this problem. A generalization to a multiple product case is discussed.

Key Words: inventory control, lot-sizing, rework, deterioration.

1 Introduction

In many manufacturing processes, some of the fabricated products can be defective due to an unstable production environment, non-perfect technology or human mistakes. Instead of being disposed of, defective items are more and more put into recovery processes in order to regain material and value added. These reuse activities are also supported by a growing environmental consciousness. Recovery actions belong to the broad field of Reverse Logistics which deals with all kinds of reuse processes in supply chains (see de Brito and Dekker [1]). As a major activity in this context we face rework which aims to recover defective products in such a way that they meet the quality level of a good item. As described in Flapper and Jensen [4] and Inderfurth and Teunter [7], integrating rework and manufacturing processes leads to challenging planning and control problems, especially if both processes are using the same equipment.

In some instances reworkable items undergo a deterioration process while they wait to be recovered. This effect is especially found in process industries as indicated in Flapper et al. [2]. If deterioration appears as continuous worsening of the state of work-in-process it can result in an increase in processing time as well as in processing cost for reworking the respective items. This can complicate planning and control activities considerably. A tractable way of modelling such deteriorating processes is to assume that rework time and cost are linearly increasing with the time span defective items are waiting to be reworked. In this paper, we address the problem of scheduling work and rework processes under the above described conditions. Due to the presence of setup costs and setup times we rely on a policy where both manufacturing and rework is performed in batches.

In the sequel we consider a situation, in which items of a single product have to be manufactured by a single facility. Deadlines d_j are given such that j good quality items should be manufactured by d_j, $j = 1, \ldots, N$. Values $d_1, \ldots, d_N$ can be obtained from demand planning. If $d_i < d_{i+1} = d_{i+2} = \cdots = d_{i+k}$, then deadline d_{i+k} can be viewed as a delivery date for k items. Therefore, the considered deadlines can be used to model demands for given numbers of good quality items by given delivery dates.

Product items are manufactured in batches and a setup time s_1 precedes the processing of each batch. In each batch, some of the produced items can be defective. We assume that the

percentage of the defective items is known and that they are uniformly distributed in each batch. Furthermore, we make a simplifying but still realistic assumption that in each batch the number of good quality items is a multiple of the number of defective items. Thus, we assume that each batch consists of an integer number of blocks where each block includes some number of good quality items followed by a single defective item. Let n be the total number of blocks.

When manufacturing of all items in the batch has been completed, the same facility is used for reworking all defective items belonging to the manufacturing batch. The start of processing the rework batch is preceded by a setup time s_2. After reprocessing, defective items are in the state of good quality items. Thus, the entire production process includes two stages that we call *work* and *rework*. Note that we consider a work/rework coordination policy where a single manufacturing batch is always followed by a single rework batch. This one-to-one policy is easy to implement and well-known from literature (see Lindner et al. [8] and Teunter and Flapper [12]).

A batch setup cost is associated with each batch. Furthermore, there is a holding cost for each time unit of a good-quality item to be in inventory and a holding cost for each time unit of a defective item to await rework. Also reprocessing costs of defective items have to be considered since it is assumed that due to deterioration the per-item rework cost is not constant, but increases linearly with the time an item waits for reprocessing. The problem is to find a sequence of batch sizes such that all demands are met in time and the total setup, rework and holding cost is minimized.

To guarantee high utilization of the production facility processing is performed such that the equipment does not stand idle from the beginning of the production until the completion of the last item. Furthermore, it has to be taken into account that caused by deterioration the reprocessing time of a defective item is increasing proportionally to the time it stays in the reworkables inventory.

There is a large body of literature dealing with production and inventory control problems in case of product deterioration (see Goyal and Giri [5] for a recent survey). We also find a great amount of publications dealing with planning and control in case of rework (see Flapper and Jensen [4] for an overview). There are, however, very few scientific contributions addressing the problem area which combines both cases like it is done in our paper. A

problem similar to that formulated above is treated by Teunter and Flapper [12] and Flapper and Teunter [3]. They consider a situation where occurrence of a defective item is random and where demand is not restrictive insofar as every good-quality item can be sold immediately. Inderfurth et al. [6] address a continuous version of the above problem where the demand rate is assumed to be constant and rework time and rework cost increase linearly with the manufacturing batch size. As far as we know, there are no results available for the particular problem studied in this paper.

In the following section, we reformulate the above problem using a traditional scheduling terminology. In Section 3, we discuss the possibility of solving the problem by a dynamic programming technique. We show that the traditional approach may lead to an exponential algorithm. As we are not aware of a specific efficient approach to solve the problem, we consider two realistic relaxations that admit efficient solution procedures. One relaxation, denoted as problem WR1, is to determine a single batch such that an average cost calculated as the ratio of the total cost to the length of the batch is minimized. This results in a policy that should be used if we face an unlimited stationary demand process. In Section 4, we show that problem WR1 can be solved in $O(N + n^2)$ time. The other relaxation is to assume that the total processing time to rework defective items in the same batch is a linear function of the batch size. This relaxation approximates the individual waiting time impact of deterioration by considering an average time behavior as being valid for each item. In Section 5, a dynamic programming algorithm is presented to solve the corresponding relaxed problem, denoted as WR2. The algorithm runs in $O(n^3 + nN)$ time. In Section 6, we consider a multi-product case and describe the ideas of the algorithms for corresponding generalizations of problems WR1 and WR2. The algorithms are polynomial if the number of products is a constant. The paper concludes with a summary of the results and suggestions for future research.

2   Formal problem description

The problem of scheduling work and rework processes discussed in the introduction can be formulated as follows. There are N jobs to be scheduled for processing on a single machine. Given a schedule, let us consider the corresponding sequence of job completion times. We say that a job is in *position j* if its completion time is jth in the above sequence. All jobs are available for processing at time zero. *Positional deadlines* $d_1, d_2, \ldots, d_N$ are given such that

a job in position j should be completed by d_j, $j = 1, \ldots, N$. We can assume without loss of generality that $d_1 \leq \cdots \leq d_N$. Jobs are processed in batches. Machine setup time s_1 precedes the processing of each batch. We assume that the machine does not stand idle from time zero until the completion of all jobs.

Jobs of each batch are classified into two types of *quality* jobs and *defective* jobs, which are processed in *blocks*. Each block consists of $v - 1$, $v \geq 2$, quality jobs processed consecutively and followed by a single defective job. Therefore, we assume that N is a multiple of v: $N = nv$, where n is the total number of blocks.

Each batch can include an arbitrary number of blocks. We call this number as *size* of the corresponding batch. Thus, if a batch includes b blocks, then it consists of bv jobs, among which there are b defective jobs. The processing of a quality job consists of a single unit-time *work operation*. The processing of a defective job consists of two operations called work and *rework operations*. We assume that each operation of any job completes immediately when its processing has been finished. Work operation of a defective job requires one unit of time. The processing time of a rework operation is equal to $p + a \cdot t$, where p and a are non-negative numbers and t is the difference between the starting time of the rework operation and the completion time of the corresponding work operation of the same job.

Given a batch, all rework operations are processed last and are preceded by a machine setup time s_2. An example of a batch structure is given in Fig. 1.
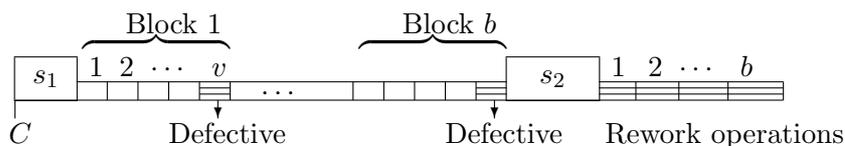

Figure 1: A batch including b blocks and starting at time C

Since all jobs are identical and machine idle times are not allowed, it is easy to see that a schedule is completely characterized by a partition of the set of jobs into batches and the sequence of rework operations in each batch.

The following costs are considered:

α – batch setup cost,

β – finished job holding cost (earliness cost),

γ – defective job holding and rework cost.

Given a schedule, denote by C_j, S_j^R and C_j^W the completion time of a job in position j, the starting time of rework operation and the completion time of work operation of the same job in position j, respectively.

Let S be a schedule with k batches and set Z of defective jobs. The total cost of such a schedule is calculated as follows:

$$F(S) = \alpha k + \beta \sum_{j=1}^{N}(d_j - C_j) + \gamma \sum_{j \in Z}(S_j^R - C_j^W).$$

The problem is to find such a schedule that the deadlines are satisfied, that is, $C_j \leq d_j$, $j = 1, \ldots, N$, and $F(S)$ is minimized. We denote this problem as WR.

The following lemma determines an optimal sequence of rework operations in each batch for problem WR.

Lemma 1 *There exists an optimal schedule for problem WR, in which rework operations of defective jobs in the same batch are executed in the reversed order of the corresponding work operations, that is, the "Last Come First Served" (LCFS) rule is optimal for sequencing rework operations of defective jobs in the same batch.*

Proof. Consider a batch in an optimal schedule. Observe that changing the sequence of rework operations within the same batch - via influencing the rework processing time - affects only defective job holding and rework cost. This cost is minimized when the total unfinished defective job holding cost is minimum. It is well known in scheduling literature that the LCFS rule, which is equivalent to the "Shortest Processing Time" (SPT) rule of Smith [11], minimizes the latter time. ∎

From now on, we consider only schedules satisfying Lemma 1.

3 A discussion of a dynamic programming algorithm and modifications of the model

Dynamic programming is a mathematical tool that is commonly used to solve problems of optimal inventory control.

A dynamic programming algorithm for solving problem WR can be proposed, in each iteration of which a batch is added to the end of a partial schedule. In this algorithm, contribution of the added batch to the objective function has to be computed.

Consider a partial schedule S in which r blocks of jobs have been assigned for processing. Assume that the last job in position rv completes at time C. Let us assign a batch of size b to the end of this schedule and calculate contribution of this batch to the objective function of problem WR.

Contribution to the batch setup cost is α. Contribution to the total finished job holding time consists of contribution of quality jobs, denoted as F_q, and contribution of defective jobs, denoted as F_d. We have

$$F_q = \sum_{j=1}^{b(v-1)}[d_{rv+j}-(C+s_1+j)]-(v-1)\sum_{i=1}^{b-1}i = \sum_{j=1}^{b(v-1)}d_{rv+j}-b(v-1)(C+s_1)-b^2\frac{v(v-1)}{2}. \quad (1)$$

In order to find F_d, we first calculate holding times and processing times of rework operations for unfinished defective jobs. Let h_j and $p_j = p + ah_j$ denote holding time and processing time of the rework operation, respectively, for unfinished defective job, whose rework operation starts as jth in the considered batch, $j = 1, \ldots, b$. Taking into account the policy "Last Come First Served", we have

$$h_1 = s_2, \ p_1 = p + ah_1,$$

$$p_j = p + ah_j, \ h_j = v + h_{j-1} + p_{j-1} = v + p + (a+1)h_{j-1}, \ j = 2, \ldots, b,$$

from where we deduce that

$$h_j = (v+p)\frac{(a+1)^{j-1}-1}{a} + (a+1)^{j-1}s_2, \ p_j = (v+p+as_2)(a+1)^{j-1} - v, \ j = 2, \ldots, b.$$

The completion time of the defective job completed as jth among the defective jobs is equal to $C + s_1 + s_2 + vb + \sum_{i=1}^{j}p_i$. Let us define $P(j) = \sum_{i=1}^{j}p_i$. Calculate

$$P(j) = (v+p+as_2)\sum_{i=1}^{j}(a+1)^{i-1} - jv = (v+p+as_2)\frac{(a+1)^j-1}{a} - jv, \ j = 1, \ldots, b. \quad (2)$$

Then

$$F_d = \sum_{j=1}^{b}[d_{rv+b(v-1)+j} - (C + s_1 + s_2 + vb + P(j))]$$

and, on substitution of $P(j), \ j = 1, \ldots, b,$

$$F_d = \sum_{j=1}^{b}d_{rv+b(v-1)+j} + \frac{(v+p+as_2)b}{a} + \frac{bv}{2} - b(C+s_1+s_2) - \frac{vb^2}{2} - \frac{(v+p+as_2)((a+1)^b-1)(a+1)}{a^2}. \quad (3)$$

Contribution to the total unfinished defective job holding time, denoted as H, is equal to

$$H = \sum_{j=1}^{b}h_j = (a+1)^b\left(\frac{v+p}{a^2} + \frac{s_2}{a}\right) - \frac{v+p}{a^2} - \frac{s_2+b(v+p)}{a}. \quad (4)$$

7

Finally, contribution of the added batch of size b to the objective function is equal to

$$\alpha + \beta(F_q + F_d) + \gamma H.$$

¿From formula (2), we note that the total processing time of rework operations in a batch of size b, $P(b)$, depends non-linearly on b. Therefore, the length of a schedule can be calculated if we know all its batch sizes. The suggested dynamic programming algorithm uses the length of the schedule to calculate contribution of the added batch to the objective function. Therefore, it should enumerate all possible batch sizes, which makes it exponential in the number of batches.

The above discussion demonstrates that we are unable to provide an efficient algorithm for solving problem WR in its original setting unless the number of batches is bounded from above by a constant. Let us modify problem WR so that it remains realistic and admits an efficient solution algorithm.

We consider the following two modifications of problem WR.

Problem WR1. Let $f(b)$ denote contribution of the first batch including b blocks to the objective function of problem WR. Let $T(b)$ denote the total setup and processing time of this batch. In problem WR1, the size of a single batch should be determined such that the *average cost* $f(b)/T(b)$ is minimized, subject to meeting deadlines for the first vb jobs.

Such an approach to modelling situations in planning production with reworkable defectives is known in the literature, see for example, Liu and Yang [9] and Teunter and Flapper [12]. It represents some kind of myopic policy applied to the dynamic planning problem. In case of a stationary demand process and an unlimited planning horizon this policy is expected to be optimal.

Problem WR2. Let $R(b)$ denote the total processing time of rework operations in a batch containing b blocks. We assume that $R(b)$ is a linear function of b. This assumption is motivated by the fact that the average waiting time of reworkable items is proportional to their average stock which itself is proportional to the batch size defined by b. Thus, the individual behavior of deteriorating defective items is replaced by an average behavior. This approach is also used in Inderfurth et al. [6]. Let $R(b) = A + Bb$. To minimize deviation from the original problem WR, we assume that the processing time of a rework operation that starts as jth among rework operations in the batch of size b is equal to $p'_j = p + j\delta(b)$,

$j = 1, \ldots, b$, where $\delta(b)$ can be found from $\sum_{j=1}^{b} p'_j = A + Bb$. We have

$$\delta(b) = \frac{2A + 2b(B - p)}{b(b + 1)}.$$

All other assumptions remain the same as in problem WR.

An important special case of problem WR2 appears when processing times of rework operations are all equal. If they are all equal to p, then we can set $A = 0$, $B = p$ and $\delta(b) = 0$ for any b in problem WR2.

4   An analytical solution to problem WR1

For problem WR1, formulas (1)-(4) can be used to calculate $f(b)$ and $T(b)$. Since $C = 0$ and $rv = 0$, we have

$$f(b) = \alpha + \beta \left[\sum_{j=1}^{bv} d_j + \frac{(v + p + as_2)b}{a} + \frac{bv}{2} - b(vs_1 + s_2) - \frac{b^2 v^2}{2} - \right.$$

$$\left. \frac{(v + p + as_2)((a + 1)^b - 1)(a + 1)}{a^2} \right] + \gamma \left[(a + 1)^b (\frac{v + p}{a^2} + \frac{s_2}{a}) - \frac{v + p}{a^2} - \frac{s_2 + b(v + p)}{a} \right],$$

$$T(b) = s_1 + s_2 + bv + P(b) = s_1 + s_2 + (v + p + as_2)\frac{(a + 1)^b - 1}{a}.$$

The set of batch sizes b feasible with respect to the deadlines, denoted as X, can be determined as follows. For quality jobs, the following inequalities should be satisfied.

$$s_1 + j \le d_{j-i}, \ \ iv + 1 \le j \le (i + 1)v - 1, \ \ i = 0, 1, \ldots, b - 1. \tag{5}$$

They are equivalent to the following.

$$s_1 \le \min\{d_{j-i} - j \mid iv + 1 \le j \le (i + 1)v - 1, \ i = 0, 1, \ldots, b - 1\} := V_b.$$

Set $L_b = \min\{d_{j-i} - j \mid bv + 1 \le j \le (b + 1)v - 1\}$. Values L_b, $b = 0, 1, \ldots, n - 1$, can be computed in $O(N)$ time. Furthermore, we have

$$V_1 = L_0 \qquad \text{and} \qquad V_b = \min\{V_{b-1}, L_{b-1}\}, \ \ b = 2, \ldots, n.$$

Therefore, values V_b, $b = 1, \ldots, n$, and consequently, values b satisfying (5) can be computed in $O(N)$ time.

Using definition of $P(j)$, we conclude that for defective jobs the following inequalities should be satisfied in order to provide the feasibility of batch size b.

$$s_1 + s_2 + (b - j)v + (v + p + as_2)\frac{(a + 1)^j - 1}{a} \leq d_{b(v-1)+j}, \ j = 1, \ldots, b. \tag{6}$$

These inequalities can be verified in $O(b)$ time and, hence, in $O(n^2)$ time for all $b = 1, \ldots, n$.

We include b, $b \in \{1, \ldots, n\}$, into the set X, if inequalities (5) and (6) are satisfied. We have shown that this set can be constructed in $O(N + n^2)$ time.

Optimal batch size b^* can be found from the equation

$$b^* = \arg\min\left\{\frac{f(b)}{T(b)} \mid b \in X\right\}.$$

Thus, problem WR1 can be solved in $O(N + n^2)$ time.

5 Dynamic programming algorithm for problem WR2

Recall that in problem WR2 the processing time of a rework operation that starts jth among rework operations in the batch of size b is equal to $p'_j = p + j\delta(b)$ and the total processing time of rework operations in the batch containing b blocks is equal to $R(b) = A + Bb$.

Consider partial feasible schedules that include k batches and r blocks. All these schedules complete at the same time

$$C(k, r) = (s_1 + s_2 + A)k + (B + v)r. \tag{7}$$

Therefore, a partial feasible schedule with minimal objective function value among all such schedules is *dominant* such that it can be extended to a complete feasible schedule with the smallest objective function value among all complete feasible schedules extended from a partial feasible schedule with k batches and r blocks.

The above discussion justifies the following dynamic programming algorithm for solving problem WR2. In this algorithm, denoted as DP, the number of batches k and the number of blocks r are *state variables*. We recursively compute value $F(k, r)$, which is the minimum objective function value for feasible schedules in the *state* (k, r).

If a batch with size b is added to the end of a schedule in state (k, r), then its contribution to the objective function, denoted as $\Delta(k, r, b)$, can be calculated as follows. Contribution to the batch setup cost is α. Similar to the discussion in Section 3, contribution to the finished

job holding time consists of contribution of quality jobs, denoted as T_q, and contribution of defective jobs, denoted as T_d. We have

$$T_q = \sum_{j=1}^{b(v-1)} d_{rv+j} - b(v-1)(C(k,r) + s_1) - b^2 \frac{v(v-1)}{2}.\qquad(8)$$

As in Section 3, let h_j denote holding time of the unfinished defective job, whose rework operation starts as jth in the considered batch, $j = 1, \ldots, b$. We have

$$h_1 = s_2, \ h_j = v + h_{j-1} + p'_{j-1} = v + p + (j-1)\delta(b) + h_{j-1}, \ j = 2, \ldots, b,$$

and inductively,

$$h_j = (v+p)(j-1) + s_2 + \delta(b) \sum_{i=1}^{j-1} i = (v+p)(j-1) + s_2 + \delta(b) \frac{j(j-1)}{2}.$$

The completion time of the defective job completed as jth among the defective jobs of the considered batch is equal to $C(k,r) + s_1 + s_2 + vb + \sum_{i=1}^{j} p'_i$. Let us define $R(j) = \sum_{i=1}^{j} p'_i$. Calculate

$$R(j) = jp + \delta(b) \sum_{i=1}^{j} i = jp + \delta(b) \frac{j(j+1)}{2}$$

and

$$T_d = \sum_{j=1}^{b} [d_{rv+b(v-1)+j} - (C(k,r) + s_1 + s_2 + vb + R(j))].$$

Using formula $\sum_{i=1}^{k} i^2 = \frac{k(k+1)(2k+1)}{6}$, we obtain

$$T_d = \sum_{j=1}^{b} d_{rv+b(v-1)+j} - \left[b(C(k,r)+s_1+s_2+vb) + p\frac{b(b+1)}{2} + \delta(b) \left(\frac{b(b+1)}{4} + \frac{b(b+1)(2b+1)}{12} \right) \right].\qquad(9)$$

Contribution to the total unfinished defective job holding time, denoted as U, is equal to

$$U = \sum_{j=1}^{b} h_j = (v+p) \frac{(b-1)b}{2} + bs_2 + \delta(b) \left(\frac{b(b+1)(2b+1)}{12} - \frac{b(b+1)}{4} \right).\qquad(10)$$

Combining equations (8), (9) and (10), we obtain

$$\Delta(k,r,b) = \alpha + \beta(T_q + T_d) + \gamma U = \alpha + \beta \left(\sum_{j=1}^{bv} d_{rv+j} - \left[bv(C(k,r)+s_1) + b^2 \frac{v(v-1)}{2} + \right. \right.$$

$$\left. \left. b(s_2 + bv) + p\frac{b(b+1)}{2} + \delta(b) \left(\frac{b(b+1)}{4} + \frac{b(b+1)(2b+1)}{12} \right) \right] \right) + \qquad(11)$$

$$\gamma \left[(v+p) \frac{(b-1)b}{2} + bs_2 + \delta(b) \left(\frac{b(b+1)(2b+1)}{12} - \frac{b(b+1)}{4} \right) \right].$$

11

Feasibility of the added batch of size b can be verified as follows. For quality jobs, the following inequalities should be satisfied.

$$C(k,r) + s_1 + j \leq d_{rv+j-i}, \ iv + 1 \leq j \leq (i+1)v - 1, \ i = 0, 1, \ldots, b-1. \tag{12}$$

Using definition of $R(j)$, for defective jobs the following inequalities should be satisfied.

$$C(k,r) + s_1 + s_2 + bv + jp + \delta(b)\frac{j(j+1)}{2} \leq d_{rv+b(v-1)+j}, \ j = 1, \ldots, b. \tag{13}$$

Now we are ready to give a formal description of algorithm DP.

Algorithm DP

Step 1 (Initialization) Set $F(k,r) = 0$ for $(k,r) = (0,0)$ and $F(k,r) = \infty$ for $(k,r) \neq (0,0)$, $k = 1, \ldots, n$, $r = k, k+1, \ldots, n$. Set $k = 0$.

Step 2 (Recursion) For $\rho = k+1, \ldots, n$, compute the following:

$$F(k+1, \rho) = \min_{b=1,\ldots,\rho, \ r=\rho-b} \begin{cases} F(k,r) + \Delta(k,r,b), & \text{if (12) and (13) are satisfied,} \\ \infty, & \text{otherwise,} \end{cases}$$

where $\Delta(k,r,b)$ is computed according to (11), and $C(k,r)$ used in (11), (12) and (13) is computed according to (7).

If $k \leq n - 2$, then set $k = k + 1$ and repeat Step 2. Otherwise, go to Step 3.

Step 3 (Optimal solution) Calculate optimal solution value

$$F^* = \min\{F(k,n) \mid k = 1, \ldots, n\}.$$

If $F^* = \infty$, then there is no schedule feasible with respect to the deadlines $d_1, \ldots, d_N$.

If $F^* < \infty$, then corresponding optimal batch schedule can be found by backtracking.

∎

It is easy to see that the time complexity of algorithm DP can be estimated as $O(n^2 Q)$, where Q is the time complexity of computing the minimum in the recursive equation. The latter computation reduces to calculating $\Delta(k,r,b)$ and verifying (12) and (13) for given k, $b = 1, \ldots, \rho$, and $r = \rho - b$.

Since $C(k,r)$ can be computed in a constant time for given k and r, the time complexity estimation for calculating $\Delta(k,r,b)$ is determined by the time of calculating $\sum_{j=1}^{bv} d_{rv+j}$. Let us introduce *aggregate deadlines* $d'_i = \sum_{j=1}^{v} d_{(i-1)v+j}$, $i = 1, \ldots, n$. It is easy to see that all aggregate deadlines can be computed in $O(N)$ time. Observe that $\sum_{j=1}^{bv} d_{rv+j} = \sum_{i=r+1}^{r+b} d'_i$. Now, let us introduce *twice aggregate deadlines* $d''_{r,q} = \sum_{i=r}^{q} d'_i$, $q = 1, \ldots, n$, $r = 1, \ldots, q$. All twice aggregate deadlines can be computed in $O(n^2)$ time. We have $\sum_{i=r+1}^{r+b} d'_i = d''_{r+1,r+b}$. Therefore, values $\Delta(k,r,b)$, $b = 1, \ldots, \rho$, $r = \rho - b$, can be computed in $O(\rho)$ time if twice aggregate deadlines are given.

Now consider inequalities (12) and (13). Inequality (12) is equivalent to

$$C(k,r) + s_1 \le \min\{d_{rv+j-i} - j \mid iv + 1 \le j \le (i+1)v - 1, \ i = 0,1,\ldots,b-1\}.$$

Denote $M_{r,i} = \min\{d_{(r+i)v+j-i} - j \mid 1 \le j \le v - 1\}$. Values $M_{r,i}$, $r = 1, \ldots, n-1$, $i = 0,1,\ldots,n-r-1$, can be computed in $O(nN)$ time. Note that inequality (12) is equivalent to

$$C(k,s) + s_1 \le \min\{M_{r,i} \mid i = 0,1,\ldots,b-1\}.$$

Having values $M_{r,i}$, $i = 0,1,\ldots,\rho$, inequality (12) can be verified for all $b = 1, \ldots, \rho$ in $O(\rho)$ time.

Similarly, inequality (13) is equivalent to

$$C(k,r) + s_1 + s_2 \le \min\left\{d_{rv+b(v-1)+j} - \left(bv + jp + \delta(b)\frac{j(j+1)}{2}\right) \mid j = 1, \ldots, b\right\}.$$

Denote $K_{r,j,b} = d_{rv+b(v-1)+j} - \left(bv + jp + \delta(b)\frac{j(j+1)}{2}\right)$. Values $K_{r,j,b}$, $r = 1, \ldots, n$, $b = 1, \ldots, n - r$, $j = 1, \ldots, b$, can be computed in $O(n^3)$ time. Denote $K^*_{r,b} = \min\{K_{r,j,b} \mid j = 1, \ldots, b\}$. Values $K^*_{r,b}$, $r = 1, \ldots, n$, $b = 1, \ldots, n-r$, can be computed in $O(n^3)$ time as well. Inequality (13) can be written as $C(k,s) + s_1 + s_2 \le K^*_{r,b}$. Having values $K^*_{r,b}$, $b = 1, \ldots, \rho$, inequality (13) can be verified for all $b = 1, \ldots, \rho$ in $O(\rho)$ time.

¿From the above discussion, we conclude that the time complexity of algorithm DP is $O(n^3)$ provided that twice aggregate deadlines and values $M_{r,i}$, $i = 0,1,\ldots,n-r$, and $K^*_{r,b}$, $b = 1, \ldots, n-r$, $r = 1, \ldots, n$, are calculated. Therefore, problem WR2 can be solved in $O(n^3 + nN)$ time.

6 A multi-product case

In the discussion given below, jobs, groups and a machine can be treated as items, products and a facility, respectively.

Assume that there are jobs of G, $G \geq 2$, groups to be scheduled on a single machine. Positional deadlines $d_{j,g}$ are given such that j jobs of group g should be completed by $d_{j,g}$, $j = 1, \ldots, N_g$, $g = 1, \ldots, G$, where N_g is the number of jobs of group g.

As before, jobs are processed in batches. Only jobs of the same group can form a batch. A batch of group g consists of blocks, each of which includes $v_g - 1$ quality jobs processed consecutively and followed by a single defective job. We assume that N_g is a multiple of v_g: $N_g = n_g v_g$, where n_g is the total number of blocks of group g. A setup time s_1^g precedes the processing of each batch of product g.

The processing of a quality job of group g consists of a single work operation with processing time u_g. The processing of a defective job of group g consists of work and rework operations with processing times u_g and $p_g + a_g \cdot t$, respectively, where t is the difference between the starting time of the rework operation and the completion time of the corresponding work operation of the same job.

Given a batch of group g, all its rework operations are processed last and are preceded by a machine setup time s_2^g.

Machine idle times are not allowed. Therefore, since all jobs of the same group are identical, a schedule is completely characterized by partitions of the groups into batches, the sequence of all batches and the sequence of rework operations in each batch.

The following costs are associated with each group g:

α_g – batch setup cost,

β_g – finished job holding cost,

γ_g – defective job holding and rework cost.

Given a schedule S, consider a job sequenced jth in group g. Denote by $C_{j,g}$, $S_{j,g}^R$ and $C_{j,g}^W$ its completion time, the starting time of its rework operation and the completion time of its work operation, respectively. Furthermore, denote by k_g and Z_g be the number of batches and the set of defective jobs of group g, respectively. The total cost of such a schedule is

calculated as follows:

$$F(S) = \sum_{g=1}^{G} \left(\alpha_g k_g + \beta_g \sum_{j=1}^{N_g} (d_{j,g} - C_{j,g}) + \gamma_g \sum_{j \in Z_g} (S_{j,g}^{R} - C_{j,g}^{W}) \right). \qquad (14)$$

The problem is to find such a schedule that the deadlines are satisfied, that is, $C_{j,g} \le d_{j,g}$, $j = 1, \ldots, N_g$, $g = 1, \ldots, G$, and $F(S)$ is minimized. We denote this problem as Multi-WR.

As for problem WR, there exists an optimal schedule for problem Multi-WR, in which rework operations of defective jobs in the same batch are executed in the reversed order of the corresponding work operations.

Statement 1 *Problem Multi-WR is NP-hard in the strong sense, even if all setup times and setup costs are equal to zero, each group consists of a single job, there are no defective jobs and there are two distinct holding costs.*

Proof. We use a reduction from the following problem 3-PARTITION which is NP-complete in the strong sense.

3-PARTITION: Given $3m + 1$ positive integers $e_1, \ldots, e_{3m}$ and E such that $\sum_{i=1}^{3m} e_i = mE$ and $E/4 < e_i < E/2$ for $i = 1, \ldots, 3m$, is there a partition of the set $\{1, \ldots, 3m\}$ into m disjoint subsets $X_1, \ldots, X_m$ such that $\sum_{i \in X_j} e_i = E$ for $j = 1, \ldots, m$?

Given an instance of 3-PARTITION, we construct the following instance of problem Multi-WR. There are $G = 4m - 1$ groups each including a single job. Among them there are $3m$ *partition* groups $1, \ldots, 3m$, and $m - 1$ *enforcer* groups. All setup times and setup costs are equal to zero and there are no defective jobs. The holding cost of any job of an enforcer group is equal to one and the holding cost of any job of a partition group is equal to zero. Jobs of partition groups have the same deadline $d = mE + m - 1$. The processing time of a job of partition group i is equal to e_i. A job of enforcer group $3m + j$ has the deadline $j(E + 1)$ and a unit processing time.

The problem is to determine whether there exists a schedule with a total cost not greater than zero. It is easy to see that in such a schedule, a job of enforcer group $3m + j$ must be scheduled in the interval $[j(E + 1) - 1, j(E + 1)]$, $j = 1, \ldots, m - 1$. Consequently, jobs of the partition groups must be scheduled in the intervals $I_1 = [0, E]$, $I_2 = [E + 1, 2E + 1], \ldots,$ $I_m = [(m - 1)E + m - 1, mE + m - 1]$. Denote the set of partition groups scheduled in the

15

interval I_j as X_j, $j = 1, \ldots, m$. The total cost of the schedule does not exceed zero if and only if $\sum_{i \in X_j} e_i = E$ for $j = 1, \ldots, m$, as required. ∎

We now consider the following modifications of problem Multi-WR.

Problem Multi-WR1. Assume that a single batch of each group should be formed and this batch may include any number of jobs of the corresponding group. In this case, a schedule is characterized by the sequence $\pi = (g_1, g_2, \ldots, g_G)$ of group indices, where the only batch of group g_i is sequenced ith, and a set of the corresponding batch sizes $Y = \{b_1, \ldots, b_G\}$. Let $F(\pi, Y)$ denote the total cost of the schedule determined by π and Y, which is calculated according to (14). Let $T(\pi, Y)$ denote the total setup and processing time of this schedule. In problem Multi-WR1, π and Y should be determined such that the average cost $F(\pi, Y)/T(\pi, Y)$ is minimized, subject to meeting deadlines for the jobs in the corresponding schedule.

Problem Multi-WR2. Let $R_g(b)$ denote the total processing time of rework operations in a batch of group g containing b blocks. For each group g, we assume that $R_g(b) = A_g + B_g b$ and the processing time of a rework operation that starts as jth among rework operations in the batch of size b is equal to $p'_{j,g} = p_g + j\delta_g(b)$, $j = 1, \ldots, b$, where $\delta_g(b)$ can be found from $\sum_{j=1}^{b} p'_{j,g} = A_g + B_g b$. All other assumptions remain the same as in problem Multi-WR.

Consider problem Multi-WR1. Similar to Section 4, for a fixed group sequence π and a set Y of fixed batch sizes, we can calculate $F(\pi, Y)$ and $T(\pi, Y)$ and verify whether the pair (π, Y) is feasible with respect to the deadlines in $O(n_\Sigma^2 + N_\Sigma)$ time, where $n_\Sigma = \sum_{g=1}^{G} n_g$ and $N_\Sigma = \sum_{g=1}^{G} N_g$. We do not give the corresponding formulas because they are rather large. Since the number of different pairs (π, Y) is $O(\prod_{g=1}^{G} gN_g)$, problem Multi-WR1 can be solved in $O((n_\Sigma^2 + N_\Sigma) \prod_{g=1}^{G} gN_g)$ time, which is polynomial if G is a constant.

A dynamic programming algorithm can be derived to solve problem Multi-WR2. As in algorithm DP from Section 5, a new partial schedule is obtained by adding a batch to the end of a current partial schedule. We associate G pairs (k_g, r_g), $g = 1, \ldots, G$, of state variables with each partial schedule, where k_g and r_g are the number of batches and the number of blocks of group g in this schedule. Similar to algorithm DP, we recursively compute value $F(k_1, \ldots, k_G, r_1, \ldots, r_G)$, which is the minimum objective function value for feasible schedules in the corresponding state. Since the description of this algorithm will contain many technical

details and huge formulas, we do not give its detailed description. We only note that the algorithm can be implemented in a polynomial time if G is a constant.

7   Conclusions

The problem of scheduling manufacturing and rework processes on a single facility has been studied. In this problem, items of the same product are manufactured in batches and a setup time precedes the processing of each batch. In each batch, some of the produced items can be defective. When manufacturing of all items in the batch has been completed, a rework process for the defective items starts on the same facility, which is preceded by another setup time. Reflecting the presence of deterioration of reworkables, time and cost to remanufacture a defective item is an increasing function of the time this item awaits reprocessing. Recovered defective items meet the standards of good-quality items. The problem is to find a sequence of batch sizes such that the demands for the items are met and the total setup, rework and holding cost is minimized.

This situation has been modelled as a problem of scheduling N jobs, subject to the given positional deadlines. A possibility of solving this problem by a dynamic programming technique has been discussed. We have shown that a traditional dynamic programming approach may lead to an exponential algorithm for the problem and suggested to consider two of its realistic relaxations, called problems WR1 and WR2. Problem WR1 is to determine a single batch such that an average cost calculated as the ratio of the total cost to the length of the batch is minimized. Problem WR2 assumes that the total processing time needed to rework defective items in the same batch is a linear function of the batch size. We presented algorithms to solve problems WR1 and WR2, which are polynomial in N. It is worth noting that the algorithms can be modified such that in their time complexity estimations N is the number of distinct deadlines. We have also discussed a multi-product case and described the ideas of the algorithms for generalizations of problems WR1 and WR2 to the case of multiple products.

An open question is the time complexity of the general problem for the case of a single product or any constant number of products. Further research can be undertaken to resolve this question.

Another issue for further research is to extend the analysis in this paper to variants of

the considered problem. Often a certain percentage of defective products might be non-reworkable. This aspect should be easy to integrate in the presented scheduling algorithm. Concerning deterioration we also find situations where this process does not take place continuously, but where defective items keep their state until they reach a critical life-time after which they turn to become non-reworkable. This situation needs a specific treatment in being modeled as scheduling problem. The same holds if we relax the restriction of a one-to-one batch coordination policy which even under stationary assumptions only will be optimal in special cases (see Minner and Lindner [10]). In general it has to be allowed that multiple rework batches follow one manufacturing batch and vice versa.

Acknowledgements

References

[1] de Brito M. and Dekker R. (2004) Reverse logistics: a framework. In: Dekker R., Fleischmann M., Inderfurth K. and van Wassenhove L. N. (eds.), *Quantitative approaches to reverse logistics*, Springer, to appear.

[2] Flapper S.D.P., Fransoo J.C., Broekmeulen R.A.C.M. and Inderfurth K. (2002) Planning and control of rework in the process industries: a review, *Production Planning & Control* **1** 26-34.

[3] Flapper S.D.P. and Teunter R.H. (2003) *Logistic planning of rework with deteriorating work-in-process.* Working paper. Department of Technology Management, Eindhoven University of Technology, The Netherlands.

[4] Flapper S.D.P. and Jensen T. (2003), Logistic planning and control of rework, *International Journal of Production Research*, to appear.

[5] Goyal S.K. and Giri B.C. (2001) Recent trends in modeling of deteriorating inventory, *European Journal of Operational Research* **134** 1-16.

[6] Inderfurth K., Lindner G. and Rahaniotis N.P. (2003) *Lotsizing in a production system with rework and product deterioration.* Working paper 1/2003. Faculty of Economics and Management, Otto-von-Guericke-University Magdeburg, Germany.

[7] Inderfurth K. and Teunter R.H. (2003) Production planning and control of closed-loop supply chains. In: Guide Jr. V.D.R and van Wassenhove L.N. (eds.), *Business perspectives on closed-loop supply chains*, Carnegie Mellon University Press, 149-173.

[8] Lindner G., Buscher U. and Flapper S.D.P. (2001) *An optimal lot and batch size policy for a single item produced and remanufactured on one machine.* Working paper 10/2001. Faculty of Economics and Management, Otto-von-Guericke-University Magdeburg, Germany.

[9] Liu J.J. and Yang P. (1996) Optimal lot sizing in an imperfect production system with homogeneous reworkable jobs, *European Journal of Operational Research* **91** 517-527.

[10] Minner S. and Lindner G. (2004) Lotsizing decisions in product recovery management. In: Dekker R., Fleischmann M., Inderfurth K. and van Wassenhove L.N (eds.), *Quantitative approaches to reverse logistics*, Springer, to appear.

[11] Smith W.E. (1956) Various optimizers for single-stage production, *Naval Research Logistics Quarterly* **3** 59-66.

[12] Teunter R.H. and Flapper S.D.P. (2003) Lot-sizing for a single-stage single-product production system with rework of perishable production defectives, *OR Spectrum* **25** 85-96.