Master's degree thesis

LOG950 Logistics

Multi-Product Batching and Scheduling with Buffered Rework: The Case of a Car Paint Shop

Sergey Kovalev

Number of pages included the first page: 33

Molde, 2009-05-25

Molde University College

Publication agreement

Title: Multi-Product Batching and Scheduling with Buffered Rework: The Case of a Car Paint Shop

Author(s): Sergey Kovalev

Subject code: LOG950

ECTS credits: 30

Year: 2009

Supervisor: Irina Gribkovskaia

Agreement on electronic publication of master thesis

Author(s) have copyright to the thesis, including the exclusive right to publish the document (The Copyright Act §2).
All theses fulfilling the requirements will be registered and published in Brage HiM, with the approval of the author(s).
Theses with a confidentiality agreement will not be published.

I/we hereby give Molde University College the right to, free of charge, make the thesis available for electronic publication: ☒**yes** ☐**no**

Is there an agreement of confidentiality? ☐**yes** ☒**no**
(A supplementary confidentiality agreement must be filled in)
- If yes: **Can the thesis be online published when the period of confidentiality is expired?** ☐**yes** ☐**no**

Date: 2009-05-25

Abstract

The problem studied in this thesis refers to scheduling activities related to the imperfect production of several products on the same facility, which is motivated by the optimal scheduling in a car paint shop. Items of the same product are identical. Operations on the items are performed sequentially in batches, where each batch is a set of operations on the same product. Some of the produced items are of the required good quality and some items can be defective. Defectiveness of an item is determined by a given function of its product, its preceding product, and the position of its operation in the batch. Defective items are kept in a buffer of a limited capacity, and then they are remanufactured on the same facility. There is a minimum waiting time for any defective item before its remanufacturing can start. For each product, there is a sequence independent setup time which precedes the production of its first batch or its batch following a batch of another product. A due date is specified for each product (all items of the same product have the same due date) and the objective is to determine a schedule which minimizes the maximum lateness of products' completion times with respect to their due dates. The problem is proved to be NP-hard in the strong sense, and therefore a heuristic group technology solution approach is suggested, analyzed and tested with the computer program. The results of the research justify the application of the group technology approach to scheduling real car paint shops with buffered rework.

Key words: production, scheduling, batching, rework, group technology, car painting.

Acknowledgements

I would like to thank my supervisor, Professor Irina Gribkovskaia, for her comprehensive help through the process of my study in Molde University College and adaptation to Norway. Her wise advices resulted in a significant improvement of my academic knowledge.

I am extremely grateful to all the teaching staff of Molde University College, especially Øyvind Halskau and Arne Løkketangen, for interesting and accessible presentation of logistic disciplines.

I would like to thank my colleagues and friends for their invaluable assistance and good mood that they have been bringing in my life.

Finally, I would like to express my infinite and ineffable gratitude to my family members for their priceless support, which I have been feeling in spite of the long periods of being away from home.

Contents

1. Introduction

Consider a painting line in a car factory. A painting line is located in a car paint shop. Painting stage in the production of cars is preceded by car body production stage and followed by assembly stage, at which a car acquires its final form. A general car production scheme is presented in Fig.1. Different types of car bodies enter a car paint shop and have to be painted in various colors. After that, painted car bodies leave a paint shop and enter an assembly shop.

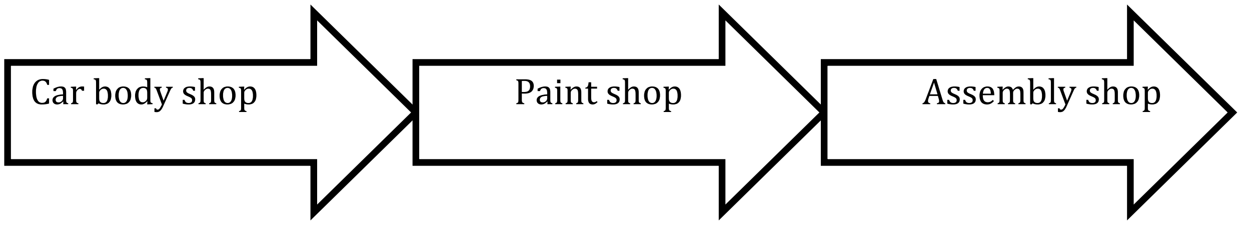


Figure 1. A general car production scheme.

Basic definitions must be given before the description of the problem of multi-product batching and scheduling with buffered rework. Let a pair (car model, color) be called a *product* and a car body of a given car model and a color an *item* of the corresponding product. Items are painted sequentially and setup times are required to switch from painting an item of one product to an item of another product. The setup time depends only on the product to be processed immediately after the setup. There is a quality inspection after which an item either goes to the inventory of good quality items or to the buffer of defective items, which has a limited capacity. The defects can be the spots of wrong colors or shades, the non-smooth surface, and the absence or low thickness of paint. They can be due to the insufficient cleaning of the spray guns, catching dust pieces by the paint, the imperfect positioning of the spray heads or the incorrect painting time. At a time moment to be decided, the last produced item goes to the inventory of good quality items or to the buffer, and, at the same time, some or all defective items leave the buffer, enter the line and are repainted. Any item can be repainted several times. Thus, the production of any item may consist of a number of operations: one *work* (manufacturing) operation and several *rework* (remanufacturing) operations. If an operation results in a defective item, this operation is called *defective*. A maximal set of operations on items of the same product manufactured and remanufactured sequentially since the last setup is called a *batch*. The *batch size* is the number of operations in this batch.

It is assumed that the defectiveness of an item is a given function of its product, its preceding product on the line, and the position of its operation in the batch. Pawlak and Rucinski [14] suggested that these characteristics are the main factors affecting an item's quality. The defectiveness function can be obtained by the historical data analysis or by a simulation. It is also assumed that there is a minimum time for any defective item to stay in the buffer, which is needed for the paint to dry. The objective is to construct a manufacturing/remanufacturing schedule such that the given product demands are satisfied close to their given due dates.

The *distinctions* of this thesis from the earlier studied problems of optimal planning work and rework processes are:

- the considered production is essentially discrete;
- the defective items are stored in a buffer of a limited capacity;
- a lower bound on the storage time is given;
- there are several products;
- product dependent setup times are given;
- no deterioration occurs to the defective items;
- the objective function does not include the inventory holding costs and the production costs.

  The inventory holding costs are not considered because in the automotive industry they are mainly determined by the costs of the storage capacities, which are given. The manufacturing costs are not considered because all the given demands must be satisfied and the cost of the production of any item is given. It is assumed that the setup costs are mainly determined by the workforce cost, which is constant. This assumption is relaxing because in reality the setup costs include the cost of the solvent used to purge the spray guns, see Gagne et al. [4].

- Minimization of maximum lateness of products' completion times with respect to their due dates is used as criterion in this research.

In the following section the literature review on the concerned problem is given. Section 3 contains problem definition with the required notations and formulation of the problem under study. Some additional assumptions not mentioned in the introduction and deadlock definition are given there as well. The proofs of the strong NP-hardness of two important special cases of the problem are presented in Section 4. A heuristic Group Technology (GT) approach to solving the problem is described and analyzed in Section 5. Section 6 presents a detailed GT algorithm in a form of pseudocode. Section 7 deals with the computational experiments carried out by the instrumentality of the GT algorithm. Section 8 states the conclusion and suggests the future research.

2. Literature review

Pawlak et al. [13] and Pawlak and Rucinski [14] observed the described situation at a real car factory. They discussed the factors that influence the appearance of the defects and suggested on-line solution procedures for minimizing the makespan and the number of color changes. In this thesis it is assumed that all the data are given and that a decision has to be made prior to the production start, i.e., off-line.

As a part of a more general car sequencing problem observed in *Groupe Renault*, the problem of scheduling a car paint shop was studied by Gagne et al. [4] and Solnon et al. [18]. The due date satisfaction criterion for the car sequencing problem observed at another car producer was studied by Guerre-Chaley et al. [5]. Boysen et al. [2] noticed that the due dates are important in an assembly-to-order environment. The car production process considered in these studies includes three main stages: the body production, the body painting and the car assembling in this order. Solnon et al. [18] assumed that the same color car bodies are produced contiguously and that the production sequence is the same for the paint shop and the assembly shop. Guerre-Chaley et al. [5], Spieckermann et al. [19] and Pawlak et al. [13] considered buffers between the stages, which are used for re-sequencing the car bodies. Meyr [10] wrote that the buffers between the stages are necessary because failures in the body and paint shops occur frequently. According to Holweg [7] cited by Meyr [10], the rework rate can be up to 40-50%. Computational complexity issues of a problem, in which a sequence of the car bodies of different models is given and a decision has to be made about the assignment of the colors to the car bodies of the same model, which minimizes the number of color changes, were studied by Epping et al. [3], Bonsma et al. [1], and Meunier and Sebo [9].

The thesis suggests using a group technology (GT) approach to build a good feasible schedule in case of buffered rework in a car paint shop. In general, GT is an approach to manufacturing and engineering management that seeks to achieve efficiency by exploiting similarities of different products and activities in their production/execution. Studies of GT were originated by Mitrofanov [11] and Opitz [12]. With respect to the scheduling, a GT environment is such that operations on the same product are never split into batches. The first publications on scheduling in the GT environments are due to Petrov [15], Yoshida et al. [21], and Ham et al. [6]. Results on the group scheduling problems were surveyed by Potts and Van Wassenhove [16] and Liaee and Emmons [8]. Properties which are sufficient for the optimality of a GT solution are established in this thesis. Most of these properties are naturally satisfied in real car paint shops, which justifies the usefulness of the GT approach to scheduling real car paint shops with buffered rework.

3. Problem definition

This section includes notations, definitions, additional assumptions, deadlock definition and example of a schedule in a car paint shop.

3.1. Notation

The following notations will be used.

F – the number of products;

n_f – the demand (number of good quality items) of product f;

p_f – the processing time requirement for any item of product f, the same value for manufacturing and remanufacturing;

s_f – the setup time required to start a batch of product f if it is sequenced first on the line or immediately after a batch of another product;

d_f – the due date for the demand satisfaction of product f;

C_f – the completion time of the last good quality item of product f in a given schedule;

$L_f = C_f – d_f$ – the lateness of product f in a given schedule;

$L_{max} = max\{L_f \mid f = 1,\ldots,F\}$ - the maximum lateness of products in a given schedule (objective function to be minimized);

B – the capacity of the buffer;

M – the minimum time that any defective item should stay in the buffer (the buffer lower time limit);

$G(g, f, r)$ – a 0-1 function of the product index g of the preceding batch, the product index f of the current batch, and the position r of an operation in this batch such that $G(g, f, r) = 1$ if this operation is defective, and $G(g, f, r) = 0$, otherwise. Here $g \in \{0, 1,\ldots, F\}$, $f \in \{1,\ldots,F\}$, $f \neq g$, and $g = 0$ applies for the case in which f is the first product on the line. Notice that, given g and f, function G values are the same for the same positions of different batches.

V_f , U_f – the lower and upper bounds, respectively, on the batch size of product f. Thus, $r \in \{1, 2,...,U_f\}$ for the function $G(g, f, r)$. Let the batch sizes be called *unbounded* if $V_f = 1$ and $U_f = \infty$, $f = 1,..., F$.

The upper bounds U_f can be used to model the requirement that the spray guns should be cleaned after a certain number of items has been painted, see Gagne et al. [3].

3.2. Additional assumptions

It is assumed that the time of transporting an item to or from the buffer is equal to zero. Furthermore, any item that has stayed in the buffer for at least M time units can leave the buffer (there is a direct access to any item in the buffer). No two items can be manufactured or remanufactured at the same time, and no item can be manufactured or remanufactured while setting the line up. Since the line is expensive equipment, no idle time is allowed if there is an item to be processed or a setup to be performed at this time.

3.3. Deadlock definition

The line can be blocked by a defective item if: 1) the buffer is full, and 2) no item can leave the buffer because no item has stayed there for at least M time units. If the line is blocked, no item can be manufactured or remanufactured. However, a setup can be performed even if the line is blocked. A situation that there is a time interval in which no item is produced and no setup is performed, while there is an item to be manufactured (not the one in the buffer) called a *deadlock*, should be avoided. The situation that may happen at the end of the production, in which there is no item to be manufactured and the buffer is not empty, is not considered as a deadlock. The deadlock can always be avoided if the total setup and production time between the completion of a defective item and the completion of B-th defective item following this item is at least M for any feasible sequence of manufacturing/remanufacturing operations. For example, the deadlock can always be avoided if the batch sizes are unbounded and $M \leq B \cdot pmin$, where $p_{\min} = \min\{p_f \mid f = 1,...,F\}$.

3.4. Problem formulation

The problem is to construct a feasible manufacturing/remanufacturing schedule such that no deadlock occurs (if such a schedule exists), all the demands are satisfied with no overproduction, and the maximum lateness L_{max} is minimized. This problem is to be denoted as *P (W, R, B,* L_{max}*)*, which is an abbreviation for *Problem (Work, Rework, Buffer,* $Lateness_{max}$*)*.

Deadlock handling methods in computer and manufacturing systems were discussed and a deadlock avoidance scheme was presented by Valckenaers and Van Brussel [19], who used in their developments data from an existing car paint shop in Sindelfingen (Germany). Notice that an ideal implementation of a feasible solution to the problem *P (W, R, B,* L_{max}*)* will not need a deadlock handling mechanism.

Due to the fact that items of the same product are identical, it can easily be seen that a search for an optimal solution can be limited to schedules in which defective items of the same product leave the buffer in the same order as they enter it, following the well known First-In-First-Out (FIFO) strategy. A managerial implication of this observation is that the buffer can be designed as a collection of unidirectional lines each of which is dedicated to a specific product.

3.5. Example of a schedule for the problem P (W, R, B, L_{max})

An example of a schedule for the problem *P (W, R, B,* L_{max}*)* is given in Fig. 2. In the corresponding problem, F = 2, n_1 = 10, n_2 = 9 and B = 2. It is assumed that $M \le B \cdot$ pmin = 2 min$\{t_3 - t_2, t_8 - t_7\}$. Therefore, no deadlock occurs.

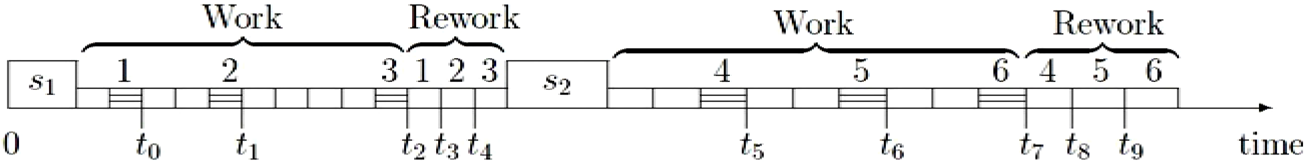


Figure 2. An example of a schedule. Hatched boxes represent defective operations.

Let vector $b(t)=(b_1(t),...,b_F(t))$ describe the buffer content at time t, where $b_f(t)$ is the number of items of product f in the buffer at time t. For the example in Fig. 2, $b(t) = (0, 0)$ for $t < t_0$ and $t \geq t_9$. Other values of $b(t)$ are given in Table 1.

$t \in$	$[t_0,t_1)$	$[t_1,t_2)$	$[t_2,t_3)$	$[t_3,t_4)$	$[t_4,t_5)$	$[t_5,t_6)$	$[t_6,t_7)$	$[t_7,t_8)$	$[t_8,t_9)$
$b(t)$	(1,0)	(2,0)	(2,0)	(1,0)	(0,0)	(0,1)	(0,2)	(0,2)	(0,1)

Table 1. Values of $b(t)=(b_1(t),b_2(t))$ for the example in Fig. 2.

4. Proving strong NP-hardness

The problem of deciding whether there exists a feasible schedule for the problem $P\ (W, R, B, L_{max})$ is denoted as the problem *Decide-Deadlock*. A special case of the problem $P\ (W, R, B, L_{max})$ in which there exists a feasible schedule is denoted as the problem *No-Deadlock*. The proof that each of the problems *Decide-Deadlock* and *No-Deadlock* is NP-hard in the strong sense is given in the following subsections. Therefore, the general problem $P\ (W, R, B, L_{max})$ is NP-hard in the strong sense as well.

Theorems 1 and 2 will show that an optimal polynomial time solution algorithm for the problem $P\ (W, R, B, L_{max})$ is unlikely to exist, and therefore efficient and practically relevant heuristic procedures are of interest.

4.1.1. Formulation of Theorem 1.

Theorem 1. *The problem Decide-Deadlock is NP-hard in the strong sense even if all setup times are equal to zero, the batch sizes are unbounded, and the function G(g, f, r) is independent of g.*

4.1.2. Proof of Theorem 1.

Proof. A reduction from the strongly NP-complete problem *3-Partition* is used.

3-Partition problem formulation: Given $3q + 1$ positive integer numbers $h_1,...,h_{3q}$ and H such that $H/4 < h_i < H/2$, $i = 1,...,3q$, and $\sum_{i=1}^{3q} h_i = qH$, is there a partition of the set {1,..., 3q} into q disjoint sets $X_1,..,X_q$ such that $\sum_{i \in X_l} h_i = H$ for $l = 1,\ldots,q$?

Given an instance of 3-Partition, the following instance of the problem *Decide-Deadlock* is constructed. There are $3q + 1$ products: $3q$ partition products $f = 1,..., 3q$, and one so-called *enforcer* product denoted as E. The demand of each partition product is one unit, i.e., $n_f = 1$, $f = 1,...,3q$, and the demand for the enforcer product is $n_E = q + 1$. The processing requirements are $p_f = h_f$, $f = 1,..., 3q$, and $p_E = 1$. The batch sizes are all unbounded. The function *G(g, f, r)* is such that operations on the partition products are all non-defective, odd operations on the enforcer product are all defective and even operations on the enforcer product are all non-defective. All the setup times are equal to zero, the buffer capacity is $B = 1$, and the buffer lower time limit is $M = H + 2$. The due dates play no role in the problem *Decide-Deadlock* because there is no constraint related to the due dates. Therefore, they can be chosen arbitrarily. It is shown below that *3-Partition* has a solution if and only if there exists a feasible schedule for the constructed instance of the problem *Decide-Deadlock*.

Consider an arbitrary feasible schedule for the constructed instance of the problem *Decide-Deadlock*. Observe that there are exactly $q + 1$ non-defective operations on the enforcer product, and the line is blocked (idle) between the last defective and the last non-defective operation on the enforcer product because otherwise these two last operations would belong to different batches and the very last operation would be the only operation of the last batch, which would be defective due to the definition of the function *G(g, f, r)*. Since the schedule is feasible, there is no idle time before the last defective operation. Due to the buffer lower time limit $M = H + 2$ and the buffer capacity $B = 1$, the completion times of any two consecutive defective operations on the enforcer product should be at least $H + 2$ time units apart each other. Therefore, there are q time intervals of length at least H before the last defective operation, which should be filled with the operations on the partition products so that there is no idle time. Since the processing times of these operations are equal to h_f, $f = 1,\ldots, 3q$, and $\sum_{f=1}^{3q} h_f = qH$, the q intervals can be filled with no idle time if and only if problem *3-Partition* has a solution.

It follows from Theorem 1 that a modification of the problem *P (W, R, B, L_{max})* in which L_{max} is replaced by any other objective function, or even if there is no objective at all is NP-hard in the strong sense.

4.2.1. Formulation of Theorem 2.

Theorem 2. *The problem No-Deadlock is NP-hard in the strong sense even if all setup times are equal to zero, the batch sizes are unbounded, and the function G(g, f, r) is independent of g.*

4.2.2. Proof of Theorem 2.

Proof. The proof is similar to the proof of Theorem 1. Given an instance of the problem *3-Partition*, the following instance of the problem *P (W, R, B, L_{max})* is constructed, which will be later shown to be an instance of the problem *No-Deadlock*. Assume that there are $5q + 2$ products: $3q$ partition products $f = 1,...,$

$3q$, $q + 2$ number of *E-enforcer* products denoted as E_j, $j = 1,...,q + 2$, and q number of *D-enforcer* products denoted as D_j, $j = 1,...,q$. All the setup times are equal to zero, the buffer capacity is $B = 1$, and the buffer lower time limit is $M = H + 2$. The demand of each product is one unit. The processing requirements are $p_f = h_f$, $f = 1,..., 3q$, $p_{Ej} = 1$, $j = 1,...,q + 2$, and $p_{Dj} = H$, $j = 1,...,q$. All the partition products have a common due date $d = (H + 2)q$. The due date of the enforcer product E_j is equal to $d_{Ej} = 2 + (H + 2)(j - 1)$, $j = 1,...,q+1$. The due date of the enforcer product $E(q + 2)$ and the due dates of D-enforcer products are sufficiently large such that they can never be exceeded. For example, they are equal to $d + \sum_{f=1}^{3q} p_f + 2p_{E(q+1)} + 2p_{E(q+2)} + M = (2H + 2)q + H + 6$. The batch sizes are all unbounded. The function $G(g, f, r)$ is such that operations on the partition products and D-enforcer products are all non-defective, odd operations on any E-enforcer product are all defective and even operations on any E-enforcer product are all non-defective.

Observe that the following schedule is feasible for the constructed instance of the problem P (W, R, B, L_{max}). There is a single batch of size two for each E-enforcer product. There is a single batch of size one for each D-enforcer product and each partition product. In the schedule, the first $q + 1$ odd batches are the batches of the E-enforcer products in the order $E1,..., E(q+1)$, and the first q even batches are batches of the D-enforcer products. The next $3q$ batches are the batches of the partition products, and the last batch is the batch of the E-enforcer product. See Fig. 3 for an illustration.

It is deduced that the constructed instance is indeed an instance of the problem *No-Deadlock*. It is shown below that *3-Partition* has a solution if and only if there exists a feasible schedule for the constructed instance of the problem *No-Deadlock* with value $L_{max} \leq 0$.

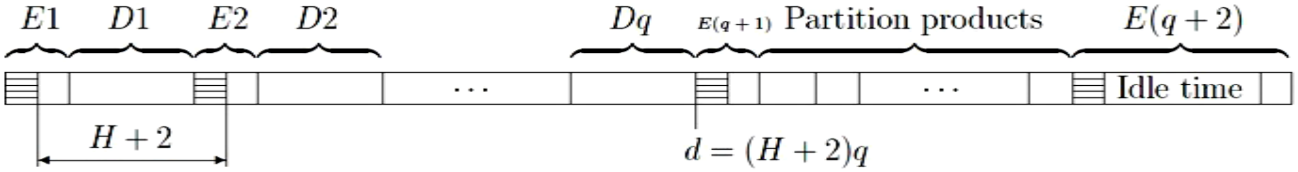


Figure 3. A feasible schedule for the constructed instance of the problem *P(W, R, B, L_{max})*.

Consider an arbitrary feasible schedule for the constructed instance of the problem *No-Deadlock* with value $L_{max} \leq 0$. Similar to the previous proof, observe that there are exactly $q + 2$ non-defective operations on the *E*-enforcer products, and the line is blocked (idle) between the last defective and the last non-defective operation on the *E*-enforcer product $E(q+2)$. Since the schedule is feasible, there is no idle time before the last defective operation. Due to the buffer lower time limit $M = H + 2$ and the buffer capacity $B = 1$, the completion times of any two consecutive defective operations on the *E*-enforcer products should be at least H + 2 time units apart each other. Therefore, the $(q + 1)$-st defective operation can start not earlier than at time $(H + 2)q$, which is due date for the partition products. Since this due date cannot be exceeded, the partition products should fill completely the q time intervals of length at least H between every two consecutive defective operations in the time period $[0, d]$. Since the processing times of these operations are equal to h_f , f =1,..., $3q$, and $\sum_{f=1}^{3q} h_f = qH$, the q intervals can be completely filled if and only if problem 3-Partition has a solution.

Theorem 2 implies that a modification of the problem *P (W, R, B, L_{max})*, in which the objective function is the total unsatisfied demand, is NP-hard in the strong sense. Furthermore, it follows from its proof that a modification of the problem *P (W, R, B, L_{max})*, in which deadlocks are allowed is NP-hard in the strong sense.

5. A group technology solution approach

A heuristic Group Technology (GT) approach to solving the problem *P (W, R, B, L_{max})* is analyzed in this section. This GT approach suggests that a single batch is formed for each product.

The group technology (GT) scheduling decisions have several benefits comparing to the non-GT decisions. First of all, they are easy to implement because the structure of the GT schedule is simple. Secondly, coordination of the GT decisions of all the participants of the production/supply process is simpler because less information is involved into the coordination process. Thirdly, a GT schedule minimizes the number of setups, and the saved working time of the setup operators can be redirected for other purposes. Furthermore, the amount of solvent used to purge the spray guns is minimized. Fourthly, a GT schedule can be easily recomputed to adjust to a new production environment such as new demands or line breakdowns. The GT solution approach may be inefficient if the succeeding production prefers an even supply of different products over time. In this case, an intermediate buffer can be used to modify the product sequence accordingly. The GT decisions are used for scheduling real car paint shops, see, for example, Solnon et al. [18]. However, no results on their theoretical or practical efficiency were reported in the literature.

Let the products be renumbered in the *Earliest Due Date (EDD)* order such that $d_1 \leq \cdots \leq d_F$. Consider a heuristic solution for the problem $P\ (W, R, B, L_{max})$, in which a single batch is formed for each product, every defective item of the same product leaves the buffer at the earliest possible time following the FIFO strategy, and the products are sequenced in the EDD order $(1,...,F)$ with ties broken arbitrarily. Let such a solution be denoted as a GT-EDD *schedule*. Notice that the GT-EDD schedule may be infeasible in general because a deadlock may occur for it. The conditions under which a GT-EDD schedule is an optimal solution for the problem $P\ (W, R, B, L_{max})$ are established in the following subsection.

5.1 Sufficient conditions for the optimality of the GT-EDD schedule

It is convenient to introduce some additional notation, which is illustrated in Fig. 4.

Consider functions $\Delta_f(g,h) := \sum_{r=1}^{h} G(g,f,r), f = 1,\dots,F$ of variables g and h, which are defined for $g = 0,1,\dots,F$, $g \neq f$, and $h \in \{1,2,\dots,U_f\}$. The value of $\Delta_f(g,h)$ is the number of defective operations among all h operations of a batch of product f preceded by a batch of product g. If the lengths of the intervals of h, in which this function has the same value, non-decreases as h increases with a possible exception for the last interval, then the function $\Delta_f(g,h)$ has a *concave staircase structure* in h.

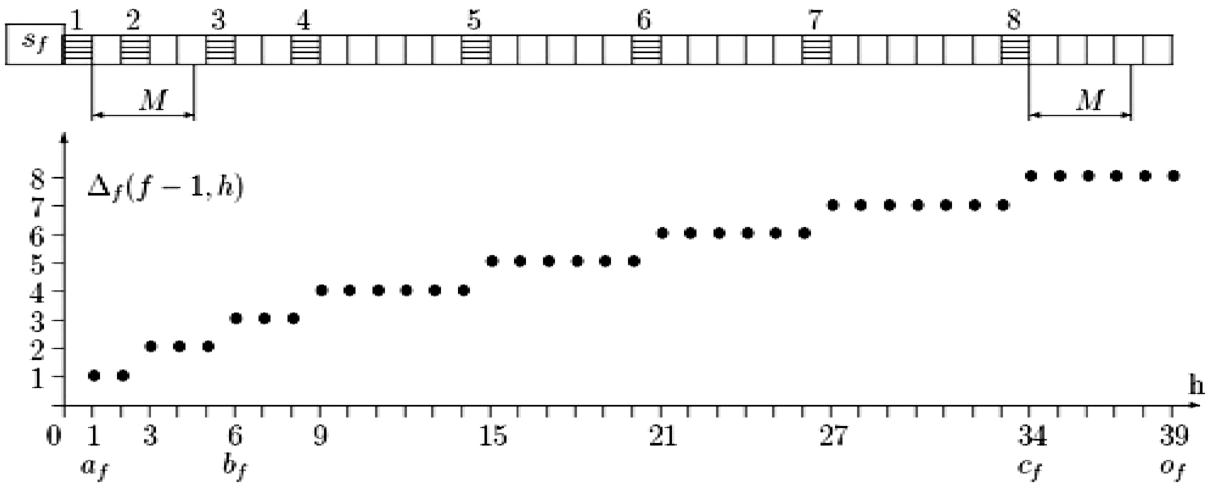


Figure 4. Example of the function $\Delta_f(f-1,h)$ with a concave staircase structure in h. For the corresponding batch with $n_f = 31$ good quality items given above, $a_f = 1, b_f = 6, c_f = 34$ and $o_f = 39$. Hatched boxes represent defective operations.

Recall that the products are numbered in the EDD order. Consider a batch of product f preceded by a batch of product $f-1$, which contains n_f non-defective operations. Denote by o_f the total number of operations in this batch. It is the unique solution of the equation $o_f - \Delta_f(f-1,o_f) = n_f$. Obviously, the last operation is non-defective, i.e. $G(f-1,f,o_f) = 0$. Denote by a_f the position of the earliest defective operation in this batch, $a_f = \min\{h \mid G(f-1,f,h) = 1\}$. Define $a_f = 0$ if there is no defective operation. Denote by b_f the earliest position in this batch, in which the first defective item can be remanufactured, $b_f = a_f + 1 + \left\lceil \frac{M}{p_f} \right\rceil$. Denote by c_f the position of the last defective operation in this batch, $c_f = \max\{h \mid h \leq o_f,\ G(f-1, f, h) = 1\}$.

5.1.1 Formulation of Theorem 3

Theorem 3. *The GT-EDD schedule is optimal for the problem P (W, R, B, L_{max}) if the following conditions (i)-(vi) are satisfied.*

(i) each function $\Delta_f(g, h)$ is minimized in g at $g = f - 1$.

(ii) each function $\Delta_f(g, h)$ has the concave staircase structure in h.

(iii) $\Delta_f(f-1, b_f-2) \leq B$ and $\Delta_f(f-1, b_f-1) \leq B+1$, $f = 1,...,F$. For the example in Fig. 4 this condition is satisfied if $B \geq 2$.

(iv) if $G(f-1, f, r) = 1$ for at least one r, $1 \leq r \leq o_f$, then $M \leq p_f(o_f - c_f - 1)$, $f = 1,...,F-1$. For the example in Fig. 4 this condition is satisfied.

(v) if $G(f-1, f, r) = 1$ for at least one r, $1 \leq r \leq o_f$, then $Vf \geq af$, $f = 1,...,F$.

(vi) $V_f \leq o_f \leq U_f$, $f = 1,...,F$.

5.1.2. Proof of Theorem 3

Proof. Consider an optimal schedule S^* for the problem *P (W, R, B, L_{max})* with the objective value L^*_{max}. Let S^* contain x_f defective operations on product *f*, $f = 1,...,F$. Now consider a batch scheduling problem, which differs from the problem *P (W, R, B, L_{max})* in that each product *f* consists of $n_f + x_f$ items, $f = 1,...,F$, and the production is perfect such that every manufactured item is of the required good quality. Denote this problem as *Perfect*. Santos [17] proved that there exists an optimal solution of the problem *Perfect*, in which no product is split into batches, and the products are sequenced in the EDD order, see also Potts and Van Wassenhove [46]. Since schedule S^* is feasible for the problem *Perfect*, optimal solution value of this problem, L^{0*}_{max}, is a lower bound for L^*_{max}: $L^{(0)}_{max} \leq L^*_{max}$.

Furthermore, the optimal solution value, $L_{max}^{(1)}$, of the problem *Perfect*, in which each product f consists of no more than $n_f + x_f$ items, is a lower bound for $L_{max}^{(0)}$: $L_{max}^{(1)} \leq L_{max}^{(0)}$. Therefore, $L_{max}^{(1)}$ is a lower bound for L_{max}^{*}.

Let the GT-EDD schedule contain y_f defective operations of product f, f = 1,...,F . Assume that the conditions (i)-(iii), (iv), (v) and (vi) are satisfied. Firstly, it will be shown that $y_f \leq x_f$, f = 1,...,F. Let there be q_f batches of product f in the optimal schedule S^*, and let the j-th batch of product f consist of $o_f^{(j)}$ manufacturing and remanufacturing operations, among which there are $x_f^{(j)}$defective operations. Consider an artificial schedule which differs from the GT-EDD schedule in that the single batch of product f consists of $\sum_{j=1}^{q_f} o_f^{(j)}$ manufacturing and remanufacturing operations, f = 1,...,F. Let $z_f^{(j)}$denote the number of defective operations among the operations in the positions $o_f^{(j-1)}$ + 1, $o_f^{(j-1)}$ + 2,…, $o_f^{(j)}$, $o_f^{(0)}$ = 0, and let z_f denote the total number of defective operations on product f in this artificial GT-EDD schedule. Due to the properties (i), (ii) and (v), it is deduced that $z_f^{(j)} \leq x_f^{(j)}$, j = 1,...,q_f , which implies $z_f \leq x_f$, f = 1,...,F. Therefore, the artificial GT-EDD schedule has the same total number of operations, the same or smaller number of defective operations and the same or larger number of non-defective operations on each product f in comparison with the schedule S^*. Since the GT-EDD schedule and the artificial GT-EDD schedule have the same sequence of operations on product f up to the operation corresponding to the production of n_f -th good quality item of this product, f = 1,...,F, it is deduced that $y_f \leq z_f$, and, hence, $y_f \leq x_f$, f =1,...,F. It follows from the above discussion that the optimal solution value, $L_{max}^{(2)}$, of the problem *Perfect*, in which each product f consists of $n_f + y_f$ items, is a lower bound for L_{max}^{*}: $L_{max}^{(2)} \leq L_{max}^{*}$.

It remains to show that there is no idle time in the GT-EDD schedule. If so, then it coincides with the optimal solution of the problem *Perfect*, in which each product f consists of $n_f + y_f$ items, and, therefore, has the value $L_{max}^{(2)}$. The no idle time property of the GT-EDD schedule is guaranteed by the conditions (ii), (iii), (iv) and (vi). Condition (vi) guarantees that the batch sizes in the GT-EDD schedule are all feasible. Provided that the buffer is empty when an execution of the batch of product *f* starts, condition (iii) ensures that the buffer capacity will not be exceeded until the earliest time when the first defective item of this product can leave the buffer. Due to the condition (ii), this statement is also satisfied for any defective item. Since the defective items leave the buffer at the earliest possible times and the buffer capacity is never exceeded, condition (iv) guarantees that all operations of the same batch can be executed with no idle time so that any defective item can leave the buffer at a time when an operation on another item completes. Furthermore, at the end of the product *f* execution, the buffer will be empty. Thus, the GT-EDD schedule contains no idle time, which completes the proof.

5.2. Examples of non-optimal GT-EDD schedules when one of the conditions (i)-(vi) fails

Now it can be shown that if one of the conditions (i)-(vi) is violated and the remaining conditions are satisfied, then there exists an instance of the problem $P(W, R, B, L_{max})$ for which no GT-EDD schedule is optimal. This statement does not mean that (i)-(vi) are the necessary conditions of the existence of an optimal GT-EDD schedule for any given instance because there may exist an instance, for which some of these conditions are violated but the GT-EDD schedule is optimal. In all the instances given below there are two products, $s_1 = s_2 = 0$, $p_1 = p_2 = 1$, and the definition of the functions $G(g, f, r)$ can be deduced from the pictures of the schedules.

Assume that condition (i) fails, $n_1 = n_2 = 1$, $d_1 = 2$, $d_2 = 3$, $M = 0$, $B = 1$, and the batch sizes are unbounded. An optimal schedule and the unique GT-EDD schedule are given in Fig. 5.

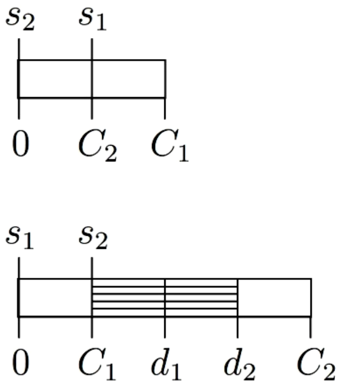


Figure 5. Condition (i) fails: $\Delta_2(1, h) > \Delta_2(0, h)$.

Assume that condition (ii) fails, $n_1 = 1$, $n_2 = 2$, $d_1 = 3$, $d_2 = 5$, $M = 0$, $B = 1$, and the batch sizes are unbounded. This example is illustrated by Fig. 6.

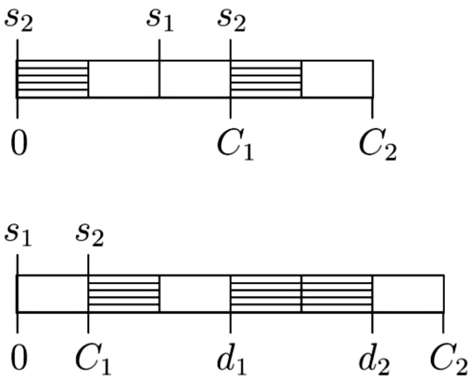


Figure 6. Condition (ii) fails: $\Delta_2(1, h)$ does not have concave staircase structure in h.

Assume that condition (iii) fails, $M = 2$, $B = 1$, $n_1 = 1$, $n_2 = 3$, $d_1 = 2$, $d_2 = 3$, and the batch sizes are unbounded. The function $G(1,2,r)$ is such that the first two operations are defective. In this case no feasible schedule exists, including the GT-EDD schedule, see Fig. 7.

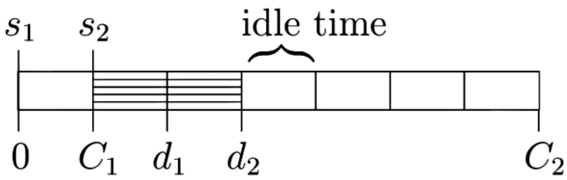
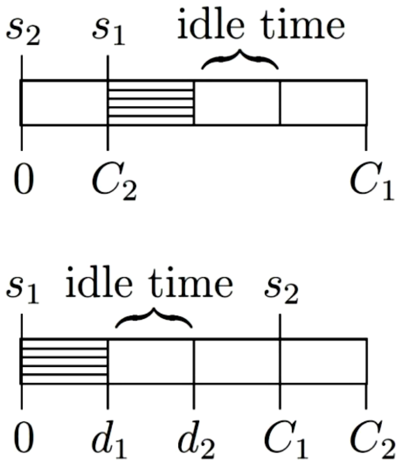


Figure 7. Condition (iii) fails: $\Delta_f\left(f-1, b_f-2\right)$ = $\Delta_2(1,2)$ = 2 $>$ B = 1.

Assume that condition (iv) fails, M = 1, B = 1, n_1 = n_2 = 1, d_1 = 1, d_2 = 2, and the batch sizes are unbounded. In this case, the GT-EDD schedule is infeasible and a feasible schedule exists, see Fig. 8.

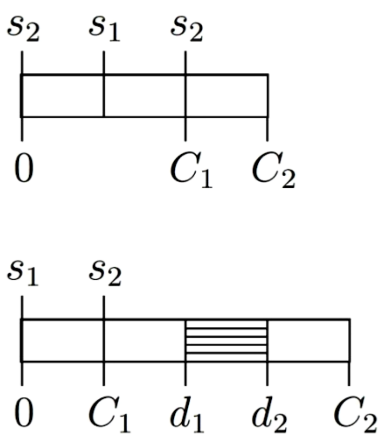


← optimal schedule, L_{max} = 3

← GT-EDD schedule (infeasible)

Figure 8. Condition (iv) fails: M = 1 > $p_1(o_1$ - c_1 - 1) = 0.

Assume that condition (v) fails, M = 0, B = 1, n_1 = 1, n_2 = 2, d_1 = 2, d_2 = 3, and the batch sizes are unbounded. An optimal schedule and the unique GT-EDD schedule are given in Fig. 9.



← optimal schedule, L_{max} = 0

← GT-EDD schedule, L_{max} = 1

Figure 9. Condition (v) fails: V2 = 1 < a2 = 2.

Finally, assume that condition (vi) fails. In this case, the GT-EDD schedule is infeasible because at least one of its batch sizes violates the corresponding lower or upper bound.

5.3 Practical relevance and managerial implications

Conditions (i) - (vi) are satisfied for the practically relevant situations, in which

- the number of defective operations is smaller if the production switches from a lighter color to the most similar darker color and the products with lighter colors have earlier due dates (condition (i));
- the probability that an operation is defective decreases as its position in the same batch increases (condition (ii));
- the buffer capacity is sufficiently large and the buffer lower time limit is sufficiently small to avoid a deadlock in the GT-EDD schedule (condition (iii));
- the buffer capacity is sufficiently large and the buffer lower time limit is sufficiently small such that no idle time occurs and the buffer is empty at the completion time of any product in the GT-EDD schedule (condition (iv));
- if the batch of any product in the GT-EDD schedule contains at least one defective operation, then every batch of this product in any schedule does (condition (v)),
- the batch sizes in the GT-EDD schedule are feasible (condition (vi)).

The established conditions justify the application of the GT solutions to scheduling the real car paint shops, even if the production is imperfect.

The managerial implications of the results in this section are twofold:

1. Conditions (i)-(vi) can be used for making a decision of implementing the GT-EDD schedule in a production environment with buffered rework.

2. Conditions (i)-(vi) can be used for making a decision of adapting the existing production environment so that the GT-EDD schedule can be expected to provide good solution quality. For example, it can be decided a priori that the products with lighter colors will have earlier due dates, the buffer lower time limit will be decreased by installing a more efficient drying device, or the buffer will be rebuilt to have a larger capacity. All these changes will increase the chances that the GT-EDD schedule minimizes the maximum deviation of the demand satisfaction times from their due dates, L_{max}, which is good for the coordination of decisions in the corresponding make-to-order supply and production chain.

6. GT-EDD algorithm

This section presents a detailed GT-EDD algorithm in a form of pseudocode with explanatory comments.

6.1. Notation

A schedule is determined by two structures: *TimeLine(i)*, $i = 0,1,\ldots,v$ and *TimeBuffer(j)*, $j = 0,1,\ldots,u$, where v and u are the last indices of the structures *TimeLine(i)* and *TimeBuffer(j)* correspondingly.

1. ***TimeLine(i) = (t(i), Event(i), ProdOnLine(i)),*** where

- *t(i)* - time instant of an event on the paint line;
- *Event(i)* ∈ *{Begin, SetupEnd, GoodEnd, DefectEnd, WaitEnd, DeadlockEnd}* - indicator variable of an event's end.
- *ProdOnLine(i)* - product on the line at a time *i*.

Variable *Begin* states for the beginning of a schedule. *SetupEnd* points on the end of the setup of a painting line for a certain product. Variables *GoodEnd* and *DefectEnd* mark the ends of good and defective operations accordingly. *WaitEnd* shows the time when the waiting for a drying item to leave the buffer ends. In other words it is idle time. The end of a deadlock is denoted as *DeadlockEnd.*

2. ***TimeBuffer(j) = (τ(j),b(j)),*** where

- *τ(j)* - time instant of an event in the buffer (right after *τ(j)* buffer content changes, but number of items in the buffer can stay the same),
- *b(j)* - number of items in the buffer right after *τ(j)*.

Auxiliary parameters:

TimeLeaveBuffer(k) - the earliest time at which *k*-th item in the buffer can leave it, *k* =1,...,*B*, where *B* is a capacity of the buffer.

It is maintained in the algorithm that *TimeLeaveBuffer(1)* < *TimeLeaveBuffer(2)* < ⋯ < *TimeLeaveBuffer(B)*.

RealNumberGood – the sum of already painted items with good quality.

BufferTime – a fixed time that any item spends in the buffer

Notation from subsections 3.1. and 5.1. will be used in pseudocode as well.

6.2. Pseudocode

Commentaries are marked in *italic font style*.

Initialization:

TimeLeaveBuffer(k) := 0, k = 1,…,B

i := 0

t(i) := 0

Event(i) := Begin

ProdOnLine(i) := 0

j := 0

τ(j) := 0

b(j) := 0

Main computations:

From f = 1 to F **Do1** – *product cycle, where products are sorted in the EDD order*;

Event(i) := SetupEnd

t(i + 1) := t(i) + s_f

ProdOnLine(i + 1) := f

RealNumberGood := 0

i := i + 1

From r = 1 to o_f **Do2 –** *operations cycle of product f*

If 1 TimeLeaveBuffer(0) > t(i) **and** RealNumberGood + b(j) = n_f **then**

Event(i) := WaitEnd

t(i + 1) := TimeLeaveBuffer(0)

ProdOnLine(i + 1) := f

i := i + 1

End If 1

If 1 *is needed to keep track of the sum of already painted items of good quality and items that in the meantime are drying in the buffer. If this sum equals the demand for this product then the line waits for the items to be dried. After that rework process continues.*

i := i + 1

t(i) := t(i - 1) + p_f

If 2 G(g, f, r) = 0 **then –** *instructions in case of a **good quality** operation*

RealNumberGood := RealNumberGood + 1

Event(i) := GoodEnd

If 2.1 TimeLeaveBuffer(1) ≤ t(i) **and** b(j) > 0 **then –** *reflects a situation when a defective item can leave the buffer*

TimeLeaveBuffer(k) := TimeLeaveBuffer(k + 1), for k = 1,…,b(j) – 1

TimeLeaveBuffer(b(j)) := 0

τ(j + 1) := t(i)

b(j + 1) := b(j) – 1

j := j + 1

Continue EndDo2

EndIf 2.1

If 2.2 TimeLeaveBuffer(1) > t(i) **and** b(j) > 0 **then** – *there is at least one defective item in the buffer but it cannot leave it, because the drying time has not passed yet*

Continue EndDo2

EndIf 2.2

If 2.3 b(j) = 0 **then –***there is nothing in the buffer yet*

Continue EndDo2

EndIf 2.3

EndIf 2

If 3 G(g, f, r) = 0 **then –** *instructions in case of a **defective** operation*

Event(i) := DefectEnd

If 3.1 b(j) = 0 **then** - *there is nothing in the buffer yet*

τ(j + 1) := t(i)

b(j + 1) := 1

TimeLeaveBuffer(1) := t(i) + BufferTime

j := j + 1

Continue EndDo2

EndIf 3.1

If 3.2 TimeLeaveBuffer(1) ≤ t(i) **and** b(j) > 0 **then –** *there is at least one item in the buffer and an item from the buffer which has already dried there can leave it*

TimeLeaveBuffer(k) := TimeLeaveBuffer(k + 1), for k = 1,…,b(j) – 1

TimeLeaveBuffer(b(j)) := t(i) + BufferTime

τ(j + 1) := t(i)

b(j + 1) := b(j)

j := j + 1

Continue EndDo2

EndIf 3.2

If 3.3 TimeLeaveBuffer (1) > t(i) **and** 0 < b(j) < B **then** - *there is at least one item in the buffer and it cannot leave the buffer yet*

TimeLeaveBuffer(b(j) + 1) := t(i) + BufferTime

τ(j + 1) := t(i)

b(j + 1) := b(j) + 1

j := j + 1

Continue EndDo2

EndIf 3.3

If 3.4 TimeLeaveBuffer (1) > t(i) **and** b(j) = B **then –** *deadlock situation*

t(i + 1) := TimeLeaveBuffer(1)

Event(i) := DeadlockEnd

ProdOnLine(i + 1) := f

i := i + 1

TimeLeaveBuffer(k) := TimeLeaveBuffer(k + 1), for k = 1,…,B – 1

TimeLeaveBuffer(B) := t(i) + BufferTime

τ(j + 1) := t(i)

b(j + 1) := B

j := j + 1

Continue EndDo2

EndIf 3.4

EndIf 3

EndDo2

C_f := t(i) – *completion time of product f*

L_f – *lateness of product f in a built schedule*

EndDo1

$L_{max} = \max\{L_f \mid f = 1,\ldots,F\}$ - *the maximum lateness of products in a given schedule*

Output:

L_{max}

TimeLine(i), i = 1,…,v, *as a table with 3 columns, representing the fields of structure TimeLine*

TimeBuffer(j), j = 1,…,u, *as a table with 2 columns, representing the fields of structure TimeBuffer*

7. Computational experiments

In order to test the GT-EDD algorithm it was programmed in C++ language using Microsoft Visual Studio 9.0 software. The code of the GT-EDD algorithm corresponds to the pseudocode in Section 6.

G(g, f, r) function works as a random number generator, based on the predefined probability of defective operations. In the test instances presented in the following subsection this probability equals 30%. *G(g, f, r)* was coded in the program as a generator of Boolean variables for every combination of *g* and *f* (preceding and current products) until the number of 0-variables (good quality operations) equals the demand for a given product. After that, products are sorted by EDD order and a schedule of single batches for all products is constructed using the values of generated *G(g ,f, r)* function. The batch sizes are unbounded.

Computer characteristics are the following:

- *Operating system*: Windows Vista Home Premium (64-bit), SP1;
- *Computer model*: HP Pavilion dv7 Notebook PC;
- *Processor*: Intel(R) Core(TM)2 Duo CPU T9400 2.53GHz;
- *Memory(RAM)*: 4.00 GB.

7.1. Test instances

There are 9 test instances that were created to test GT-EDD algorithm. Parameters for the test instances are shown in Table 2.

Test instance, #	Number of products	Probability of a defective operation	Demand, range, units	Setup time, range, min	Processing time, range, min	Due date, range, days	Buffer capacity, units	Buffer time, min
1	30	30%	[10, 20]	[10, 20]	[2, 10]	[1, 3]	2	20
2	30	30%	[10, 20]	[10, 20]	[2, 10]	[1, 3]	3	20
3	30	30%	[10, 20]	[10, 20]	[2, 10]	[1, 3]	5	20
4	40	30%	[10, 20]	[10, 20]	[2, 10]	[1, 3]	2	20
5	40	30%	[10, 20]	[10, 20]	[2, 10]	[1, 3]	3	20
6	40	30%	[10, 20]	[10, 20]	[2, 10]	[1, 3]	5	20
7	50	30%	[10, 20]	[10, 20]	[2, 10]	[1, 3]	2	20
8	50	30%	[10, 20]	[10, 20]	[2, 10]	[1, 3]	3	20
9	50	30%	[10, 20]	[10, 20]	[2, 10]	[1, 2]	5	20

Table 2. The parameters for the 9 test instances.

Test instances are divided into 3 groups of 30, 40 and 50 products. Each group consists of 3 instances. These instances differ from each other only by the buffer capacity which takes the values of 2, 3, or 5 units. Probability of a defective operation and a fixed time that any item spends in the buffer denoted in Table 2 as buffer time are held constant for all instances. Demand, setup times, processing times and due dates are randomly generated within the ranges specified in Table 2. Due dates vary between 1 and 3 days in minutes.

Notice that buffer time is twice larger than the largest processing time. Moreover, the generated $G(g,f,r)$ function has not a concave staircase structure in h. Therefore, none of GT-EDD schedules for these test instances will be optimal.

7.2. Computational results

The computational results of the 9 test instances are shown in Table 3. The achieved results were computed in negligibly small amount of time (less than a second). Only variable parameters are included in Table 3. Objective values L_{max} are marked in **bold font type.**

Test instance, #	Number of products	Buffer capacity, units	**L_{max}, min**
1	30	2	**400**
2	30	3	**283**
3	30	5	**247**
4	40	2	**2032**
5	40	3	**1943**
6	40	5	**1919**
7	50	2	**3553**
8	50	3	**3367**
9	50	5	**3348**

Table 3. Computational results of the 9 test instances.

As it is seen from Table 3, L_{max} increases as the number of products grows. The correlation coefficient for these arrays equals 0.997

Meanwhile, the variability of L_{max} values dependent on buffer capacity is not high. Waitings and deadlocks have the same impact on a painting process in that the line stands idle. However, the number of waitings for drying items to leave the buffer is much higher than the number of deadlocks and it is more stable, see Table 4. A larger buffer capacity entails the number of deadlocks to decrease. Tables 3 and 4 reflect this regularity: the larger the buffer capacity the smaller is the number of deadlock. But the number of waitings does not change significantly with the change of the buffer capacity.

Test instance, #	Number of waitings	Number of deadlocks
1	40	24
2	41	9
3	43	0
4	39	18
5	41	4
6	40	0
7	54	33
8	58	6
9	58	0

Table 4. Comparison of the number of waitings and deadlocks for the 9 test instances.

The correlation coefficient between L_{max} and buffer capacity arrays is -0.046, although the dependence between them for the first 3 instances with 30 products is high – correlation coefficient equals -0.884.

Thus, having minimization of L_{max} as criterion, the managerial decision on the buffer capacity should be based on the number of products to be produced, the cost of lateness and the cost of the buffer itself, because an advantage in L_{max} induced by a larger buffer capacity may be too expensive considering the cost of extension. Although common sense prompts to extend the buffer capacity when the number of products increases, it can be seen from Table 3 that the positive effect on the objective function value L_{max} is not impressive. A preliminary use of GT-EDD heuristic can help to take the right decision, considering all the given data.

8. Conclusion

A problem of scheduling work and rework processes on a single facility with buffered rework is studied in this thesis. The problem is motivated by the optimal scheduling decisions in a car paint shop. The specificity of the problem is that the production is essentially discrete, the defective items are stored in a buffer of a limited capacity, a lower bound on the storage time is given, there are product dependent setup times, no deterioration occurs to the defective items, and the objective function is to minimize the maximum lateness of the product demand satisfaction times with respect to their given due dates. The defectiveness of an item is determined by a given function of three variables: the product of this item, the preceding product in the manufacturing/remanufacturing sequence, and the position of an operation on this item in its batch. An optimal search is limited to schedules which contain no deadlock. The deadlock is a situation when the buffer is full, a defective item blocks the line, and there is an item to be manufactured but it cannot because the line is blocked. The problem is proved to be NP-hard in the strong sense for two special cases in which the existence of a deadlock is unknown and known, respectively.

A heuristic Group Technology (GT) solution approach is suggested, which constructs the GT-EDD schedule, in which there is a single batch for each product, the products are sequenced in the Earliest Due Date (EDD) order, and defective items leave the buffer as soon as possible following the First-In-First-Out strategy. Sufficient conditions for the GT-EDD schedule to be an optimal solution for the studied problem are established. These conditions justify the application of the GT solutions in scheduling car paint shops with buffered rework.

Computational experiments were held so as to test the GT-EDD algorithm coded in C++. The computational results showed that having L_{max} as an objective function to be minimized, the managerial decision on the buffer capacity should be based on the number of products to produce, the cost of lateness and the cost of the buffer itself. The useful information on which this decision should be based on can be obtained by using the GT-EDD heuristic.

For future research, it is interesting to study related problems with the following features:

- The objective is to minimize the total (weighted) unsatisfied demand.
- Various strategies for emptying the buffer.

• Various layouts of the buffer.

• Batch sizes are bounded.

References

[1] Bonsma, P., Epping, Th., Hochstattler, W. (2006) Complexity results on restricted instances of a paint shop problem for words, *Discrete Applied Mathematics* **154** 1335-1343.

[2] Boysen, N., Fliedner, M., Scholl, A. (2009) Sequencing mixed-model assembly lines: Survey, classification and model critique, *European Journal of Operational Research* **192** 349-373.

[3] Epping, Th., Hochsta ttler, W., Oertel, P. (2004) Complexity results on a paint shop problem, *Discrete Applied Mathematics* **136** 217-226.

[4] Gagne, C., Gravel, M., Price, W.L. (2006) Solving real car sequencing problems with ant colony optimization, *European Journal of Operational Research* **174** 1427-1448.

[5] Guerre-Chaley, F., Frein, Y., Bouffard-Vercelli, R. (1995) An efficient procedure for solving a car sequencing problem. In: INRIA/IEEE Symposium on Emerging Technologies and Factory Automation **2** 385-393.

[6] Ham, I., Hitomy, K., Yoshida, T. (1985) Group technology: applications to production management, Kluwer-Nijhoff, Boston.

[7] Holweg, M. (2002) Product variety -friend or foe? Technical Report, Lean Enterprise Research Centre, Cardiff Business School, United Kingdom.

[8] Liaee, M.M., Emmons, H. (1997) Scheduling families of jobs with setup times, *International Journal of Production Economics* **51** 165-176.

[9] Meunier, F., Sebo, A. (2008) Paintshop, odd cycles and necklace splitting, *Discrete Applied Mathematics*, article in press.

[10] Meyr, H. (2004) Supply chain planning in the German automotive industry, *OR Spectrum* **26** 447-470.

[11] Mitrofanov, S.P. (1966) Scientific principles of group technology, translated by Harris, E., Yorkshire, National Lending Library.

[12] Opitz, H. (1970) A classification system to describe workpieces: Parts I and II, Pergamon, Oxford.

[13] Pawlak, G., Lichocki, P., Plaza, M. (2006) On-line algorithms for minimizing the number of colour changes in a car paint shop buffer. In In: *PMS 2006, Tenth International Workshop on Project Management and Scheduling, April 26-28, 2006, Poznan, Poland. Abstracts*, Wydawnictwo Nakom, Poznan, 275-280.

[14] Pawlak, G., Rucinski, M. (2006) On-line algorithm for minimizing the makespan in the re-entrant single machine system. In In: *PMS 2006, Tenth International Workshop on Project Management and Scheduling, April 26-28, 2006, Poznan, Poland. Abstracts*, Wydawnictwo Nakom, Poznan, 286-291.

[15] Petrov, V.A. (1966) Flowline group production planning, Business Publications, London.

[16] Potts, C.N., Van Wassenhove, L.N. (1992) Integrating scheduling with batching and lot-sizing: a review of algorithms and complexity, *Journal of the Operational Research Society* **43** 395-406.

[17] Santos, C. (1984) Batching and sequencing decisions under lead time considerations for single machine problems. MSc Thesis, Department of Management Sciences, University of Waterloo, Waterloo.

[18] Solnon, C., Cung, V.D., Nguyen, A., Artigues, C. (2008) The car sequencing problem: Overview of state-of-the-art methods and industrial case-study of the ROADEF2005 challenge problem, *European Journal of Operational Research* **191** 912-927.

[19] Spieckermann, S., Gutenschager, K., Voss, S. (2004) A sequential ordering problem in automotive paint shops, *International Journal on Production Research* **42** 1865-1878.

[20] Valckenaers, P., Van Brussel, H. (2003) Deadlock avoidance in flexible flow shops with loops, *Journal of Intelligent Manufacturing* **14** 137-144.

[21] Yoshida, T., Nakamura, N., Hitomi, K. (1973) A study of production scheduling: optimization of group scheduling on a single production stage, *Transactions of the Japanese Society of Mechanical Engineers* **39** 1993-2003.