Available online at www.sciencedirect.com

ScienceDirect

Mathematical and Computer Modelling 46 (2007) 1373–1387

MATHEMATICAL
AND
COMPUTER
MODELLING

www.elsevier.com/locate/mcm

Integer programming approach to reactive scheduling in make-to-order manufacturing[☆]

Tadeusz Sawik[*]

AGH University of Science & Technology, Department of Operations Research and Information Technology, Faculty of Management, Al.Mickiewicza 30, 30-059 Kraków, Poland

Received 2 June 2006; received in revised form 15 December 2006; accepted 10 January 2007

Abstract

New algorithms based on mixed integer programming formulations are proposed for reactive scheduling in a dynamic, make-to-order manufacturing environment. The problem objective is to update a long-term production schedule subject to service level and inventory constraints, whenever the customer orders are modified or new orders arrive. Different rescheduling policies are proposed, from a total reschedule of all remaining and unmodified customer orders to a non-reschedule of all such orders. In addition, a medium restrictive policy is considered for rescheduling only a subset of remaining customer orders awaiting material supplies. Numerical examples modeled after a real-world scheduling/rescheduling of customer orders in the electronics industry are presented and some results of computational experiments are reported.

Keywords: Production scheduling; Dynamic rescheduling; Make-to-order environment; Integer programming

---

1. Introduction

In make-to-order manufacturing the performance of production planning and scheduling is evaluated by customer satisfaction and production costs. A typical measure of the customer satisfaction is customer service level, i.e., the fraction of customer orders filled on or before their due dates; see e.g. [1,2]. On the other hand, to achieve low unit production cost, both the input inventory of purchased materials waiting for processing in the system and the output inventory of finished products waiting for delivery to the customers should be minimized.

To reduce the required input inventory of purchased materials, the materials should be delivered as late as possible, i.e., the order earliness should be as small as possible. On the other hand the smaller the earliness of customer orders, the smaller is the output inventory of finished products completed before customer required shipping dates and waiting for delivery to the customers. However, if for some customer orders the earliness is smaller than the minimum earliness, i.e., ready periods and due dates are closer to each other, then reallocation of orders to the earlier

---

[☆] The author is grateful to two reviewers for providing comments which helped to improve this paper. This work has been partially supported by KBN research grant # 3 T11F 010 28 and AGH grant # 10.10.200.164 (Poland) and by Motorola (USA).

[*] Tel.: +48 12 617 39 92; fax: +48 12 617 39 84.
E-mail address: ghsawik@cyf-kr.edu.pl.

0895-7177/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mcm.2007.01.010

periods with surplus of capacity is restricted due to later material availability. As a result, the number of tardy orders may increase, or some orders may even remain unscheduled during the planning horizon.

A make-to-order manufacturing environment is dynamic in nature and the customers may modify, cancel or add orders during the planning horizon. As a result, a predetermined production schedule may become inefficient and may need to be revised in reaction to unexpected changes of customer orders. In practice, reactive scheduling algorithms are applied for a dynamic rescheduling. A review of the literature (e.g. [3]) indicates that research on reactive scheduling is mostly focused on heuristic approaches such as genetic algorithms [4] or various AI techniques [5,6].

In the literature on production planning and scheduling, integer programming models have been widely used; see e.g. [2,7]. In industrial practice, however, the application of integer programming for production scheduling is limited, in particular in make-to-order manufacturing; see e.g. [8–11]. Integer programming approaches for reactive scheduling have been mainly used in the process industries; see e.g. [12,13]. An exception is the work [9], where a scheduling tool with rescheduling capabilities based on an integer programming formulation is presented for the semiconductor industry. However, the model is solved by an approximate technique and optimal solution was not attempted.

The purpose of this paper is to present new algorithms, based on integer programming formulations, for reactive scheduling in a dynamic, make-to-order manufacturing, where customers may modify or cancel their orders or place new orders during the planning horizon. In the algorithms, different rescheduling policies are proposed, from a total reschedule of all remaining and unmodified customer orders to a non-reschedule of all such orders. In addition, a medium restrictive policy is considered for rescheduling only a subset of remaining customers orders awaiting material supplies. The scheduling objective is to dynamically assign/reassign customer orders with various due dates to planning periods with limited processing capacities, to minimize the number of tardy orders and the total input and output inventory over a planning horizon. No other algorithm capable of solving to optimality the above problem exists in the literature.

The paper is organized as follows. In the next section the description of make-to-order production scheduling in a flexible flowshop is provided. The basic integer programming formulations for production scheduling/rescheduling are presented in Section 3. Rescheduling algorithms based on the proposed mixed integer programming models are described in Section 4 and some formulae for the calculation of the input and output inventory are derived in Section 5. Numerical examples modeled after a real-world, make-to-order electronics manufacturing and some computational results are provided in Section 6. Conclusions are made in the last section.

2. Problem description

The production system under study is a flexible flowshop that consists of m processing stages in series, and each stage $i \in I = \{1, \dots, m\}$ is made up of $m_i \geq 1$ identical, parallel machines. In the system various types of products are produced in a make-to-order environment responding directly to customer orders. Let J be the set customer orders that are known ahead of a planning horizon. Each order $j \in J$ is described by a triple (a_j, d_j, s_j), where a_j is the order arrival date (e.g. the earliest period of material availability), d_j is the customer due date (e.g. customer required shipping date), and s_j is the size of the order (the quantity of ordered products of specified type). Each order requires processing in various processing stages; however, some orders may bypass some stages. Let $p_{ij} \geq 0$ be the processing time in stage i of each product in order $j \in J$, i.e., for each stage i, p_{ij} depends on the type of ordered product. The orders are processed and transferred among the stages in lots of various size that depends on the ordered product type; let b_j be the size of the production lot for order j.

The planning horizon consists of h planning periods (e.g. working days). Let $T = \{1, \dots h\}$ be the set of planning periods and c_{it} the processing time available in period t on each machine in stage i.

The following two types of the customer orders are considered:

1. Small-size (single-period) orders, where each order can be fully processed in a single time period, e.g. during one day. The single-period orders are referred to as indivisible orders.
2. Large-size (multi-period) orders, where each order cannot be completed in one period and must be split and processed in more than one time period. The multi-period orders are referred to as divisible orders.

For divisible orders $s_j \geq b_j$.

In practice, two types of customer orders are simultaneously scheduled. Denote by $J1 \subseteq J$, and $J2 \subseteq J$, respectively, the subset of indivisible, and divisible orders, where $J1 \bigcup J2 = J$, and $J1 \bigcap J2 = \emptyset$.

Table 1
Notation: Initial scheduling

Indices

i = processing stage, $i \in I = \{1, \ldots, m\}$
j = customer order, $j \in J = \{1, \ldots, n\}$
k = product type, $k \in K = \{1, \ldots, r\}$
t = planning period, $t \in T = \{1, \ldots, h\}$

Input parameters

a_j, d_j, s_j = arrival date, due date, size of order j
b_j = production lot size for order j
c_{it} = processing time available in period t on each machine in stage i
m_i = number of identical, parallel machines in stage i
n = number of customer orders to be scheduled
p_{ij} = processing time in stage i of each product in order j
$J1 \subset J$ = subset of small (single-period) customer orders
$J2 \subset J$ = subset of large (multi-period) customer orders
$J_k \subset J$ = subset of customer orders for product type k
$\overline{E}$ = upper limit on maximum earliness
$\overline{U}$ = upper limit on number of tardy orders

Decision variables

$u_j = 1$, if order j is completed after due date; otherwise $u_j = 0$ (unit penalty for tardy orders)
$x_{jt} = 1$, if order j is performed in period t; otherwise $x_{jt} = 0$ (order assignment variable)
$y_{jt} \geq 0$ = fraction of customer order j to be processed in period t (order allocation variable)
$E_{\max}$ = maximum earliness of orders
U_{sum} = number of tardy orders

It is assumed that each customer order $j \in J1$ must be fully completed in exactly one planning period, and each customer order $j \in J2$ must be completed in two consecutive planning periods. The latter assumption, however, can be easily relaxed [10] to allow for the completing of large orders in more than two consecutive periods.

A dynamic, make-to-order manufacturing environment is considered with a dynamic planning horizon used to successively update a production schedule when old, yet uncompleted customer orders are modified or new customer orders arrive during the horizon. The modifications of customer orders may include changes of order size, e.g. increase, decrease or cancellation, and/or changes of due dates, e.g. postponement of delivery date, occurring during the planning horizon. The horizon can be progressively shifted to take into account the order modifications.

The objective of the long-term reactive scheduling is to assign/reassign customer orders to planning periods over a planning horizon to maximize the customer service level with limited input and output inventory.

3. Mixed integer programs for reactive scheduling

In this section mixed integer programming formulations are proposed for customer order assignment/reassignment over a long-term planning horizon, to maximize service level, implicitly subject to the inventory constraints. Basic notation is presented in Table 1.

3.1. Basic model

The basic model used to update the current production schedule, whenever some customer orders are modified, is presented below. The extent of required changes in the current schedule depends on the applied policy (see Section 4) and the changes in size and due dates of the modified customer orders. The updated schedule takes into account the current input inventory that is implicitly considered in the model by the upper bound $\overline{E}$ on the maximum earliness $E_{\max}$ of customer orders.

Model MaxSL($\overline{E}$): *Customer order assignment to Maximize Service Level subject to maximum earliness constraints*

Maximize

$$1 - U_{\text{sum}}/n \tag{1}$$

or

Minimize

$$U_{\text{sum}} = \sum_{j \in J} u_j \tag{2}$$

subject to

1. *Order assignment constraints*
   - each indivisible customer order is assigned to exactly one planning period,
   $$\sum_{t \in T: t \geq a_j} x_{jt} = 1; \quad j \in J1 \tag{3}$$
   - each divisible customer order is assigned to at most two consecutive planning periods,
   $$x_{jt} + x_{jt+1} \leq 2; \quad j \in J2, t \in T : a_j \leq t \leq h-1 \tag{4}$$
   $$x_{jt} + x_{j\tilde{t}} \leq 1; \quad j \in J2, t \in T, \tilde{t} \in T : a_j \leq t \leq h-2, \tilde{t} \geq t+2. \tag{5}$$
2. *Order allocation constraints*
   - each order must be completed,
   $$\sum_{t \in T: t \geq a_j} y_{jt} = 1; \ \ j \in J \tag{6}$$
   - each indivisible order is completed in a single period,
   $$x_{jt} = y_{jt}; \quad j \in J1, t \in T : t \geq a_j \tag{7}$$
   - each divisible order is allocated among all the periods that are selected for its assignment,
   $$x_{jt} \geq y_{jt}; \quad j \in J2, t \in T : t \geq a_j \tag{8}$$
   - the minimum portion of a divisible order allotted to one period is not less than the batch size,
   $$y_{jt} \geq b_j x_{jt}/s_j; \quad j \in J2, t \in T : t \geq a_j. \tag{9}$$
3. *Tardy order constraints*
   - an indivisible tardy order is completed after its due date,
   $$u_j = \sum_{t \in T: t > d_j} x_{jt}; \quad j \in J1 \tag{10}$$
   - a divisible tardy order is partly assigned after its due date,
   $$u_j \geq \sum_{t \in T: t > d_j} y_{jt}; \quad j \in J2 \tag{11}$$
   $$u_j \leq \sum_{t \in T: t > d_j} x_{jt}; \quad j \in J2. \tag{12}$$
4. *Capacity constraints*
   - in every period the demand on capacity at each processing stage cannot be greater than the maximum available capacity in this period,
   $$\sum_{j \in J_i} p_{ij} s_j y_{jt} \leq c_{it} m_i; \quad i \in I, t \in T. \tag{13}$$
5. *Maximum earliness constraints*
   - for each early order j assigned to period $t < d_j$, its earliness $(d_j - t)$ cannot exceed the maximum earliness $\overline{E}$
   $$(d_j - t) x_{jt} \leq \overline{E}; \quad j \in J, t \in T : t \geq a_j. \tag{14}$$
6. *Variable nonnegativity and integrality constraints*
   $$u_j \in \{0, 1\}; \quad j \in J \tag{15}$$
   $$x_{jt} \in \{0, 1\}; \quad j \in J, t \in T : t \geq a_j \tag{16}$$
   $$0 \leq y_{jt} \leq 1; \quad j \in J, t \in T : t \geq a_j. \tag{17}$$

The objective function represents the customer service level, i.e., the fraction of non-delayed customer orders to be maximized (1), or equivalently the number of tardy orders to be minimized (2). The solution to MaxSL($\overline{E}$) determines the assignment of indivisible customer orders to single planning periods and the allocation of divisible orders among the pairs of consecutive planning periods such that the customer service level is maximized subject to the limited maximum earliness of orders and by this limited total input and output inventory level.

Model $\mathrm{MaxSL}(\overline{E})$ can be briefly rewritten as follows:

$$\mathrm{MaxSL}(\overline{E}) = \max\{(1)\colon (2)\text{–}(17)\}. \tag{18}$$

3.2. Models for initial scheduling

The beginning production schedule for the original customer orders known ahead of the planning horizon is determined by solving the following sequence of two mixed integer programs.

1. *Model* MaxSL: *Customer order assignment to Maximize Service Level*

   $$\mathrm{MaxSL} = \max\{(1)\colon (2)\text{–}(13), (15)\text{–}(17)\} \tag{19}$$

   where all materials are assumed to be available at the beginning, i.e. $a_j = 1$ for each order $j \in J$.
2. *Model* $\mathrm{MinME}(\overline{U})$: *Customer order assignment to Minimize Maximum Earliness subject to service level constraints*

   $$\mathrm{MinME}(\overline{U}) = \min\{E_{\max} : (2)\text{–}(13), (15)\text{–}(17), U_{\mathrm{sum}} \le \overline{U}, (d_j - t)x_{jt} \le E_{\max};\ j \in J, t \in T : t \ge a_j\} \tag{20}$$

   where $1 - \overline{U}/n$ is the solution value of (19).

   The objective function of (20) implicitly limits the maximum level of the total input and output inventory over the planning horizon.

   It is proved in [14] that for a given number of tardy orders (i.e. the required service level), the total inventory increases with $E_{\max}$, i.e., both the input inventory of product-specific materials and the output inventory of finished products can be reduced when ready periods and due dates of customer orders are closer. Therefore, the maximum level of the total input and output inventory can be implicitly minimized by minimizing the maximum earliness $E_{\max}$ of early orders, given the minimum number U_{sum} of tardy orders.

In the above sequential decision-making framework, first the solution to MaxSL determines the maximum service level. Then, the minimum value $E^*_{\max}$ of the maximum earliness $E_{\max}$ is found using model $\mathrm{MinME}(\overline{U})$ to implicitly limit the total inventory, subject to service level constraints. The solution to $\mathrm{MinME}(\overline{U})$ determines the optimal allocation $\{x^*_{jt}, y^*_{jt}\}$ of customer orders among planning periods.

4. Rescheduling algorithms

In this section different rescheduling algorithms based on the mixed integer programming models are proposed.

Let t_{mod} be the first planning period immediately after the order modification. It is assumed that the customer orders completed before t_{mod} or with due dates smaller than t_{mod} cannot be modified. In practice different rescheduling policies can be applied, from a total reschedule of all remaining customer orders, i.e. reschedule of all unmodified orders that have been assigned to periods not less than t_{mod} (algorithm REALL), to a non-reschedule of all such orders (algorithm RENON).

In addition to the above two extreme rescheduling policies a medium restrictive algorithm REMAT is proposed for rescheduling of the remaining customers orders awaiting material supplies. For each order j, product-specific materials are assumed to be unavailable earlier than $E^*_{\max}$ periods ahead of the order due date d_j. Therefore, each order j cannot be assigned to periods earlier than $d_j - E^*_{\max}$. In particular, in period t_{mod} product-specific materials are not available for orders due in periods greater than $t_{\mathrm{mod}} + E^*_{\max}$, and hence all such orders can be rescheduled. On the other hand, the unmodified orders with product-specific materials supplied by period t_{mod} (i.e., orders assigned to periods $t = t_{\mathrm{mod}}, \ldots, t_{\mathrm{mod}} + E^*_{\max}$) are considered non-reschedulable in algorithm REMAT.

In all these algorithms the planning horizon is progressively shifted to take into account modifications of the customer orders (changes of order size and/or due date) occurring during the horizon. Table 2 presents the notation used in the rescheduling algorithms.

In all algorithms, first the set J_{old} of orders remaining for completion without modification is split into two disjoint subsets: J^S_{old} of reschedulable orders and J^N_{old} of fixed, non-reschedulable orders for which the assignment to planning

Table 2
Notation: Rescheduling

Input parameters

$h' =$ new planning horizon
$t_{\text{mod}} =$ the planning period immediately following modification of orders
$J_{\text{mod}} =$ set of modified orders
$J_{\text{old}} =$ subset of orders in J remaining for completion without modification
$J^N_{\text{old}}, J^S_{\text{old}} =$ subset of orders in J_{old}, respectively non-reschedulable, reschedulable
$T_{\text{new}} = \{h+1, \ldots, h'\} =$ set of new planning periods
$T_{\text{old}} = \{t_{\text{mod}}, \ldots, h\} =$ subset of remaining planning periods in T
$T^N_{\text{old}} =$ subset of periods in T_{old} with fixed assignment of orders in J_{old}
$J' = J_{\text{mod}} \bigcup J_{\text{old}} =$ updated set of orders
$T' = T_{\text{old}} \bigcup T_{\text{new}} =$ updated set of planning periods

Prime ($'$) denotes updated parameters after modification of orders.

periods cannot be changed. For example, in algorithm REMAT, the subset of non-reschedulable orders contains orders in J_{old} remaining for completion, such that have been assigned to periods in the subset $T^N_{\text{old}} = \{t_{\text{mod}}, \ldots, t_{\text{mod}} + E^*_{\max}\}$ of remaining periods in $T_{\text{old}} = \{t_{\text{mod}}, \ldots, h\}$.

In what follows, we denote by prime ($'$) the updated values of some parameters and decision variables after each modification of orders. For example s'_j denotes the modified size of customer order $j \in J' = J_{\text{old}} \bigcup J_{\text{mod}}$, where $s'_j = s_j,\ j \in J_{\text{old}}$ and $s'_j \neq s_j,\ j \in J_{\text{mod}}$.

Algorithm REALL

Step 0. Split the set J_{old} of orders remaining for completion into two disjoint subsets: J^S_{old} of reschedulable orders and J^N_{old} of fixed, non-reschedulable orders.

$$J^N_{\text{old}} = \emptyset \tag{21}$$

$$J^S_{\text{old}} = J_{\text{old}}. \tag{22}$$

Step 1. Determine the new planning horizon h' for the updated set J' of customer orders.

$$h_1 = \min\left\{ h1 : \max_{i \in I} \left(\frac{\sum\limits_{j \in J'} p_{ij} s'_j}{m_i \sum\limits_{t=t_{\text{mod}}}^{h1} c_{it}} \right) \leq 1 \right\} \tag{23}$$

$$h_2 = \max_{j \in J'}(d_j). \tag{24}$$

If $\max\{h_1, h_2\} \leq h$, then set $h' = h$.
Otherwise set $h' = \max\{h_1, h_2\}$, $T_{\text{new}} = \{h+1, \ldots, h'\}$ and $T' = T_{\text{old}} \bigcup T_{\text{new}}$.

Step 2. Do not change the assignment in period t_{mod} of partially completed, two-period orders in $J2$, i.e.,

$$y'_{j,t_{\text{mod}}} = y_{j,t_{\text{mod}}}, \quad j \in J2 : x_{j,t_{\text{mod}}-1} = 1. \tag{25}$$

Step 3. Solve MaxSL($\overline{E}$), (18) for $\overline{E} = E^*_{\max}$ and subject to fixed assignment constraints from Step 2, to find a new schedule for the updated set J' of customer orders, updated set of planning periods T' and updated material availability periods

$$a'_j = \begin{cases} \max\{1, d_j - E^*_{\max}\} & \text{if } j \in J_{\text{old}} \\ \max\{t_{\text{mod}}, d_j - E^*_{\max}\} & \text{if } j \in J_{\text{mod}}. \end{cases} \tag{26}$$

In the algorithms REMAT and RENON presented below, Step 1 and Step 3 are identical to the corresponding steps of REALL.

Algorithm REMAT

Step 0. Split the set J_{old} of orders remaining for completion into two disjoint subsets: J^S_{old} of reschedulable orders and J^N_{old} of fixed, non-reschedulable orders.

$$J^N_{\text{old}} = \left\{ j \in J_{\text{old}} : \sum_{t_{\text{mod}} \le t \le t_{\text{mod}} + E^*_{\max}} x_{jt} = 1 \right\} \tag{27}$$

$$J^S_{\text{old}} = J_{\text{old}} \setminus J^N_{\text{old}}. \tag{28}$$

Set $T^N_{\text{old}} = \{t_{\text{mod}}, \ldots, t_{\text{mod}} + E^*_{\max}\}$.

Step 2. Do not change the assignment of non-reschedulable orders $j \in J^N_{\text{old}}$, i.e.,

$$y'_{jt} = y_{jt}, \quad j \in J^N_{\text{old}}, t \in T^N_{\text{old}} \tag{29}$$

$$y'_{j,t_{\text{mod}}+E^*_{\max}+1} = y_{j,t_{\text{mod}}+E^*_{\max}+1}, \quad j \in J^N_{\text{old}} \bigcap J2 : x_{j,t_{\text{mod}}+E^*_{\max}} = 1. \tag{30}$$

Algorithm RENON

Step 0. Split the set J_{old} of orders remaining for completion into two disjoint subsets: J^S_{old} of reschedulable orders and J^N_{old} of fixed, non-reschedulable orders.

$$J^N_{\text{old}} = J_{\text{old}} \tag{31}$$

$$J^S_{\text{old}} = \emptyset. \tag{32}$$

Step 2. Do not change the assignment of all orders in J_{old}, i.e.,

$$y'_{jt} = y_{jt}, \quad j \in J_{\text{old}}, t \in T_{\text{old}}. \tag{33}$$

The above MIP-based rescheduling algorithms differ in the extent of imposed schedule changes. In practice, the selection of a particular algorithm may depend on the impact of disruptions induced by rescheduling policy on the predetermined schedule.

5. Input and output inventory

In this section some formulae are derived for calculating the input inventory of raw materials and the output inventory of finished products. The input inventory of product-specific raw materials only is considered with no common materials for different product types taken into account. Furthermore, to make the calculations clearer it is assumed that each product requires one unit of the corresponding product-specific material (e.g. one printed wiring board of a specific design per electronic device of the corresponding type). As a result, for each order j the required quantity of product-specific material equals the quantity of the ordered products s_j.

The original amount of product-specific materials required for customer orders $j \in J_{\text{mod}}$ such that $d_j - E^*_{\max} < t_{\text{mod}} \le d_j$ and supplied before t_{mod} differs from the modified amount of those materials required after the order modification. As a result, the actual input inventory level in period t_{mod} may be higher or lower than the required level. For each product type $k \in K$, the shortage ($\varDelta\text{INP}_k < 0$) or surplus ($\varDelta\text{INP}_k > 0$) of product-specific material inventory in period $t_{\text{mod}} - 1$ with respect to the amount required for the modified orders $j \in J_{\text{mod}}$ is

$$\varDelta\text{INP}_k = \sum_{j \in J_k \bigcap J_{\text{mod}} : d_j - E^*_{\max} < t_{\text{mod}} \le d_j} (s'_j - s_j); \quad k \in K. \tag{34}$$

It is assumed that the shortage or the surplus of product-specific materials is balanced with higher or lower supplies in period t_{mod}, respectively.

The input inventory $\text{INP}(t)$ of product-specific materials can be calculated as below.

$$\begin{aligned}\text{INP}(t) = \text{INP}(t_{\text{mod}} - 1) &+ \sum_{j \in J' : t_{\text{mod}} \le a'_j \le t} s'_j - \sum_{j \in J_{\text{mod}} : d_j - E^*_{\max} < t_{\text{mod}} \le d_j} s_j \\ &- \sum_{j \in J', \tau \in T' : a'_j \le \tau \le t} s'_j y'_{j\tau}; \quad t \in T' : t \ge t_{\text{mod}}\end{aligned} \tag{35}$$

where $\text{INP}(t_{\text{mod}} - 1)$ is the input inventory remaining in period $t_{\text{mod}} - 1$.

In (35), the input inventory $\mathrm{INP}(t)$ in each period t is calculated as the difference between the amount of product-specific materials supplied by period t and the amount of these materials processed into finished products by this period. The first summation term with negative sign in the right-hand side of (35) balances in period t_{mod} the shortage or the surplus of product-specific materials supplied by period $t_{\mathrm{mod}} - 1$.

Similarly, the output inventory $\mathrm{OUP}(t)$ of finished products can be expressed as below.

$$\mathrm{OUP}(t) = \sum_{d>t} \mathrm{OUP}_d(t_{\mathrm{mod}} - 1) + \sum_{j \in J', \tau \in T' : a'_j \le \tau \le t < d_j} s'_j y'_{j\tau}; \quad t \in T' : t \ge t_{\mathrm{mod}} \tag{36}$$

where $\mathrm{OUP}_d(t_{\mathrm{mod}} - 1)$ is the output inventory of finished products remaining in period $t_{\mathrm{mod}} - 1$, due in period $d > t_{\mathrm{mod}}$.

In (36), the output inventory $\mathrm{OUP}(t)$ in each period t is calculated as the amount of finished products processed by period t before the customer required shipping dates.

The total inventory $\mathrm{TOT}(t) = \mathrm{INP}(t) + \mathrm{OUP}(t)$ in each period t can be found by summing the corresponding right-hand sides of (35) and (36). In particular, the total input and output inventory $\mathrm{TOT}(t_{\mathrm{mod}} - 1)$ in the last period of the previous schedule can be expressed as below.

$$\mathrm{TOT}(t_{\mathrm{mod}} - 1) = \sum_{j \in J : d_j \le t_{\mathrm{mod}} - 1} s_j \left(1 - \sum_{a_j \le \tau \le t_{\mathrm{mod}} - 1} y_{j\tau} \right) + \sum_{j \in J : t_{\mathrm{mod}} \le d_j \le t_{\mathrm{mod}} - 1 + E^*_{\max}} s_j. \tag{37}$$

The first summation term in (37) is the inventory of product-specific materials for customer orders due by period $t_{\mathrm{mod}} - 1$, and the second term is the inventory of product-specific materials and finished products of customer orders due after period $t_{\mathrm{mod}} - 1$, respectively waiting for processing in the system and for shipping to customers. The first term represents the input inventory in period $t_{\mathrm{mod}} - 1$ of product-specific materials for tardy orders and is greater than zero only if some customer orders are tardy; otherwise this term is equal to zero. The second term increases with the maximum earliness $E^*_{\max}$. Given the tardy orders, the total inventory in $t_{\mathrm{mod}} - 1$ increases with $E^*_{\max}$, i.e. both the input inventory of product-specific materials and the output inventory of finished products can be reduced when ready periods and due dates of customer orders are closer; see [14].

6. Computational experiments

In this section numerical examples and some computational results are presented to illustrate possible applications of the proposed algorithms for reactive scheduling, based on the mixed integer programming formulations. The examples are modeled after a real-world distribution center for high-tech products, where finished products are assembled for shipping to customers.

The distribution center can be modeled as a flexible flowshop made up of six processing stages with parallel machines. In the distribution center ten product types of three product groups are assembled. The processing stages are the following: material preparation stage, where all materials required for assembly of each product are prepared; postponement stage, where products for some orders are customized; three flashing/flexing stages in parallel, one for each group of products, where required software is downloaded; and a packing stage, where products and required accessories are packed for shipping.

Customer orders require processing in at most four stages: material preparation stage, postponement stage, one flashing/flexing stage, and packing stage (see Fig. 1). However, some orders do not need postponement.

Customer orders are split into production lots of fixed sizes, each to be processed as a separate job. Each large-size (multi-period) customer order must be completed in at most two planning periods (two days).

A brief description of the production system, production process, products and the beginning customer orders is given below.

1. Production system
    - six processing stages: 10 parallel machines in each stage $i = 1, 2$; 20 parallel machines in each stage $i = 3, 4, 5$; and 10 parallel machines in stage $i = 6$.
2. Products
    - 10 product types of three product groups, each to be processed on a separate group of flashing/flexing machines,

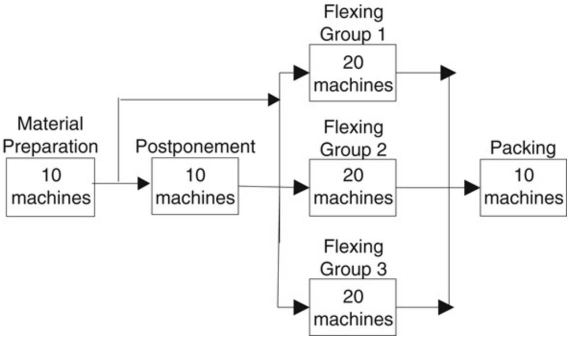

Fig. 1. Distribution center: A flexible flowshop.

Table 3
Computational results for scenario I

Model/t_{mod}	Var.	Bin.	Cons.	Nonz.	Solution values[a]	CPU[b]
Initial scheduling for beginning demand						
Max SL	29 310	14 656	18 198	133 276	$U^*_{\text{sum}} = 0$	3.60
Min ME(0)	31 507	15 753	33 057	148 946	$E^*_{\max} = 6$	387.88
REALL						
MaxSL(6)/$t_{\text{mod}} = 6$	22 522	11 565	19 779	316 229	$U^*_{\text{sum}} = 2, h' = 31$	20.61
MaxSL(6)/$t_{\text{mod}} = 14$	15 558	8013	13 116	195 823	$U^*_{\text{sum}} = 3, h' = 32$	28.24
MaxSL(6)/$t_{\text{mod}} = 24$	4059	2112	3928	40 811	$U^*_{\text{sum}} = 5, h' = 33$	1.22
REMAT						
MaxSL(6)/$t_{\text{mod}} = 6$	15 817	8145	11 237	186 527	$U^*_{\text{sum}} = 3, h' = 31$	9.07
MaxSL(6)/$t_{\text{mod}} = 14$	7656	3959	5610	77 634	$U^*_{\text{sum}} = 6, h' = 32$	1.69
MaxSL(6)/$t_{\text{mod}} = 24$	222	105	373	1898	$U^*_{\text{sum}} = 8, h' = 33$	0.02
RENON						
MaxSL(6)/$t_{\text{mod}} = 6$	371	152	1281	6121	$U^*_{\text{sum}} = 8, h' = 35$	0.04
MaxSL(6)/$t_{\text{mod}} = 14$	537	248	1625	8479	$U^*_{\text{sum}} = 10, h' = 35$	0.09
MaxSL(6)/$t_{\text{mod}} = 24$	382	184	759	4344	$U^*_{\text{sum}} = 11, h' = 36$	0.09
Ex post scheduling for updated demand						
MaxSL(6)	32 457	16 592	26 199	527 271	$U^*_{\text{sum}} = 2, h' = 32$	80.40

[a] U^*_{sum} — number of tardy orders, $E^*_{\max}$ — maximum earliness, h' — planning horizon.
[b] CPU seconds for proving optimality on a PC with a Pentium IV processor, 1.8 GHz, RAM 1 GB/CPLEX v.9.1.

- the beginning demand is made up of 100 customer orders, each consisting of several suborders (customer required shipping volumes). The total number of suborders is 816, and the beginning total demand for all products is 537 760.
- production (and transfer) lot sizes: 200, 200, 300, 100, 100, 100, 200, 200, 300, 100, respectively for product type 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

3. Processing times (in seconds) for product types:

product type/stage	1	2	3	4	5	6
1	20	0	120	0	0	15
2	20	0	140	0	0	15
3	10	0	160	0	0	10
4	15	5	0	120	0	15
5	15	10	0	140	0	15
6	10	5	0	160	0	10
7	15	10	0	180	0	15
8	20	5	0	0	120	15
9	15	0	0	0	140	10
10	15	0	0	0	160	10

Table 4
Computational results for scenario II

Model/t_{mod}	Var.	Bin.	Cons.	Nonz.	Solution values[a]	CPU[b]
Initial scheduling for beginning demand						
MaxSL	29 310	14 656	18 198	133 276	$U^*_{\mathrm{sum}} = 0$	3.60
Min ME(0)	31 507	15 753	33 57	148 946	$E^*_{\max} = 6$	387.88
REALL						
MaxSL(6)/$t_{\mathrm{mod}} = 6$	20 637	10 579	17 555	277 344	$U^*_{\mathrm{sum}} = 0, h' = 30$	0.79
MaxSL(6)/$t_{\mathrm{mod}} = 14$	13 034	6726	12 069	151 584	$U^*_{\mathrm{sum}} = 0, h' = 30$	2.28
MaxSL(6)/$t_{\mathrm{mod}} = 24$	2906	1520	3011	24 399	$U^*_{\mathrm{sum}} = 0, h' = 30$	0.09
REMAT						
MaxSL(6)/$t_{\mathrm{mod}} = 6$	14 338	7386	9264	15 9004	$U^*_{\mathrm{sum}} = 0, h' = 30$	0.64
MaxSL(6)/$t_{\mathrm{mod}} = 14$	5898	3094	4305	54 439	$U^*_{\mathrm{sum}} = 3, h' = 30$	0.22
MaxSL(6)/$t_{\mathrm{mod}} = 24$	64	25	92	382	$U^*_{\mathrm{sum}} = 3, h' = 30$	0.01
RENON						
MaxSL(6)/$t_{\mathrm{mod}} = 6$	76	22	75	327	$U^*_{\mathrm{sum}} = 1, h' = 30$	0.01
MaxSL(6)/$t_{\mathrm{mod}} = 14$	297	130	615	3624	$U^*_{\mathrm{sum}} = 5, h' = 32$	0.02
MaxSL(6)/$t_{\mathrm{mod}} = 24$	111	50	153	835	$U^*_{\mathrm{sum}} = 5, h' = 32$	0.01
Ex post scheduling for updated demand						
MaxSL(6)	28 612	14 644	23 028	438 409	$U^*_{\mathrm{sum}} = 0, h' = 30$	1.60

[a] U^*_{sum} — number of tardy orders, $E^*_{\max}$ — maximum earliness, h' — planning horizon.
[b] CPU seconds for proving optimality on a PC with a Pentium IV processor, 1.8 GHz, RAM 1 GB/CPLEX v.9.1.

Table 5
Computational results for scenario III

Model/t_{mod}	Var.	Bin.	Cons.	Nonz.	Solution values[a]	CPU[b]
REALL						
MaxSL(6)/$t_{\mathrm{mod}} = 24$	2798	1461	2914	23 512	$U^*_{\mathrm{sum}} = 0, h' = 30$	0.06
REMAT						
MaxSL(6)/$t_{\mathrm{mod}} = 24$	39	12	41	159	$U^*_{\mathrm{sum}} = 3, h' = 30$	0.01
RENON						
MaxSL(6)/$t_{\mathrm{mod}} = 24$	103	45	137	700	$U^*_{\mathrm{sum}} = 5, h' = 32$	0.01
Ex post scheduling for updated demand						
MaxSL(6)	28 654	14 665	23 073	438 760	$U^*_{\mathrm{sum}} = 0, h' = 30$	3.40

[a] U^*_{sum} — number of tardy orders, $E^*_{\max}$ — maximum earliness, h' — planning horizon.
[b] CPU seconds for proving optimality on a PC with a Pentium IV processor, 1.8 GHz, RAM 1 GB/CPLEX v.9.1.

4. Planning horizon: $h = 30$ days, each of length $L = 2 \times 9$ h.

Notice that the suborders in the computational examples play the role of orders in the mathematical formulation. Now, the problem objective is to assign/reassign customer suborders over the planning horizon to minimize the number of tardy suborders as a measure of the customer service level subject to maximum earliness constraints to limit the total inventory level.

In the computational experiments the following three scenarios with three modifications of customer orders during the planning horizon are considered:

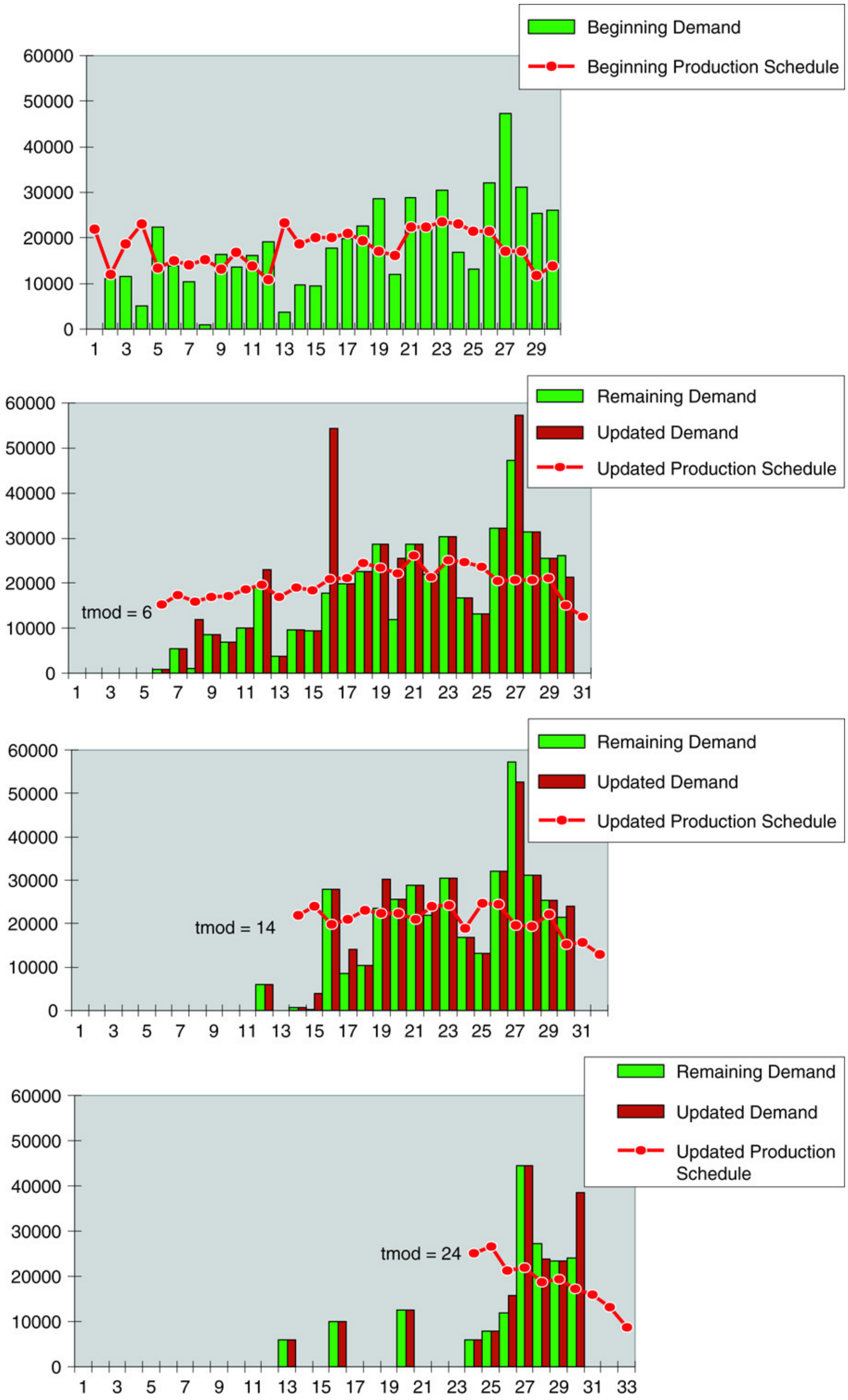

Fig. 2. Distribution of demand and aggregate production for algorithm REALL: scenario I.

1. Scenario I with total demand increased by 101 110 products
    - 13 customer orders due in periods 8–30 are modified in period $t_{\text{mod}} = 6$. The resulting total change of demand is +70 200 products.

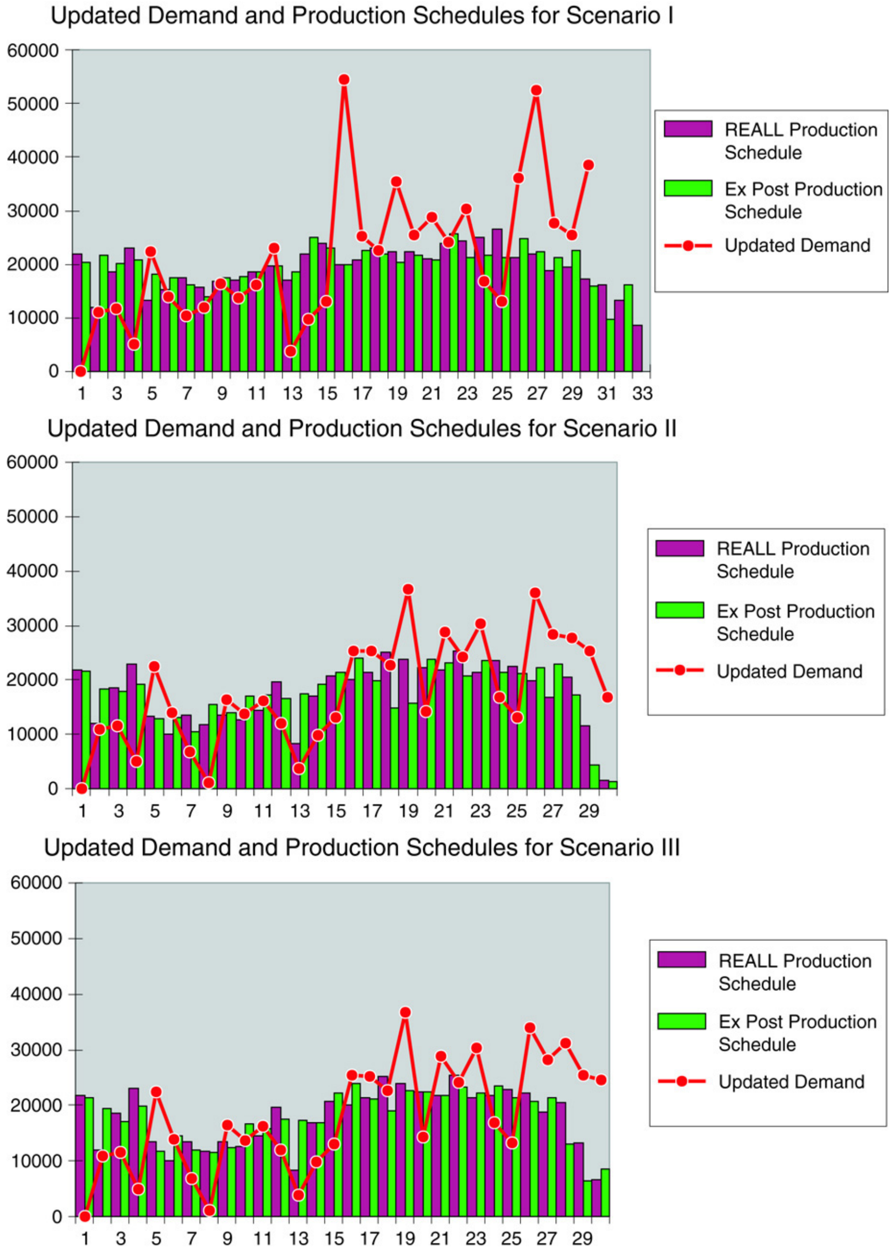

Fig. 3. Production schedules for updated demand.

- 13 customer orders due in periods 15–30 are modified in period $t_{\text{mod}} = 14$. The resulting total change of demand is +15 950 products.
- 8 customer orders due in periods 26–30 are modified in period $t_{\text{mod}} = 24$. The resulting total change of demand is +14 960 products.

2. Scenario II with total demand decreased by 9330 products
   - 11 customer orders due in periods 7–30 are modified in period $t_{\text{mod}} = 6$. The resulting total change of demand is −32 395 products.
   - 14 customer orders due in periods 15–30 are modified in period $t_{\text{mod}} = 14$. The resulting total change of demand is +29 880 products.

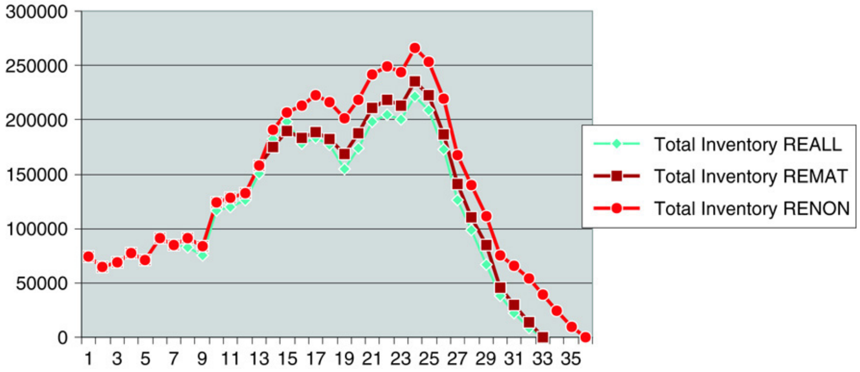

Fig. 4. Total input and output inventory: scenario I.

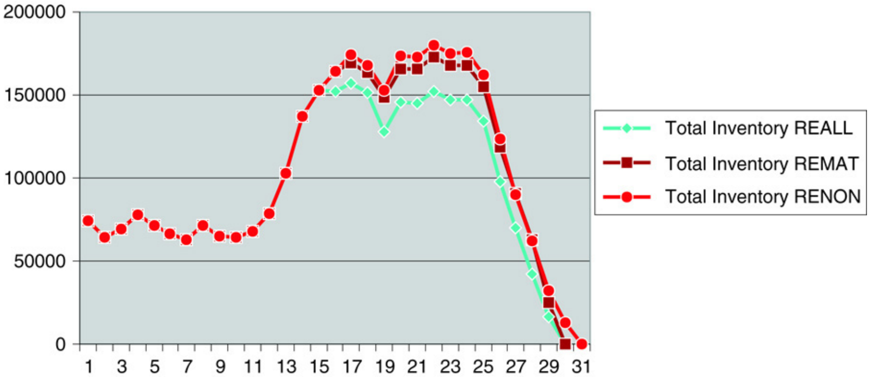

Fig. 5. Total input and output inventory: scenario II.

- 7 customer orders due in periods 26–30 are modified in period $t_{\mathrm{mod}} = 24$. The resulting total change of demand is -6815 products.

3. Scenario III with total demand unchanged
   - Modifications of customer orders in period $t_{\mathrm{mod}} = 6$ are the same as in the scenario with decreasing demand.
   - Modifications of customer orders in period $t_{\mathrm{mod}} = 14$ are the same as in the scenario with decreasing demand.
   - 3 customer orders due in periods 26–30 are modified in period $t_{\mathrm{mod}} = 24$. The resulting total change of demand is $+2515$ products.

The computational experiments are summarized in Tables 3–5, respectively for scenarios with increasing, decreasing and unchanged total demand. The tables present solution results and the characteristics of integer programs MaxSL, MinME($\overline{U}$) (with $\overline{U} = 0$) for the initial scheduling of the beginning demand, MaxSL($\overline{E}$) (with $\overline{E} = 6$) for the three rescheduling algorithms REALL, REMAT and RENON, and for the ex post scheduling (determined after all modifications are known) of the updated demand. The size of each integer program is represented by the total number of variables, Var., number of binary variables, Bin., number of constraints, Cons., and number of nonzero elements in the constraint matrix, Nonz. The last two columns of the table present the optimal solution values of U_{sum} for MaxSL, $E_{\max}$ for MinME($\overline{U}$), U_{sum}, h' for MaxSL($\overline{E}$), and CPU time in seconds required to find the proven optimal solution. The computational experiments were performed using the AMPL programming language and the CPLEX v.9.1 solver on a laptop with a Pentium IV processor at 1.8 GHz and 1 GB RAM.

Since scenarios II and III are identical for planning periods $t < t_{\mathrm{mod}} = 24$, Table 5 presents solution results only for the remaining periods $t \geq t_{\mathrm{mod}}$.

Tables 3–5 indicate that the best results (the minimum number of tardy orders over the planning horizon and the smallest horizon length) are obtained for algorithm REALL, where total reschedule of all remaining customer orders is applied each time some orders are modified. In contrast, algorithm RENON, where the assignment of all remaining

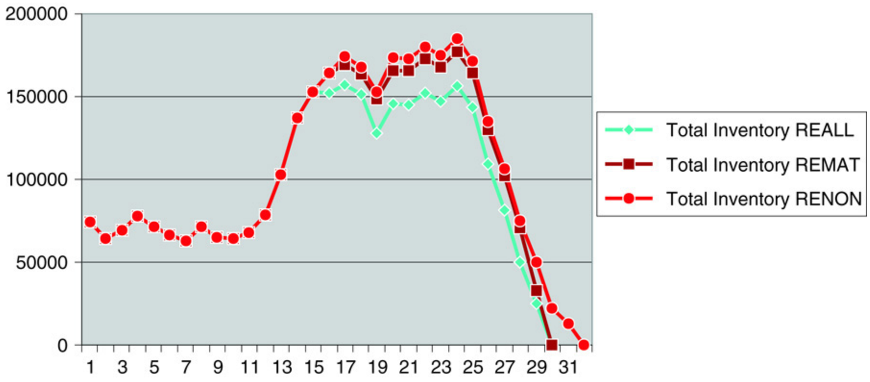

Fig. 6. Total input and output inventory: scenario III.

orders is not changed, produces the worst results. On the other hand RENON requires the least, and REALL the greatest, CPU time to find proven optimal schedules.

The distribution of initial demand ahead of a monthly horizon, demand remaining and updated after each modification of orders, and the corresponding aggregated production schedules obtained using scheduling/rescheduling algorithm REALL are shown in Fig. 2 for scenario I.

For a comparison, Fig. 3 shows for each scenario the distribution over the horizon of the updated demand, optimal production schedule obtained ex post for the updated demand and the concatenated production schedule for the entire horizon constructed by using the REALL algorithm. The figure indicates that reactive scheduling by the REALL algorithm constructs production schedules very close to the optimal ex post schedule obtained when the updated demand is known for the entire horizon.

Finally, Figs. 4–6 show how the total inventory of product-specific materials and finished products varies over the horizon for each rescheduling algorithm, respectively for each scenario with increasing, decreasing and unchanged total demand. Again, the best results, i.e., the lowest maximum inventory level is achieved for algorithm REALL, whereas RENON leads to the highest inventory level.

7. Conclusion

In this paper, various reactive scheduling policies and rescheduling algorithms based on the mixed integer programming formulations are proposed for a dynamic, make-to-order manufacturing environment. The proposed algorithms differ both in performance and the extent of induced schedule changes. The computational results have indicated that the proposed mixed integer programming approach can be applied for reactive scheduling to iteratively update production schedules over a dynamic planning horizon. The rescheduling algorithms are capable of finding proven optimal schedules in short CPU times for large-size problems that can be encountered in industrial practice.

References

[1] E.A. Silver, D.F. Pyke, R. Petersen, Inventory Management and Production Planning and Scheduling, John Wiley, New York, 1998.

[2] J.F. Shapiro, Mathematical programming models and methods for production planning and scheduling, in: S.C. Graves, A.H.G. Rinnooy Kan, P.H. Zipkin. (Eds.), Handbook in Operations Research and Management Science: Logistics of Production and Inventory, North-Holland, Amsterdam, 1993.

[3] G.E. Vieira, J.W. Herrman, E. Lin, Rescheduling manufacturing systems: A framework of strategies, policies and methods, Journal of Scheduling 6 (1) (2003) 39–62.

[4] R. Rangsaritratsamee, W.G. Ferrel, M.B. Kurz, Dynamic rescheduling that simultaneously considers efficiency and stability, Computers and Industrial Engineering 46 (2004) 1–15.

[5] S.F. Smith, Reactive scheduling systems, in: In: D.E. Brown, W.T. Scherer (Eds.), Intelligent Scheduling Systems, Kluwer Academic Publishers, Boston, 1995, pp. 155–192.

[6] J. Sun, D. Xue, A dynamic reactive scheduling mechanism for responding to changes of production orders and manufacturing resources, Computers in Industry 46 (2001) 189–207.

[7] G.L. Nemhauser, L.A. Wolsey, Integer and Combinatorial Optimization, John Wiley, New York, 1998.

[8] M.A. Carravilla, J. Pinho de Sousa, Hierarchical production planning in a make-to-order company: A case study, European Journal of Operational Research 86 (1995) 43–56.

[9] D.-Y. Liao, S.-C. Chang, K.W. Pei, C.-M. Chang, Daily scheduling for R&D semiconductor fabrication, IEEE Transactions on Semiconductor Manufacturing 9 (1996) 550–560.

[10] T. Sawik, Integer programming approach to production scheduling for make-to-order manufacturing, Mathematical and Computer Modelling 41 (1) (2005) 99–118.

[11] T. Sawik, Hierarchical approach to production scheduling in make-to-order assembly, International Journal of Production Research 44 (4) (2006) 801–830.

[12] J. Röslof, I. Harjunkoski, J. Björkqvist, T. Karlsson, T. Westerlund, An MILP-based reordering algorithm for complex industrial scheduling and rescheduling, Computers and Chemical Engineering 25 (2001) 821–828.

[13] J. Röslof, I. Harjunkoski, T. Westerlund, J. Isaksson, Solving a large scale industrial scheduling problem using MILP combined with a heuristic procedure, European Journal of Operational Research 138 (2002) 29–42.

[14] T. Sawik, Multi-objective master production scheduling in make-to-order manufacturing, International Journal of Production Research (2006) (in press). Available on line: doi: 10.1080/00207540600847152.