Tadeusz SAWIK Akademia Górniczo-Hutnicza

REACTIVE SCHEDULING IN MAKE-TO-ORDER MANUFACTURING BY MIXED INTEGER PROGRAMMING

Summary. New algorithms based on mixed integer programming models are proposed for reactive scheduling in a dynamic, make-to-order manufacturing environment. The problem objective is to update production schedule subject to service level and inventory constraints, whenever customer orders are modified. Numerical examples modeled after a real-world production scheduling/rescheduling in the electronics industry are presented and some results of computational experiments are reported.

REAKTYWNE HARMONOGRAMOWANIE PRODUKCJI NA ZAMÓWIENIE METODĄ PROGRAMOWANIA CAŁKOWITOLICZBOWEGO

Streszczenie. W pracy przedstawiono nowe algorytmy reaktywnego harmonogramowania produkcji zamawianej, oparte na modelach programowania całkowitoliczbowego. Zamówienia mogą być modyfikowane przez odbiorców w całym horyzoncie planowania. Celem harmonogramowania jest minimalizacja liczby spóźnionych zamówień oraz łącznych zapasów materiałów i gotowych wyrobów. Zastosowanie proponowanych algorytmów ilustrują przykłady liczbowe zaczerpnięte z przemysłu elektronicznego oraz wyniki eksperymentów obliczeniowych.

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, e.g. [4, 7]. 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 each other, then reallocation of orders to the earlier periods with surplus of capacity is restricted due to later material availability. As a result, the number of tardy orders may increase or even some orders may remain unscheduled during the planning horizon.

The purpose of this paper is to present new algorithms, based on integer programming formulations, for reactive scheduling ([8, 9, 10]) in a dynamic, make-to-order manufacturing, where customers may modify or cancel their orders or place new orders during the planning horizon. The problem objective is to dynamically assign/reassign customer orders with various due dates to planning periods with limited capacities, to minimize the number of tardy orders and the input and output inventory over a planning horizon.

In the literature on production planning and scheduling the integer programming models have been widely used, e.g. [3, 7]. In industrial practice, however, the application of integer programming for production scheduling is limited, in particular in make-to-order manufacturing, e.g. [1, 2, 4, 5, 6]. For example, a scheduling tool with rescheduling capabilities for the semiconductor industry, based on integer programming formulation is presented in [2]. However, the model is solved by an approximate technique and optimal solution was not attempted.

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 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 $i \in I = \{1, \ldots, m\}$ is made up of $m_i \ge 1$ 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 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 $p_{ij} \ge 0$ be the processing time in stage *i* of each product in order $j \in J$. The orders are processed and transferred among the stages in lots of various size that depends on the ordered product type and let b_j be the size of production lot for order *j*.

The planning horizon consists of h planning periods (e.g. working days). Let $T = \{1, \ldots, h\}$ 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.

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 $J1 \cup J2 = J$, and $J1 \cap J2 = \emptyset$.

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, however, the latter assumption can be easily relaxed [5] to allow for 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 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 horizon. The horizon can be progressively shifted to take into account the orders 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 by minimizing the number of tardy orders, with limited maximum earliness of orders and by this the limited total input and output inventory.

3. Mixed Integer Programs for Reactive Scheduling

In this section mixed integer programming formulations are proposed for customer orders 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 me on the maximum earliness E_{max} of customer orders.

Table 1

	1.10						
Indices							
i	=	processing stage, $i \in I = \{1, \dots, m\}$					
j	=	customer order, $j \in J = \{1, \dots, n\}$					
k	=	product type, $k \in K = \{1, \dots, r\}$					
t	=	planning period, $t \in T = \{1, \dots, h\}$					
Input Parameters							
a_j, d_j, s_j	=	arrival date, due date, size of order j					
b_j	=	production lot for order j					
Cit	=	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							
ui	=	1, if order j is completed after due date; otherwise $u_j = 0$ (unit					
		penalty for tardy orders)					
xit	=	1, if order j is performed in period t; otherwise $x_{jt} = 0$ (order					
		assignment variable)					
$y_{jt} \ge 0$	=	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					

Notation: Initial Scheduling

Model $MaxSL(\overline{E})$: Customer orders assignment to Maximize Service Level subject to Maximum Earliness constraints Maximize

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

or Minimize

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

subject to

1. Order assignment constraints

- each indivisible customer order is assigned to exactly one planning period,

$$\sum_{t\in T: \ t\geqslant a_j} x_{jt} = 1; \ j\in J1 \tag{3}$$

- each divisible customer order is assigned to at most two consecutive planning periods,

$$x_{jt} + x_{jt+1} \leq 2; \ j \in J2, t \in T : \ a_j \leq t \leq h-1$$
 (4)

$$x_{jt} + x_{j\tilde{t}} \leqslant 1; \ j \in J2, t \in T, \tilde{t} \in T: \ a_j \leqslant t \leqslant h-2, \ \tilde{t} \geqslant t+2 \tag{5}$$

2. Order allocation constraints

- each order must be completed,

$$\sum_{t \in T: \ t \ge a_j} y_{jt} = 1; \ j \in J \tag{6}$$

- each indivisible order is completed in a single period,

$$x_{jt} = y_{jt}; \ j \in J1, \ t \in T: \ t \ge a_j \tag{7}$$

- each divisible order is allocated among all the periods that are selected for its assignment,

$$x_{jt} \ge y_{jt}; \ j \in J2, \ t \in T : t \ge a_j \tag{8}$$

- the minimum portion of a divisible order alloted to one period is not less than the batch size,

$$y_{jt} \ge b_j x_{jt} / s_j; \ j \in J2, \ t \in T : \ t \ge a_j \tag{9}$$

3. Tardy orders constraints

- an indivisible tardy order is completed after its due date,

$$u_j = \sum_{t \in T: \ t > d_j} x_{jt}; \ j \in J1$$
(10)

- a divisible tardy order is partly assigned after its due date,

$$u_j \ge \sum_{t \in T: \ t > d_j} y_{jt}; \ j \in J2 \tag{11}$$

$$u_j \leqslant \sum_{t \in T: \ t > d_j} x_{jt}; \ j \in J2 \tag{12}$$

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,

$$\sum_{j \in J_i} p_{ij} s_j y_{jt} \leqslant c_{it} m_i; i \in I, t \in T$$
(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 E_{max} ,

$$(d_j - t)x_{jt} \leqslant E_{max}; \ j \in J, t \in T : \ t \geqslant a_j \tag{14}$$

 $E_{max} \leqslant \overline{E}$ (15)

6. Variable nonnegativity and integrality constraints

$$u_j \in \{0, 1\}; \ j \in J$$
 (16)

$$x_{jt} \in \{0,1\}; \ j \in J, t \in T : \ t \ge a_j$$
(17)

$$0 \leqslant y_{jt} \leqslant 1; \ j \in J, t \in T : \ t \geqslant a_j \tag{18}$$

$$E_{max} \ge 1, integer$$
 (19)

The objective function represents 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(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 limited maximum earliness of orders and by this the limited total input and output inventory level.

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

$$MaxSL(E) = \max\{(1) : (2) - (19)\}$$
(20)

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 orders assignment to Maximize Service Level

$$MaxSL = \max\{(1): (2) - (13), (16) - (18)\}$$
(21)

where all materials are assumed to be available at the beginning, i.e. $a_j = 1$ for each order $j \in J$.

2. Model $MinME(\overline{U})$: Customer orders assignment to Minimize Maximum Earliness subject to service level constraints

 $MinME(\overline{U}) = \min\{E_{max} : U_{sum} \leq \overline{U}, (2) - (14), (16) - (19)\}$ (22)

where $1 - \overline{U}/n$ is the solution value of (21)

The objective function of (22) implicitly limits the maximum level of the total input and output inventory over the planning horizon.

In the above sequential decision making framework, first the solution to MaxSL determines the maximum service level. Then, the minimum value of the maximum earliness is found using model $MinME(\overline{U})$ to implicitly limit total inventory, subject to service level constraints. The solution to $MinME(\overline{U})$ determines the optimal allocation $\{x_{it}^*, y_{it}^*\}$ 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 orders 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 waiting for material supplies, i.e. rescheduling of all unmodified orders assigned to periods greater than $t_{mod} + E_{max}$.

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 is split into two disjoint subsets: J_{old}^S of schedulable orders and J_{old}^N of fixed, non-schedulable orders for which the assignment to planning periods cannot be changed. In particular, in algorithm REMAT that accounts for the input inventory of product specific materials supplied before t_{mod} and available E_{max} periods ahead of the orders due dates at the latest, the subset of nonschedulable orders contains orders in J_{old} remaining for completion, such that have been assigned to periods in the subset $T_{old}^N = \{t_{mod}, \ldots, t_{mod} + E_{max}\}$ of remaining periods in $T_{old} = \{t_{mod}, \ldots, h\}$.

In the sequel, denote by apostrophe (') the updated values of some parameters and decision variables after each modification of orders. For example

Table 2

Notation: Rescheduling

Input Parameters								
h'	=	new planning horizon						
t_{mod}	=	the planning period immediately following modification of orders						
Jmod	=	set of modified orders						
Jold	=	subset of orders in J remaining for completion						
J_{old}^N, J_{old}^S	=	subset of orders in J_{old} , respectively non-						
		schedulable, schedulable						
$T_{new} = \{h+1,\ldots,h'\}$	=	set of new planning periods						
$T_{old} = \{t_{mod}, \dots, h\}$	=	subset of remaining planning periods in T						
T_{old}^N	=	subset of periods in T_{old} with fixed assignment						
		of orders in J_{old}						
$J' = J_{mod} \cup J_{old}$	=	updated set of orders						
$T' = T_{old} \cup T_{new}$	=	updated set of planning periods						

apostrophe (') denotes updated parameters after modification of orders

 s'_j denotes the modified size of customer order $j \in J' = J_{old} \cup J_{mod}$, where $s'_j = s_j, j \in J_{old}$ and $s'_j \neq s_j, j \in J_{mod}$.

Algorithm REALL

Step 0. Split the set J_{old} of orders remaining for completion into two disjoint subsets: J_{old}^S of schedulable orders and J_{old}^N of fixed, non-schedulable orders.

$$J_{old}^N = \emptyset \tag{23}$$

$$J_{old}^{S} = J_{old} \tag{24}$$

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

$$h_1 = \min\{h1 : \max_{i \in I} (\frac{\sum_{j \in J'} p_{ij} s'_j}{m_i \sum_{t=t_{mod}}^{h1} c_{it}}) \le 1\}$$
(25)

$$h_2 = \max_{j \in J'} (d_j) \tag{26}$$

If $\max\{h_1, h_2\} \leq h$, then set h' = h. Otherwise set $h' = \max\{h_1, h_2\}$, $T_{new} = \{h+1, \dots, h'\}$ and $T' = T_{old} \cup T_{new}$. Step 2. Do not change the assignment in period t_{mod} of partially completed, two-period orders in J2, i.e.,

$$y'_{j,t_{mod}} = y_{j,t_{mod}}, \ j \in J2: \ x_{j,t_{mod}-1} = 1$$
 (27)

Step 3. Solve $MaxSL(\overline{E})$, (20) 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

$$a'_{j} = \begin{cases} \max\{1, d_{j} - E_{max}\} & \text{if } j \in J_{old} \\ \max\{t_{mod}, d_{j} - E_{max}\} & \text{if } j \in J_{mod}. \end{cases}$$
(28)

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

Algorithm REMAT

Step 0. Split the set J_{old} of orders remaining for completion into two disjoint subsets: J_{old}^S of schedulable orders and J_{old}^N of fixed, non-schedulable orders.

$$J_{old}^N \doteq \{j \in J_{old} : \sum_{t_{mod} \leqslant t \leqslant t_{mod} + E_{max}} x_{jt} = 1\}$$
(29)

 $J_{old}^S = J_{old} \setminus J_{old}^N \tag{30}$

Set $T_{old}^N = \{t_{mod}, \dots, t_{mod} + E_{max}\}.$

Step 2. Do not change the assignment of non-schedulable orders $j \in J_{old}^N$, i.e.,

$$y'_{jt} = y_{jt}, \ j \in J^N_{old}, t \in T^N_{old}$$
(31)

$$y'_{j,t_{mod}+E_{max}+1} = y_{j,t_{mod}+E_{max}+1}, \ j \in J^{N}_{old} \cap J2: \ x_{j,t_{mod}+E_{max}} = 1$$
(32)

Algorithm RENON

Step 0. Split the set J_{old} of orders remaining for completion into two disjoint subsets: J_{old}^S of schedulable orders and J_{old}^N of fixed, non-schedulable orders.

$$J_{old}^N = J_{old} \tag{33}$$

$$J_{old}^S = \emptyset \tag{34}$$

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

$$y'_{jt} = y_{jt}, \ j \in J_{old}, t \in T_{old} \tag{35}$$

5. Input and Output Inventory

In this section some formulae are derived for calculation 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 more transparent 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 one 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 $j \in J_{mod}$ such that $d_j - E_{max} < t_{mod} \leq d_j$ and supplied before t_{mod} differs from the modified amount of those materials required after the orders 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 $(\Delta INP_k < 0)$ or surplus $(\Delta INP_k > 0)$ of product-specific material inventory in period $t_{mod} - 1$ with respect to the amount required for the modified orders $j \in J_{mod}$ is

$$\Delta INP_k = \sum_{j \in J_k \cap J_{mod}: d_j - E_{max} < t_{mod} \leqslant d_j} (s'_j - s_j); \ k \in K$$
(36)

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 INP(t) of product-specific materials can be calculated as below.

$$INP(t) = INP(t_{mod} - 1) + \sum_{j \in J': t_{mod} \leqslant a'_j \leqslant t} s'_j - \sum_{j \in J_{mod}: d_j - E_{max} < t_{mod} \leqslant d_j} s_j - \sum_{j \in J', \tau \in T \cap T': a'_j \leqslant \tau \leqslant t} s'_j y'_{j\tau}; \ t \in T': t \geqslant t_{mod}$$
(37)

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

In (37), the input inventory 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 (37) balances in period t_{mod} the shortage or the surplus of product-specific materials supplied by period t mod -1.

Similarly, the output inventory OUP(t) of finished products can be expressed as below.

$$OUP(t) = \sum_{d>t} OUP_d(t_{mod} - 1) + \sum_{j \in J', \tau \in T \cap T': a_j' \leq \tau \leq t < d_j} s_j' y_{j\tau}'; \ t \in T': t \ge t_{mod}(38)$$

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

In (38), the output inventory 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 TOT(t) = INP(t) + OUP(t) in each period t can be found by summing the corresponding right hand sides of (37) and (38). In particular, the total input and output inventory $TOT(t_{mod} - 1)$ in the last period of the previous schedule can be expressed as below.

$$TOT(t_{mod}-1) = \sum_{j \in J: d_j \leqslant t_{mod}-1} s_j (1 - \sum_{a_j \leqslant \tau \leqslant t_{mod}-1} y_{j\tau}) + \sum_{j \in J: t_{mod} \leqslant d_j \leqslant t_{mod}-1 + E_{max}} s_j (39)$$

The first summation term in (39) is the inventory of product-specific materials for customer orders due by period $t_{mod} - 1$, and the second term is the inventory of product-specific materials and finished products of customer orders due after period $t_{mod} - 1$, respectively waiting for processing in the system and for shipping to customers.

The first term represents the input inventory in period $t_{mod} - 1$ of productspecific 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 $t_{mod} - 1$ 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.

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 is a flexible flowshop made up of six processing stages with parallel machines. In the distribution center 10 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. 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

P

- 10 product types of three product groups, each to be processed on a separate group of flashing/flexing machines,
- 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 537760.
- 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:

roduct	type/stage	e 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

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

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 number of tardy suborders as a measure of the customer service level subject to maximum earlines constraints to limit the total inventory level. In the computational experiments the following three modifications of customer orders during the planning horizon are considered:



Fig. 1. Distribution of demand and aggregate production for algorithm REALL

Table 3

Computational results

						S CWLW S SALLA LE	
$Model/t_{mod}$	Var.	Bin.	Cons.	Nonz.	Solution values [†]	CPU^{\dagger}	
INITIAL SCHEDULING FOR BEGINNING DEMAND							
MaxSL	29310	14656	18198	133276	$U_{sum}^{*} = 0$	3.60	
MinME(0)	31507	15753	33057	148946	$E_{max}^* = 6$	387.88	
REALL							
$MaxSL(6)/t_{mod} = 6$	22522	11565	19779	316229	$U_{sum}^* = 2, h' = 31$	20.61	
$MaxSL(6)/t_{mod} = 14$	15558	8013	13116	195823	$U_{sum}^* = 3, h' = 32$	28.24	
$MaxSL(6)/t_{mod} = 24$	4059	2112	3928	40811	$U_{sum}^* = 5, h' = 33$	1.22	
REMAT							
$MaxSL(6)/t_{mod} = 6$	15817	8145	11237	186527	$U_{sum}^* = 3, h' = 31$	9.07	
$MaxSL(6)/t_{mod} = 14$	7656	3959	5610	77634	$U_{sum}^* = 6, h' = 32$	1.69	
$MaxSL(6)/t_{mod} = 24$	222	105	373	1898	$U_{sum}^* = 8, h' = 33$	0.02	
RENON							
$MaxSL(6)/t_{mod} = 6$	371	152	1281	6121	$U_{sum}^* = 8, h' = 35$	0.04	
$MaxSL(6)/t_{mod} = 14$	537	248	1625	8479	$U_{sum}^* = 10, h' = 35$	0.09	
$MaxSL(6)/t_{mod} = 14$	382	184	759	4344	$U_{sum}^* = 11, h' = 36$	0.09	
*							

 $^\dagger U^*_{sum}$ - number of tardy orders, E^*_{max} - maximum earliness, h' - planning horizon

[‡] CPU seconds for proving optimality on a PC Pentium IV, 1.8GHz, RAM 1GB /CPLEX v.9.1



Fig. 2. Total input and output inventory for various rescheduling algorithms

- 13 customer orders due in periods 8-30 are modified in period $t_{mod} = 6$. The resulting increase of demand is 70200 products.
- 13 customer orders due in periods 15-30 are modified in period $t_{mod} = 14$. The resulting increase of demand is 15950 products.
- 8 customer orders due in periods 26-30 are modified in period $t_{mod} = 24$. The resulting increase of demand is 14960 products.

The resulting total increase of demand is 101110 products.

The characteristics of integer programs MaxSL, $MinME(\overline{U})$ (with $\overline{U} = 0$) for the initial scheduling and $MaxSL(\overline{E})$ (with $\overline{E} = 6$) for the three rescheduling algorithms REALL, REMAT and RENON, and the solution results are summarized in Table 3. 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 AMPL programming language and the CPLEX v.9.1 solver on a laptop with Pentium IV at 1.8GHz and 1GB RAM.

Table 3 indicates 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 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 production schedules obtained using scheduling/rescheduling algorithm REALL are shown in fig.1. For a comparison, fig.2 shows how the total inventory of product specific materials and finished products varies over the horizon for each rescheduling algorithm. The lowest maximum inventory level is achieved for REALL whereas RENON leads to the highest level.

7. Conclusion

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

REFERENCES

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

- Liao D.Y., Chang S.C., Pei K.W., Chang C.M.: Daily scheduling for R&D semiconductor fabrication. IEEE Transactions on Semiconductor Manufacturing, 1996, 9, p. 550–560.
- Nemhauser G.L., Wolsey L.A.: Integer and Combinatorial Optimization. 1998, (Wiley: New York).
- Sawik T.: Master scheduling in make-to-order assembly by integer programming. In: M. Zaborowski (ed.), Automation of Discrete Processes, 2004, (WNT: Warszawa), p. 285–294.
- Sawik T.: Integer programming approach to production scheduling for make-to-order manufacturing. Mathematical and Computer Modelling, 2005, 41(1), p. 99–118.
- Sawik T.: Hierarchical approach to production scheduling in make-to-order assembly. International Journal of Production Research, 2006, 44(4), p. 801– 830.
- Shapiro J.F.: Mathematical Programming Models and Methods for Production Planning and Scheduling. In: Handbook in Operations Research and Management Science: Logistics of Production and Inventory, edited by S.C.Graves, A.H.G.Rinnooy Kan, and P.H.Zipkin. 1993, (North-Holland: Amsterdam).
- Smith S.F.: Reactive scheduling systems. In: D.E.Brown and W.T.Scherer (eds.), Intelligent Scheduling Systems, 1995 (Kluwer Academic Publishers: Boston), p. 155–192.
- Sun J., Xue D.: A dynamic reactive scheduling mechanism for responding to changes of production orders and manufacturing resources. Computers in Industry, 2001, 46, p. 189–207.
- Vieira G.E., Herrman J.W., Lin E.: Rescheduling manufacturing systems: a framework of strategies, policies and methods. Journal of Scheduling, 2003, 6(1), p. 39–62.

Recenzent: Prof. dr hab. inż. Andrzej Świerniak

Omówienie

W pracy przedstawiono nowe algorytmy reaktywnego harmonogramowania produkcji zamawianej, oparte na modelach programowania całkowitoliczbowego. Zamówienia mogą być modyfikowane przez odbiorców w całym horyzoncie planowania. Celem harmonogramowania jest minimalizacja liczby spóźnionych zamowień oraz łącznych zapasów materiałów i gotowych wyrobów. Zastosowanie proponowanych algorytmów ilustrują przykłady liczbowe zaczerpnięte z przemysłu elektronicznego oraz wyniki eksperymentów obliczeniowych.