MULTI-OBJECTIVE MASTER PRODUCTION SCHEDULING IN MAKE-TO-ORDER DISCRETE MANUFACTURING[1]

Tadeusz Sawik

AGH University of Science & Technology
Department of Computer Integrated Manufacturing
Al.Mickiewicza 30, 30-059 Kraków, POLAND
tel.: +48 12 617 39 92
e-mail: ghsawik@cyf-kr.edu.pl

Abstract:
This paper presents a lexicographic approach and integer programming formulations for multi-objective, long-term production scheduling in make-to-order manufacturing environment. The problem objective is to assign customer orders for various product types to planning periods and to select machines for assignment in every period to complete all the orders with minimum number of tardy or early orders as a primary optimality criterion and to level machine assignments over a planning horizon as a secondary criterion. Some cutting constraints that make the integer programming formulations stronger are derived. The approach has been applied to optimize master production schedules in a flexible flowshop made up of several processing stages in series and an output buffer of limited capacity to hold completed products before delivery to the customer. Each processing stage consists of identical, parallel machines. Numerical examples modeled after a real-world make-to-order flexible assembly line in the electronics industry are provided and some computational results are reported.

Keywords:
Production scheduling, Flexible flowshop, Integer programming, Make-to-order manufacturing.

1. Introduction

One of the basic goals of production scheduling in make-to-order discrete manufacturing is to maximize customer service level, that is, to maximize the fraction of customer orders filled on or before their due dates. A typical customer due date performance measure is minimization of the number of tardy orders, e.g. [1,2,4].

The purpose of this paper is to present integer programming formulations for a multi-objective master production scheduling in make-to-order manufacturing environment and to apply the approach for long-term production scheduling in a flexible flowshop. The scheduling objective is to allocate a set of customer orders with various due dates among planning periods to minimize number of tardy or early orders and to level production resources over a planning horizon, e.g. [3,5].

In the literature on production planning and scheduling the integer programming models have been widely used. In industrial practice, however, the application of integer programming for scheduling is limited. For example, a hierarchical approach and integer programs for production scheduling in make-to-order company are presented in [1], however computational results are based on developed heuristics. An integer goal programming formulation for master scheduling with a due date related criterion is also presented in [2], and the focus is again on application of heuristic approaches.

[1]This work has been partially supported by AGH and KBN (Poland), and by Motorola (USA).

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 integer programming formulations for a lexicographic approach to master scheduling are presented in Section 3. Some cutting constraints are derived in Section 4. Numerical examples modeled after a real-world make-to-order assembly system and some computational results are provided in Section 5. Conclusions are made in the last section.

2. Problem Description

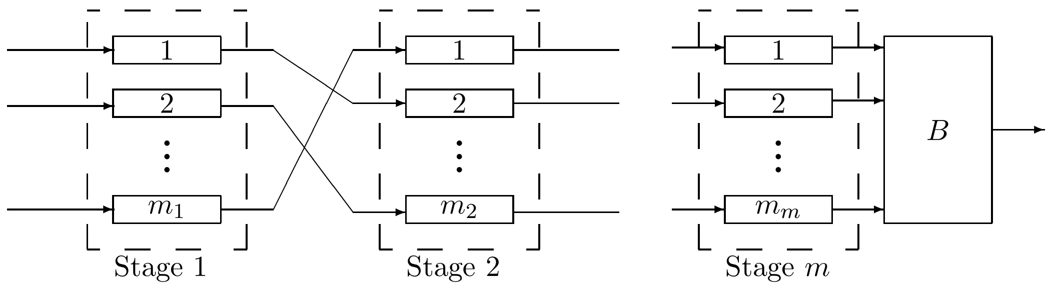

Figure 1. A flexible flowshop with finite output buffer

The production system under study is a flexible flowshop (Fig. 1) that consists of m processing stages in series and an output buffer of limited capacity B to hold completed products before delivery to the customer. Typically, the capacity B of output buffer is not large to keep low inventory costs and to limit an early completion of customer orders before the customer required shipping dates.

Each stage $i \in I = \{1, \ldots, m\}$ 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. 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 (quantity of ordered products). Denote by $J(d)$ the subset of orders with the same due date $d \in D$, where $D = \{d_j : j \in J\}$ is the set of distinct due dates of all customer orders.

Each order requires processing in various processing stages, however some orders may bypass some stages. Let $J_i \subset J$ be the subset of orders that must be processed in stage i, and let $p_{ij} > 0$ be the processing time in stage i of each product in order $j \in J_i$. The orders are processed and transferred among the stages in lots of various size 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.

It is assumed that each customer order must be fully completed in exactly one planning period and the available capacity is sufficient to schedule all orders during the planning horizon. Otherwise, the orders that cannot be scheduled in periods 1 through h due to insufficient capacity can be assigned at a significant penalty to a dummy planning period $h + 1$ with infinite capacity.

The objective of master production scheduling is to assign customer orders to planning periods and to select machines for assignment in every period to minimize number of tardy or early orders as a primary optimality criterion and to level machine assignments as a secondary criterion.

3. Multi-Objective Master Scheduling

In this section the integer programming formulations are presented for the multi-objective master production scheduling. First, for a monolithic approach (model **MPS**) and then for a hierarchical, two-level approach (models **MPS1** and **MPS2**).

Table 1. Decision variables

v_{jt}	=	1, if order j is performed in period t; otherwise $v_{jt} = 0$ (order assignment variable)
w_{it}	=	number of machines selected for assignment in stage i in period t (machine selection variable)
E_{sum}	=	number of early orders
M_{max}	=	maximum number of machines selected for assignment in a single planning period
U_{sum}	=	number of tardy orders

Model MPS: *Master Production Scheduling*

Minimize

$$\lambda_1 \sum_{j \in J: t > d_j} v_{jt} + \lambda_2 \sum_{j \in J: t < d_j} v_{jt} + \lambda_3 M_{max} \tag{1}$$

subject to

1. Order assignment constraints:

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

$$\sum_{t \in T} v_{jt} = 1; \; j \in J \tag{2}$$

2. Machine assignment constraints:

- in every period the number of machines selected for assignment at each stage is not greater than the maximum number of available machines, and the total number of assigned production lots,

- in every period the total number of machines selected for assignment cannot exceed the maximum number of machine assignments to be minimized,

- in every period the demand on capacity at each processing stage cannot be greater than the total capacity of machines selected for assignment in this period,

$$w_{it} \le m_i; \; i \in I, t \in T \tag{3}$$

$$w_{it} \le \sum_{j \in J_i} \lceil s_j / b_j \rceil v_{jt}; \; i \in I, t \in T \tag{4}$$

$$\sum_{i \in I} w_{it} \le M_{max}; \; t \in T \tag{5}$$

$$\sum_{j \in J_i} p_{ij} s_j v_{jt} \le c_{it} w_{it}; i \in I, t \in T \tag{6}$$

3. Output buffer capacity constraints:

- in every period the total number of products completed before their due dates and waiting for shipping to customers cannot exceed the output buffer capacity,

$$\sum_{j \in J, \tau \in T: a_j \le \tau \le t < d_j} s_j v_{j\tau} \le B; \; t \in T \tag{7}$$

4. Variable integrality constraints

$$v_{jt} \in \{0,1\};\ j \in J, t \in T:\ t \geq a_j \tag{8}$$

$$w_{it} \geq 0, integer;\ i \in I, t \in T \tag{9}$$

$$M_{max} \geq 0, integer \tag{10}$$

where $\lceil \cdot \rceil$ is the least integer not less than $\cdot$.

The objective of the master scheduling problem is to minimize weighted sum of tardy and early orders and to level machine assignments over a planning horizon.

In the objective function (1) tardy orders are penalized at a much higher rate than early orders and maximum machine assignments, i.e., $\lambda_1 \gg \lambda_2 > \lambda_3$. Therefore, a lexicographic approach can be applied, where first the customer orders are allocated among planning periods to find the optimal numbers of tardy and early orders and then the final master schedule is determined to level machine assignments over the horizon for a minimum number of tardy and early orders, see Fig. 2.

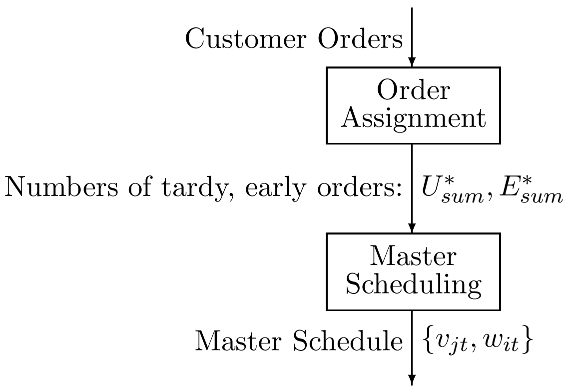

Figure 2. *A lexicographic approach to multi-objective master scheduling*

Model MPS1: *Order assignment to minimize number of tardy and early orders*

Minimize

$$\lambda_1 \sum_{j \in J: t > d_j} v_{jt} + \lambda_2 \sum_{j \in J: t < d_j} v_{jt} \tag{11}$$

subject to (2), (7), (8) and

Capacity constraints:

- in every period the demand on capacity at each processing stage cannot be greater than the maximum available capacity in this period,

$$\sum_{j \in J_i} p_{ij} s_j v_{jt} \leq c_{it} m_i; i \in I, t \in T \tag{12}$$

In the objective function (11) tardy orders are penalized at a higher rate than early orders, i.e., $\lambda_1 > \lambda_2$.

The solution to MPS1 determines optimal order assignment $\{v^*_{jt}\}$ and the resulting minimum number of tardy and early orders, respectively $U^*_{sum} = \sum_{j \in J: t > d_j} v^*_{jt}$ and $E^*_{sum} = \sum_{j \in J: t < d_j} v^*_{jt}$. Given the optimal values of $U_{sum},\ E_{sum}$, the next step is to find master schedule such that allocates customer orders among planning periods and levels machine assignments over the planning horizon for a minimum number of tardy and early orders.

Model MPS2: *Master scheduling for a minimum number of tardy and early orders*

Minimize

$$M_{max} \tag{13}$$

subject to (2) - (10) and
- numbers of tardy and early orders are at minimum

$$\sum_{j\in J:t>d_j} v_{jt} = U^*_{sum} \tag{14}$$

$$\sum_{j\in J:a_j\leq t<d_j} v_{jt} = E^*_{sum} \tag{15}$$

The solution to MPS2 determines the optimal master schedule, i.e., the optimal allocation of customer orders among planning periods, $\{v^*_{jt}\}$ and a leveled machine assignment over the horizon, $\{w^*_{it}\}$ such that numbers of tardy and early orders are at minimum.

4. Cutting Constraints

A necessary condition for problem **MPS** to have a feasible solution with all customer orders completed during the planning horizon is that for each processing stage $i \in I$ the total demand on capacity does not exceed total available capacity, that is,

$$\frac{\sum_{j\in J} p_{ij}s_j}{m_i\sum_{t\in T} c_{it}} \leq 1; \forall i \in I \tag{16}$$

If the customer orders were arbitrarily divisible and could be completed during more than one planning period, all orders were ready for completing at the beginning of the planning horizon and the output buffer capacity was unlimited to allow for any earliness of the order completion, then the feasibility condition (16) would be sufficient. Due to the discrete nature of customer orders, however, it is possible that some planning periods will not be filled exactly to their capacities. As a result the necessary condition (16) is not sufficient for all orders to be scheduled during the planning horizon.

In this section some cutting constraints on decision variables are derived by relating for each due date the local demand on required capacity to available capacity and the cumulative demand on required capacity to available cumulative capacity.

For each due date $d \in D$ denote by $lcr(d)$ the local capacity ratio defined below.

$$lcr(d) = \max_{i\in I}\left(\frac{\sum_{j\in J(d)} p_{ij}s_j}{m_i c_{id}}\right); d \in D \tag{17}$$

Denote by D_{l0}, D_{l1}, the subset of locally satisfiable, locally unsatisfied due dates, respectively:

$$D_{l0} = \{d \in D : lcr(d) \leq 1\}, \quad D_{l1} = \{d \in D : lcr(d) > 1\} \tag{18}$$

For each locally unsatisfied due date at least one order should be moved to another period to reduce local demand on capacity, which leads to the following cutting constraint:

Cutting constraint on assignment of orders with locally unsatisfied due dates

$$\sum_{j\in J(d)} v_{jd} \leq |J(d)| - 1; d \in D_{l1} \tag{19}$$

where $|\cdot|$ is the power of set $\cdot$.

Constraint (19) ensures that for each locally unsatisfied due date the number of orders completed on the due date is strictly less than the total number of orders with this due date.

A due date $d \in D$ can be satisfied if for each processing stage i and any period $t < d$ the cumulative demand on capacity of all orders with due dates not greater than d and arrival dates not less than t does not exceed the cumulative capacity available in this stage in periods t through d. For each due date $d \in D$ denote by $ccr(d)$ the cumulative capacity ratio defined below.

$$ccr(d) = \max_{i \in I, t \in T: t < d} \left(\frac{\sum_{f \in D, j \in J(f): t \le a_j \le f \le d} p_{ij} s_j}{m_i \sum_{\tau \in T: t \le \tau \le d} c_{i\tau}} \right); d \in D \tag{20}$$

Denote by D_{c0}, D_{c1}, the subset of globally satisfiable, globally unsatisfied due dates, respectively:

$$D_{c0} = \{d \in D : ccr(d) \le 1\}, \quad D_{c1} = \{d \in D : ccr(d) > 1\} \tag{21}$$

The due dates both locally and globally satisfiable are referred to as satisfiable due dates. In contrast, the due dates with local and cumulative demand on capacity exceeding available capacity are referred to as potentially unsatisfied due dates.

Cutting constraints on assignment of orders with satisfiable due dates

$$v_{jt} = 0; \; j \in J(d), \text{ if } D_{c1} = \emptyset \text{ then } d \in D \text{ else } d \in D_{l0} \bigcap D_{c0}, t \in T : t > d_j \tag{22}$$

$$\sum_{t=a_j}^{d_j} v_{jt} = 1; \; j \in J(d), \text{ if } D_{c1} = \emptyset \text{ then } d \in D \text{ else } d \in D_{l0} \bigcap D_{c0} \tag{23}$$

$$\begin{aligned} v_{jd} = 1; \; j \in J(d), d \in D_{l0} \bigcap D_{c0} : d < \min\{a_j : j \in J(d'), d' \in D_{l1}\} \\ \text{or } d > \max\{d' : d' \in D_{l1}\} \end{aligned} \tag{24}$$

Constraints (22) and (23) guarantee non delayed completion of each order with a satisfiable due date. Furthermore, (24) ensures completion on due date of each order with a satisfiable due date that is earlier than the earliest arrival date or later then the latest due date of all orders in D_{l1}.

In order to meet a potentially unstaisfied due date $d \in D_{l1} \bigcap D_{c1}$, some orders $j \in J(f)$, $f \le d$ should be moved forward (delayed) to periods with slack capacity. In particular, at least one order with the locally unsatisfied due date not greater than the earliest potentially unsatisfied due date $first(D_{l1} \bigcap D_{c1}) = \min\{d : d \in D_{l1} \bigcap D_{c1}\}$ will be tardy and reallocated to some period in the set of locally satisfiable due dates after $first(D_{l1} \bigcap D_{c1})$ to reduce cumulative demand on capacity in periods 1 through $first(D_{l1} \bigcap D_{c1})$.

Cutting constraint on assignment of orders with potentially unsatisfied due dates

$$\sum_{j \in J(d), d \in D_{l1}, t \in D_{l0}:\, d \le first(D_{l1} \bigcap D_{c1}) < t} v_{jt} \ge 1 \tag{25}$$

Constraint (25) ensures that at least one order with the locally unsatisfied due date not greater than $first(D_{l1} \bigcap D_{c1})$ will be tardy and moved to some period $t > first(D_{l1} \bigcap D_{c1})$ in the set of locally satisfiable due dates.

5. Computational Examples

In this section numerical examples and some computational results are presented to illustrate possible applications of the proposed formulations and to compare the monolithic with the lexicographic approach (model **MPS** with a pair of models **MPS1** and **MPS2**). The examples are modeled after a real world distribution center for high-tech products, where finished products are assembled for shipping to customers.

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

1. Production system
   - six processing stages: 10 machines in each stage $i = 1, 2$; 20 machines in each stage $i = 3, 4, 5$; and 10 machines in stage $i = 6$,
   - output buffer capacity $B = 40000$ products.
2. Products
   - 10 product types,
   - 100 customer orders, each consisting of several suborders. Every suborder has a different volume ranging from 5 to 5200 products, the same arrival date (period 1), and different due date, and each suborder is to be completed in a single day,
   - production lot sizes: 100,100,150,50,50,50,100,100,150,50, 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	120	0	0	10
4	15	5	0	120	0	15
5	15	10	0	180	0	15
6	10	5	0	120	0	10
7	15	10	0	180	0	15
8	20	5	0	0	100	15
9	15	0	0	0	80	10
10	15	0	0	0	100	10

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

In the computational experiments four test problems are constructed with the following four regular patterns of demand:

1. Increasing, with demand skewed toward the end of the planning horizon.
2. Decreasing, with demand skewed toward the beginning of the planning horizon.
3. Unimodal, where demand peaks in the middle of the planning horizon and falls under available capacity in the first and last days of the horizon.
4. Bimodal, where demand peaks at the beginning and at the end of the planning horizon and slumps in mid-horizon.

Pattern 1 requires some orders to be completed earlier, for pattern 2 a majority of orders must be moved later in time, whereas patterns 3 and 4 require that orders are moved both early and late to reach feasibility.

In models **MPS** and **MPS1**, the weights used for tardy orders, early orders and maximum machine assignments are $\lambda_1 = 100$, $\lambda_2 = 5$ and $\lambda_3 = 1$, respectively.

The characteristics of integer programs for the examples with various demand patterns and the solution results are summarized in Table 2. The size of the integer programs (enhanced with the cutting constraints) 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 Table 2 present the solution values and CPU time in seconds required to prove optimality of the solution.

The computational experiments have been performed using AMPL programming language and the CPLEX v.7.1 solver on a laptop with Pentium III at 450MHz and 64MB RAM. The results have indicated that the lexicographic approach outperforms the monolithic approach and is capable of finding proven optimal schedules in a reasonable CPU time for large size problems with typical patterns of demand that can be encountered in the industrial practice.

Table 2. Computational results

Demand pattern/Model	Var.	Bin.	Cons.	Nonz.	$U_{sum},\ E_{sum},\ M_{max}$	CPU[†]
Increasing/MPS	16037	15856	2044	164599	0, 21, 62	420
Increasing/MPS1	15856	15856	1824	108173	0, 21, -	6.7
Increasing/MPS2	16037	15856	2044	164599	0, 21, 62	57
DecreasingMPS	17505	17324	1113	131070	1, 20, 69	>3600
DecreasingMPS1	17864	17864	946	85702	1, 20, -	49
DecreasingMPS2	17505	17324	1115	148377	1, 20, 69	230
Unimodal/MPS	7159	6981	1350	71521	0, 16, 66	190
Unimodal/MPS1	6981	6981	1192	46485	0, 16, -	5.1
Unimodal/MPS2	7159	6981	1350	71521	0, 16, 66	37
Bimodal/MPS	8684	8506	1539	79527	0, 22, 70	420
Bimodal/MPS1	8506	8506	1360	49504	0, 22, -	6.9
Bimodal/MPS2	8684	8506	1540	87434	0, 22, 70	59

[†] CPU seconds for proving optimality on a PC Pentium 450MHz, RAM 64MB /CPLEX v.7.1

6. Conclusions

This paper has presented a lexicographic approach and integer programming formulations for multi-objective master production scheduling in make-to-order manufacturing. The basic formulation has been strengthened by the addition of some cutting constraints that were derived by relating the demand on required capacity to available capacity for each subset of orders with the same due date. The computational experiments have indicated that the approach is capable of finding proving optimal solutions for large size problems in a reasonable computation time using commercially available software for integer programming.

References

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

[2] Markland, R.E., Darby-Dowman, K.H., and Minor, E.D., (1990), Coordinated production scheduling for make-to-order manufacturing. *European Journal of Operational Research,* **45**(2-3), 155–176, .

[3] Neumann, K., and Zimmermann, J., (1999), Resource levelling for projects with schedule-dependent time windows. *European Journal of Operational Research,* **117**, 591–605.

[4] Sawik, T., (2004), Integer programming approach to production scheduling for make-to-order manufacturing, *Mathematical and Computer Modelling,* in print.

[5] Sawik, T., (2004), Master scheduling in make-to-order assembly by integer programming. In: M. Zaborowski (ed.), *Automation of Discrete Processes*, WNT, Warszawa, 285–294.