Schedule Optimization with Simultaneous Lot Sizing in Chemical Process Plants

Reiner F. H. Musier
Rohm and Haas Company, Philadelphia, PA 19137

Lawrence B. Evans
Massachusetts Institute of Technology, Dept. of Chemical Engineering, Cambridge, MA 02139

Operational problems are of increasing importance as the process industries turn to the batch manufacture of high-value specialty products. A method is presented to minimize stockout for multiunit plants where customer orders may constitute nonintegral batch multiples. Product- and unit-dependent processing times and batch sizes, sequence-dependent changeover times, and product-to-unit assignment constraints are modeled for nonidentical parallel processes. Arbitrary initial conditions are also modeled. The algorithm determines the number of production runs of each product, run length (lot sizing), the assignment of run to units, and run sequence on each unit.

Scheduling with simultaneous lot sizing has not previously been addressed in the literature. The evolutionary method presented here obtains near-optimal (and often globally optimal) solutions on industrial-sized test problems. These results are significant since near-optimal scheduling of many industrial process plants now becomes possible, although problems of this complexity are intractable using exact methods.

Introduction

The Chemical Process Industries (CPI) are increasingly turning to the manufacture of high-value, specialty products, produced in smaller quantities by batch processes. For this reason, process operations have taken on an increasing importance in the CPI. While plant responsiveness to customer demands and operating efficiency often appear to be conflicting goals to production personnel, a systematic approach to optimizing manufacturing operations can help production managers to maximize both.

The scheduling problem arises in chemical process plants, because a potentially large number of customer orders (or similar production demands) compete for the limited production capacity of the plant. Whether the plant operates in a batch or semicontinuous mode, the plant is distinguished from a continuous process plant in that different products are produced at different times on a given process unit. For each unit in a process plant, the scheduling problem involves deciding which product should be processed at every point in time to optimize some suitable criterion.

Developments in production scheduling techniques are particularly timely, given the current interest in specialty chemicals and biotech products, many of which are produced in batch operations. Further, traditional industries employing batch and semicontinuous operations also express interest in computer-aided scheduling since they see it as an extension of current practice. In many plants, the individual currently performing production scheduling by hand makes use of computer-generated reports on current customer orders, inventory levels, and plant status. Providing the scheduler with optimization tools is the next logical step in the evolution of process operations management.

In response to these needs, a number of small firms are marketing commerical scheduling systems that assist in the management of scheduling information. An overview of these firms and their products is given elsewhere (Musier and Evans, 1990). These systems rely heavily on the input of the human production scheduler to obtain a feasible schedule. None of these systems has optimization features capable of handling the complex production constraints occurring in most process plants.

Based on a recent survey of production scheduling problems occurring in the process industries (Musier and Evans, 1989),

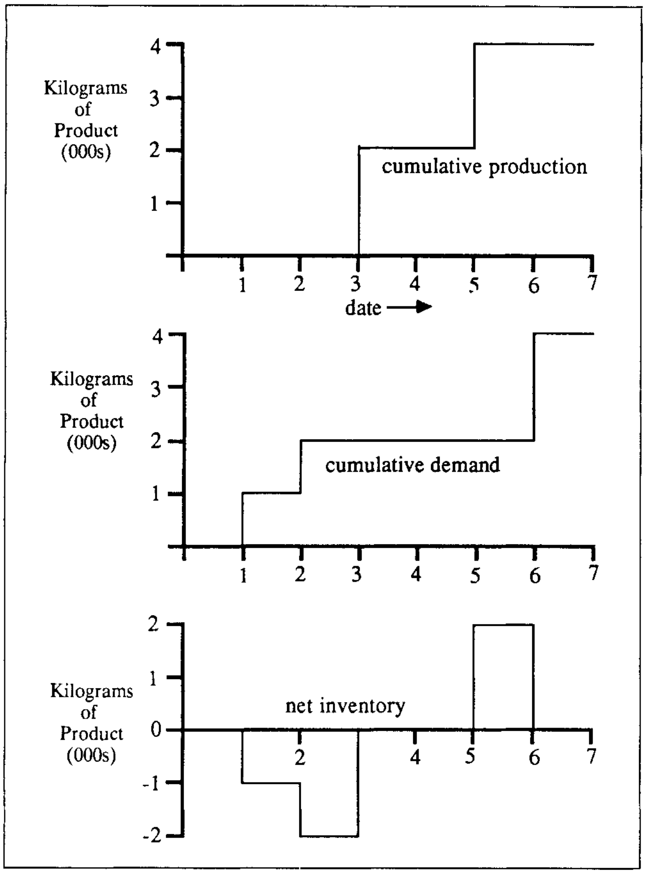

Figure 1. Production, demand, and net inventory curves for a single product.

this article focuses on schedule optimization for problems involving a single stage of nonidentical process units. Plants with sets of reactors, blenders, or distillation columns, which operate in parallel, occur frequently in industry. A less obvious example of a parallel unit process comprises a set of five extrusion lines, each consisting of a blender, extruder, quench bath and pelletizer. These may be considered as a single set of five parallel process units, since each equipment group processes only one product at a time and the production quantity is determined by the batch size of the blending vessel.

The survey also found that the most important goal in the short-term scheduling of industrial processes is to satisfy customer orders on time. This article, therefore, focuses on the objective of minimizing the total time for which all products are out-of-stock. If a given product is out-of-stock, then one or several customer orders cannot be filled on time. If no product is out-of-stock, then no customer orders are late.

Table 1. Production Unit Sizes

Unit	Size (L)
1	1,000
2	1,200
3	1,400
4	1,200
5	800

Table 2. Initial Unit Status

Unit	Time Available	Previous Product
1	3	100
2	2	200
3	2	300
4	3	400
5	3	500

Table 3. Assignment of Production Capacity

Unit	Prod.	Start	Finish	Process Time	Size (kg)
1	400	3	8	5	3,892
1	500	8	13	5	3,939
1	600	13	19	6	3,946
1	100	19	25	6	4,118
2	800	2	6	4	4,715
2	700	6	11	5	5,003
2	600	11	16	5	4,851
2	300	16	22	6	4,692
2	600	22	27	5	4,851
3	600	2	8	6	5,345
3	200	8	13	5	5,557
3	400	13	20	7	5,702
3	100	20	27	7	5,855
4	200	3	10	7	4,782
4	300	10	16	6	4,680
4	700	16	24	8	4,859
5	400	3	7	4	3,281
5	700	7	11	4	3,205
5	700	11	15	4	3,205
5	300	15	20	5	3,249
5	600	20	24	4	3,112
5	700	24	28	4	3,205

Table 4. Batch Processing Time

	Unit				
Product	1	2	3	4	5
100	6	0	7	0	0
200	0	4	5	7	0
300	0	6	6	6	5
400	5	0	7	6	4
500	5	0	0	0	6
600	6	5	6	0	4
700	0	5	0	8	4
800	0	4	0	0	0

Table 5. Batch Sizes

	kg/Product Batch/Unit				
Product	1	2	3	4	5
100	4,118	0	5,855	0	0
200	0	4,909	5,557	4,782	0
300	0	4,692	5,816	4,680	3,249
400	3,892	0	5,702	4,567	3,281
500	3,939	0	0	0	3,104
600	3,946	4,851	5,345	0	3,112
700	0	5,003	0	4,859	3,205
800	0	4,715	0	0	0

Table 6. Cleanout Matrix

Previous Product	Next Product 100	200	300	400	500	600	700	800
100	0	2	2	0	2	2	2	0
200	2	0	0	0	2	0	2	0
300	2	2	0	2	2	0	0	2
400	0	0	0	0	0	2	0	2
500	2	2	0	0	0	0	2	2
600	0	0	0	2	2	0	0	0
700	0	2	0	2	2	0	0	2
800	2	2	2	2	2	2	0	0

The previous literature in the area of production scheduling is of limited help in addressing the problem of industrial scheduling with lot sizing. Most previous work has focused on the problem of sequencing a fixed number of batches and is reviewed elsewhere (Musier and Evans, 1989). The fact that customer orders are, in general, either larger or smaller than the batch size of the plant's process units is usually ignored. Thus, the number of batches that needs to be scheduled to satisfy a slate of customer orders is a variable, since the process units have different capacities. Consequently, detailed bookkeeping of the inventory level of each product over time is unavoidable.

For example, the scheduler must decide whether to produce two batches on the 2,000-kg unit or one batch on the 3,500-kg unit to fill a 3,000-kg order. In the first instance, 1,000 kg of excess product must be stored as inventory, while in the second case only 500 kg is placed into inventory. Given that there may be 50 other customer orders to consider and eight process units to be scheduled, it is not obvious which decision will result in the best overall production schedule.

In an excellent review of production scheduling, Graves (1981) defined lot sizing as the "determination of run quantities." Much of the previous work in lot sizing has investigated the trade-off between changeover costs and inventory holding costs. While these costs are not considered explicitly in this article, they are considered indirectly, since the scheduling goal is to satisfy customer orders on time. In a highly utilized plant, avoidable product changeovers must be minimized to satisfy orders on time, as well as the production and stocking of inventory far ahead of its due date. Changeover and inventory policies are determined to minimize the scheduling objective.

Graves goes on to note that in most previous work "sequencing decisions are assumed to be made after the lot sizes have been determined." In industrial process plants, however, the sequencing and lot-sizing decisions must be made simultaneously, resulting in complex optimization problems. Most of these formulations are NP-complete, indicating that the time required to guarantee that an optimal solution is found cannot be bounded by a polynomial in the characteristic size of the problem (Graham et al., 1979). Since the rigorous solution of these problems is intractable, industrial-scale problems must be solved using approximate methods.

Table 7. Cumulative Production Curves with Product Release Dates

Time	Product 100	200	300	400	500	600	700	800
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	4,715
7	0	0	0	3,281	0	0	0	4,715
8	0	0	0	7,173	0	5,345	0	4,715
9	0	0	0	7,173	0	5,345	0	4,715
10	0	4,782	0	7,173	0	5,345	0	4,715
11	0	4,782	0	7,173	0	5,345	8,208	4,715
12	0	4,782	0	7,173	0	5,345	8,208	4,715
13	0	10,339	0	7,173	3,939	5,345	8,208	4,715
14	0	10,339	0	7,173	3,939	5,345	8,208	4,715
15	0	10,339	0	7,173	3,939	5,345	11,413	4,715
16	0	10,339	4,680	7,173	3,939	10,196	11,413	4,715
17	0	10,339	4,680	7,173	3,939	10,196	11,413	4,715
18	0	10,339	4,680	7,173	3,939	10,196	11,413	4,715
19	0	10,339	4,680	7,173	3,939	14,142	11,413	4,715
20	0	10,339	7,929	12,875	3,939	14,142	11,413	4,715
21	0	10,339	7,929	12,875	3,939	14,142	11,413	4,715
22	0	10,339	12,621	12,875	3,939	14,142	11,413	4,715
23	0	10,339	12,621	12,875	3,939	14,142	11,413	4,715
24	0	10,339	12,621	12,875	3,939	17,254	16,272	4,715
25	4,118	10,339	12,621	12,875	3,939	17,254	16,272	4,715
26	4,118	10,339	12,621	12,875	3,939	17,254	16,272	4,715
27	9,973	10,339	12,621	12,875	3,939	22,105	16,272	4,715
28	9,973	10,339	12,621	12,875	3,939	22,105	19,477	4,715
29	9,973	10,339	12,621	12,875	3,939	22,105	19,477	4,715
30	9,973	10,339	12,621	12,875	3,939	22,105	19,477	4,715
Dates Release	18	3	10	1	7	2	3	2

Table 8. Customer Orders

Order	Due Date	kg	Product
1	6	3,500	800
2	6	1,215	800
3	7	3,281	400
4	8	3,892	400
5	8	5,345	600
6	10	4,782	200
7	11	2,000	700
8	11	6,208	700
9	13	3,000	200
10	13	2,000	200
11	13	557	200
12	13	3,939	500
13	15	3,205	700
14	16	1,000	300
15	16	3,680	300
16	16	1,000	600
17	16	3,000	600
18	16	851	600
19	19	3,946	600
20	20	1,000	300
21	20	2,249	300
22	20	1,500	400
23	20	1,500	400
24	20	1,000	400
25	20	1,702	400
26	22	4,500	300
27	22	192	300
28	24	3,112	600
29	24	1,500	700
30	24	3,000	700
31	24	359	700
32	25	4,118	100
33	27	4,500	100
34	27	1,355	100
35	27	4,851	600
36	28	3,000	700
37	28	205	700

Table 10. Initial Schedule

Unit	Batch Label	Product	kg	Start Time	Finish Time	Process Time
1	12	400	3,892	3	8	5
1	26	500	3,939	8	13	5
1	9	100	4,118	18	24	6
2	4	800	4,715	2	6	4
2	10	600	4,851	8	13	5
2	15	300	4,692	13	19	6
2	16	600	4,851	19	24	5
2	31	700	5,003	24	29	5
3	3	600	5,345	2	8	6
3	8	200	5,557	8	13	5
3	5	600	5,345	13	19	6
3	18	100	5,855	19	26	7
3	14	400	5,702	26	33	7
3	28	100	5,855	33	40	7
4	1	200	4,782	3	10	7
4	2	300	4,680	10	16	6
4	24	200	4,782	18	25	7
4	27	700	4,859	27	35	8
5	11	400	3,281	3	7	4
5	6	300	3,249	10	15	5
5	7	700	3,205	15	19	4
5	13	600	3,112	19	23	4
5	17	700	3,205	23	27	4
5	19	300	3,249	27	32	5
5	20	600	3,112	32	36	4
5	21	700	3,205	36	40	4
5	22	700	3,205	40	44	4
5	23	400	3,281	46	50	4
5	25	700	3,205	50	54	4
5	29	500	3,104	56	62	6
5	30	600	3,112	62	66	4
5	32	600	3,112	66	70	4

Model Formulation

The process plant consists of M parallel process units where each unit k has a standard capacity, b_k (in L, kg, etc.). The plant produces P products. Not all products may be produced on every unit. For example, if a plant consists of five blenders and only blenders nos. 1 and 2 have feeders for flame-retardant additives, then products requiring these additives may be produced only on blenders nos. 1 and 2, which are referred to as the set of allowable units for this product. Every product i has an associated set of allowable units M_i.

For every unit k in M_i, there is an associated batch processing time t_{ik} and a yield factor y_{ik}, which specifies the quantity of desired product obtained per unit capacity of unit k. Thus, the batch processing time may be both product- and unit-dependent. This also holds true for the batch yield for each product on each unit.

All products have a release date, r_i, which corresponds to the earliest time at which a batch of product i can begin processing on any unit. For example, if the catalyst for product i will not be delivered to the plant until the 10th of the month, then no batch of that product can begin processing before the 10th: $r_i = 10$.

When a batch of product j follows a batch of product i, a cleanout time c_{ij} must be scheduled between batches. An initial inventory I_i of each product is also specified input data.

Table 9. Required Number of Batches

Product	Min. Yield (kg)	Max. Yield (kg)	Demand (kg)	Initial Inventory	Min. Batch	Max. Batch
100	4,118	5,855	9,973	0	2	3
200	4,782	5,557	10,339	0	2	3
300	3,249	5,816	12,621	0	3	4
400	3,281	5,702	12,875	0	3	4
500	3,104	3,939	3,939	0	1	2
600	3,112	5,345	22,105	0	5	8
700	3,205	5,003	19,477	0	4	7
800	4,715	4,715	4,715	0	1	1

Table 11. Net Inventory Curves for Initial Solution

	Product							
Time	100	200	300	400	500	600	700	800
1–10	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	−8,208	0
12	0	0	0	0	0	0	−8,208	0
13	0	0	0	0	0	4,851	−8,208	0
14	0	0	0	0	0	4,851	−8,208	0
15	0	0	3,249	0	0	4,851	−11,413	0
16	0	0	3,249	0	0	0	−11,413	0
17	0	0	3,249	0	0	0	−11,413	0
18	0	0	3,249	0	0	0	−11,413	0
19	0	0	7,941	0	0	1,399	−8,208	0
20	0	0	4,692	−5,702	0	1,399	−8,208	0
21	0	0	4,692	−5,702	0	1,399	−8,208	0
22	0	0	0	−5,702	0	1,399	−8,208	0
23	0	0	0	−5,702	0	4,511	−8,208	0
24	4,118	0	0	−5,702	0	6,250	−13,067	0
25	0	4,782	0	−5,702	0	6,250	−13,067	0
26	5,855	4,782	0	−5,702	0	6,250	−13,067	0
27	0	4,782	0	−5,702	0	1,399	−9,862	0
28	0	4,782	0	−5,702	0	1,399	−13,067	0
29	0	4,782	0	−5,702	0	1,399	−8,064	0
30	0	4,782	0	−5,702	0	1,399	−8,064	0
31	0	4,782	0	−5,702	0	1,399	−8,064	0
32	0	4,782	3,249	−5,702	0	1,399	−8,064	0
33	0	4,782	3,249	0	0	1,399	−8,064	0
34	0	4,782	3,249	0	0	1,399	−8,064	0
35	0	4,782	3,249	0	0	1,399	−3,205	0
36	0	4,782	3,249	0	0	4,511	−3,205	0
37	0	4,782	3,249	0	0	4,511	−3,205	0
38	0	4,782	3,249	0	0	4,511	−3,205	0
39	0	4,782	3,249	0	0	4,511	−3,205	0
40	5,855	4,782	3,249	0	0	4,511	0	0
41	5,855	4,782	3,249	0	0	4,511	0	0
42	5,855	4,782	3,249	0	0	4,511	0	0
43	5,855	4,782	3,249	0	0	4,511	0	0
44	5,855	4,782	3,249	0	0	4,511	3,205	0
45	5,855	4,782	3,249	0	0	4,511	3,205	0
46	5,855	4,782	3,249	0	0	4,511	3,205	0
47	5,855	4,782	3,249	0	0	4,511	3,205	0
48	5,855	4,782	3,249	0	0	4,511	3,205	0
49	5,855	4,782	3,249	0	0	4,511	3,205	0
50	5,855	4,782	3,249	3,281	0	4,511	3,205	0
51	5,855	4,782	3,249	3,281	0	4,511	3,205	0
52	5,855	4,782	3,249	3,281	0	4,511	3,205	0
53	5,855	4,782	3,249	3,281	0	4,511	3,205	0
54	5,855	4,782	3,249	3,281	0	4,511	6,410	0
55+	5,855	4,782	3,249	3,281	0	4,511	6,410	0

N customer orders are to be scheduled. For each order n, the product type p_n, product quantity q_n, and due data d_n are specified. We note that for plants that do not make to order, these production orders may, for example, correspond to weekly production goals for each specified product. We refer to customer orders, even though the model and solution method may be used in cases involving other kinds of production requirements.

The scheduling horizon time T is determined by the latest due date for any product. In general, each process unit k may currently be processing a batch of product o_k and may not be available to begin processing a new batch until time e_k. We assume that processing times, cleanout times, and due dates are integer-valued. In summary, the problem specification consists of 12 data sets:

Unit specifications	b_k	$k=1,..,P$
Product specifications	M_i, t_{ik}, y_{ik}, r_i	$i=1,..,P$
Cleanout requirements	c_{ij}	$j=1,..,P$
Order information	p_n, q_n, d_n	$n=1,..,N$
Initial conditions	e_k, o_k, I_i	

The scheduling objective (to minimize the total time during which all products are out-of-stock) is calculated as the total time during which the net inventory for each product is negative. This is illustrated for a single product in Figure 1. The net inventory curve is obtained by subtracting the cumulative demand curve from the cumulative production curve. Each step in the cumulative demand curve corresponds to the due date of one of more customer orders, while the height of the

Table 12. Final Schedule

Unit	Batch Label	Product	kg	Start Time	Finish Time	Process Time
1	12	400	3,892	3	8	5
1	26	500	3,939	8	13	5
1	5	600	3,946	13	19	6
1	9	100	4,118	19	25	6
2	4	800	4,715	2	6	4
2	17	700	5,003	6	11	5
2	10	600	4,851	11	16	5
2	15	300	4,692	16	22	6
2	16	600	4,851	22	27	5
3	3	600	5,345	2	8	6
3	8	200	5,557	8	13	5
3	14	400	5,702	13	20	7
3	18	100	5,855	20	27	7
3	6	300	5,816	29	35	6
3	28	100	5,855	37	44	7
4	1	200	4,782	3	10	7
4	2	300	4,680	10	16	6
4	31	700	4,859	16	24	8
4	24	200	4,782	26	33	7
5	11	400	3,281	3	7	4
5	27	700	3,205	7	11	4
5	7	700	3,205	11	15	4
5	19	300	3,249	15	20	5
5	30	600	3,112	20	24	4
5	21	700	3,205	24	28	4
5	13	600	3,112	28	32	4
5	29	500	3,104	34	40	6
5	25	700	3,205	42	46	4
5	22	700	3,205	46	50	4
5	32	600	3,112	50	54	4
5	20	600	3,112	54	58	4
5	23	400	3,281	60	64	4

step is determined by the size of the order(s). Similarly, each step in the cumulative production curve corresponds to the completion time of one or more batches, and the height of the step is determined by the batch quantity.

To allow a valid comparison of problems of different sizes, results will be presented in terms of a dimensionless objective function:

$$\text{Objective} = \frac{\text{Total stockout time}}{\text{Total production time}}$$

The total production time is defined as T^*M.

Solution Method for Scheduling with Simultaneous Lot Sizing

The heuristic improvement method (HIM) used here is an extension of an approach which has been successfully used to optimize production schedules for single-stage process plants, in which the production quantity equals the order quantity (Musier and Evans, 1989). As mentioned previously, this is not true in general since the size of a customer order will be either larger or smaller than the plant's batch quantities. Consequently, the problem addressed here is considerably more complex than the earlier one, since the optimal inventory level of each product is to be determined at each point in time.

The method involves several steps. The first is to determine the number of batches of each product which must be produced to guarantee that all demand may be satisfied. The second step is to determine an initial schedule in which all batches are scheduled. The final step is the evolutionary improvement of the initial schedule until certain conditions for optimality are satisfied and a local optimum is found.

Since the required number of batches of each product will depend on the position of those batches in the schedule, the first step of the algorithm involves the determination of the upper bound (N_U) and lower bound (N_L) on the required number of batches of each product. The need for this step arises because processing units are nonidentical and the total production quantity of each product depends on the unit processing each batch of product. For example, if product A can be produced on either a 1,000-kg or 1,500-kg batch unit and the total demand for A is 7,500 kg, then the upper bound on the total required number of batches of A is eight. If eight batches are produced, then enough production is being scheduled to fill all demand for A irrespective of which unit is processing the batches. (Obviously, the lower bound on the number of batches is five if all batches are processed on the 1,500-kg unit.) To guarantee that all demand can be satisfied, the number of batches carried along by the algorithm is eight, even though fewer than eight batches may be needed to meet demand if several batches are scheduled on the larger unit. Any batches that appear in the final optimized schedule after all demand for A has been met are pruned from the schedule. In summary, the number of batches of each product i equals the total demand of i divided by the batch size of the smallest unit k in M_i, rounded up to the nearest integer.

Once the number of batches is determined for each product, an initial schedule is determined using one of three best fit methods (BFM). Each BFM uses a different method of sorting the batches. Batches are considered in the sort sequence and inserted into the schedule at the point that reduces the value of the objective function most. For the next batch i, the objective is evaluated for all possible schedule positions (prior to each of the previously scheduled batches and at the end of the schedule for each allowable unit). Once the best position is determined, the batch i is inserted into the current schedule where it "fits best." The insertion process is repeated for the next batch in the list and continues until all batches have been scheduled.

Three different sorting criteria are considered. In the first case (designated MULTI), each batch is assigned a random rank number between 1 and 10,000, and the batches are sorted according to nondecreasing rank. Multiple initial solutions may easily be generated using this method. In the second case (designated NAU), batches are sorted according to the nondecreasing number of allowable units on which the product may be produced. The idea is that those batches which can be produced on the fewest number of units will be considered first. In the third case (designated PDD), each batch is assigned a rank equal to the time at which the cumulative demand for that product is greater than or equal to the smallest batch size for that product. This is called the pseudo due data (PDD) for that batch. Once such a time is found, the cumulative demand is reduced by the smallest batch size for that product. The PDD assignment continues until all batches have been assigned a rank. Then, the batches are sorted according to nondecreasing PDD. The rationale behind this ordering is that it is an

Table 13. Net Inventory Curves for Final Solution

Time	Product 100	200	300	400	500	600	700	800
1-31	0	0	0	0	0	0	0	0
32	0	0	0	0	0	3,112	0	0
33	0	4,782	0	0	0	3,112	0	0
34	0	4,782	0	0	0	3,112	0	0
35	0	4,782	5,816	0	0	3,112	0	0
36	0	4,782	5,816	0	0	3,112	0	0
37	0	4,782	5,816	0	0	3,112	0	0
38	0	4,782	5,816	0	0	3,112	0	0
39	0	4,782	5,816	0	0	3,112	0	0
40	0	4,782	5,816	0	3,104	3,112	0	0
41	0	4,782	5,816	0	3,104	3,112	0	0
42	0	4,782	5,816	0	3,104	3,112	0	0
43	0	4,782	5,816	0	3,104	3,112	0	0
44	5,855	4,782	5,816	0	3,104	3,112	0	0
45	5,855	4,782	5,816	0	3,104	3,112	0	0
46	5,855	4,782	5,816	0	3,104	3,112	3,205	0
47	5,855	4,782	5,816	0	3,104	3,112	3,205	0
48	5,855	4,782	5,816	0	3,104	3,112	3,205	0
49	5,855	4,782	5,816	0	3,104	3,112	3,205	0
50	5,855	4,782	5,816	0	3,104	3,112	6,410	0
51	5,855	4,782	5,816	0	3,104	3,112	6,410	0
52	5,855	4,782	5,816	0	3,104	3,112	6,410	0
53	5,855	4,782	5,816	0	3,104	3,112	6,410	0
54	5,855	4,782	5,816	0	3,104	6,224	6,410	0
55	5,855	4,782	5,816	0	3,104	6,224	6,410	0
56	5,855	4,782	5,816	0	3,104	6,224	6,410	0
57	5,855	4,782	5,816	0	3,104	6,224	6,410	0
58	5,855	4,782	5,816	0	3,104	9,336	6,410	0
59	5,855	4,782	5,816	0	3,104	9,336	6,410	0
60	5,855	4,782	5,816	0	3,104	9,336	6,410	0
61	5,855	4,782	5,816	0	3,104	9,336	6,410	0
62	5,855	4,782	5,816	0	3,104	9,336	6,410	0
63	5,855	4,782	5,816	0	3,104	9,336	6,410	0
64+	5,855	4,782	5,816	3,281	3,104	9,336	6,410	0

attempt to match the sequence in which various product quantities are required with the sequence in which batches should be produced.

After generating an initial schedule using the BFM, an evolutionary strategy is implemented, referred to as a heuristic improvement method (HIM). The strategy is based on the fact that in an optimal schedule, moving any batch from its current position in the production schedule to a new position in the schedule will not reduce the value of the objective function. Similarly, a pairwise interchange of batch positions will not reduce the value of the objective, since the schedule is optimal.

The aim of the evolutionary strategy is to determine a schedule in which no single-batch repositioning or pairwise interchange of batch positions will improve the current schedule.

Table 14. Performance of Best-Fit Methods

Horizon	No. of Batches	Avg. Perf. of HIM Using Sorting Heuristics (% as Best Heuristic) PDD	NAU	MULTI
30	42	0.0363 (65)	0.0617 (25)	0.0490 (35)
35	50	0.0395 (55)	0.0839 (15)	0.0825 (30)
40	56	0.0381 (70)	0.0889 (10)	0.0769 (30)
45	65	0.0843 (60)	0.1510 (10)	0.1300 (30)
50	72	0.0714 (70)	0.1388 (10)	0.1168 (20)

The premise is that a schedule, which shares these properties with a globally optimal schedule and is locally optimal with respect to these criteria, will have an objective value close to that of the globally optimal solution and can be obtained with a reasonable amount of computational effort.

The HIM employs two subroutines to perform (1) the single-batch repositioning (EVOLVE-1) and (2) the pairwise interchange of batch positions (EVOLVE-2). Each subroutine performs a single pass through the current schedule. If a schedule modification reduces the objective value, the repositioning or interchange is retained and the process continues until each batch or batch pair has been checked once. The subroutines are implemented alternatively (EVOLVE-1, EVOLVE-2, EVOLVE-1, etc.) until no further improvement in the current schedule is found. The final schedule is then locally optimal with respect to these criteria.

For each subroutine, the number of repositionings or interchanges investigated by the algorithm is $O(n^2)$, where n is the number of batches. In practice, the computational effort is reduced for several reasons:

- In EVOLVE-2, the evaluation of an interchange of batches of the same product is unnecessary.
- If a batch pair consists of different products, their interchange need not be investigated if the current processing unit of one is not in the set of allowable units for the other.

The number of evaluations in EVOLVE-1 is similarly reduced for several reasons:

- A batch of product i can only be repositioned to allowable unit k.
- It is unnecessary to investigate the repositioning of a batch before a batch of the same product.
- If a batch is removed from the schedule (to be repositioned elsewhere) and the value of the objective is reduced, then the batch is positioned at the end of the schedule for the same process unit.
- If the value of the objective increases when the batch is removed, then the only new positions investigated are those for which the batch begins processing prior to its old completion time. This works well in practice since the value of the objective is most likely to be reduced if the batch is produced earlier in time. This heuristic reduces the computational effort by approximately 50%.

The evaluation of the impact of each batch repositioning or interchange requires updating the objective function. In turn, this requires that the net inventory curves of each product affected by the schedule modification be updated. The integrality of the time-related problem data allows the use of a data structure in which the inventory level of each product is specified at each integral time period. If $t=T$ is the integral time horizon that begins at $t=1$, then the number of operations required to update the entire inventory and evaluate the objective value grows as $O(T)$ for both the single batch repositioning and the batch pair position interchange procedures. It is noteworthy that through the use of this data structure, the number of operations is not a function of the number of orders, the number of batches, the number of process units, or the number of products.

Test Problems

A series of test problems were generated and solved for examples in which zero stockout schedules are known to exist. Test problems were generated in which the plant is highly utilized (i.e., an optimal schedule exists with no idle time between batches). Since the goal is to demonstrate the effectiveness of the approach on typical industrially-sized problems, we have selected a base case with eight process units, 20 products, and a horizon varying from 25 to 50 days. This results in problems with 35 to 70 batches. The data for these problems were generated such that approximately 50% of the product sequences require a cleanout and the average product can be produced on approximately 50% of the process units. Batch processing times were selected to vary by as much as 100% from unit to unit and from product to product. Test problems representative of industrial-scale problems are generated as follows:

1. The number of units (M), the number of products (P), and the maximum horizon time (T) are selected.
2. A unit size of 800, 1,000, 1,200 or 1,400 L is randomly assigned to each of the process units.
3. A "base" processing time of 4, 5 or 6 days is randomly assigned to each of the process units.
4. The processing time for each product on each unit is determined by randomly adding 0, 1, or 2 to the "base" processing time for each unit (determined in step 3). Thus, batch processing times vary between 4 and 8 days and are product and unit-dependent.
5. A base chronological time is arbitrarily selected as $t=1$. All units may not be available for processing new batches until the current batches are finished. Consequently, the time at which each unit initially becomes available is randomly selected from $\{1,2,3,4\}$, as is the previous product for each unit.
6. The yield per batch is selected for each product on each unit and is the product of the size of the unit (in L) and a yield factor which is randomly selected between 3.8 and 4.2 (in kg of final product per L of reactor volume). This allows the determination of the batch size of each product on each allowable unit.
7. The initial inventory of each product is set at zero, although it could take on a nonzero value without loss of generality.
8. A schedule S is randomly generated in which batches are assigned to each unit until the specified production horizon time is exceeded. Cumulative production inventory curves are generated for each product corresponding to S.
9. Zeros are randomly inserted in the processing time matrix (for product-unit assignments which do not appear in S) until the fraction of allowable units is less than 0.50, if possible. The zeros indicate that a given product may not be produced on a given unit.
10. Every product sequence is initially assigned a cleanout time of 2 days. This value is arbitrarily picked to be 1/3 of the average batch processing time. Sequences in S are assigned a cleanout time of 0 in the cleanout matrix. Then, the fraction of product sequences requiring a cleanout is evaluated. If the fraction is greater than 0.50, go to step 11. If the fraction is less than 0.50, a time of 2 is randomly assigned to product sequences in the cleanout matrix until the total fraction of sequences requiring a cleanout greater than 0.50.
11. The release date is assigned for each product i and is equal to the completion time of the first batch of product i in S minus the largest process time for i on any unit.
12. Customer orders for each product i are selected as follows. An order size x between 1,000 and 7,500 kg (in increments of 500 kg) is selected. The inventory curve is searched (from $t=1$ to $t=T$) until a time t is found at which the cumulative inventory of i is greater than x. If such a time is found before the end of the horizon, then an order for x kg of i with a due date t is generated. The quantity x is subtracted from the cumulative inventory curve. If such a time is not found, then an order with a due date T is generated with an order quantity equal to the remaining inventory of i.
13. All orders are sorted by increasing due date.

Example Problem

The solution approach is illustrated using a small example problem with five process units, eight products, and a time horizon of 30 days. A problem specification is generated for which a zero stockout solution exists using the method outlined previously. The data describing the generated example problem are given in Tables 1 to 8. The sizes of the production units are listed in Table 1, while the initial status of each unit is shown in Table 2. Note that the eight products are labeled 100 to 800. In Table 3, batches have been assigned to the production schedule. Customer orders will then be determined such that

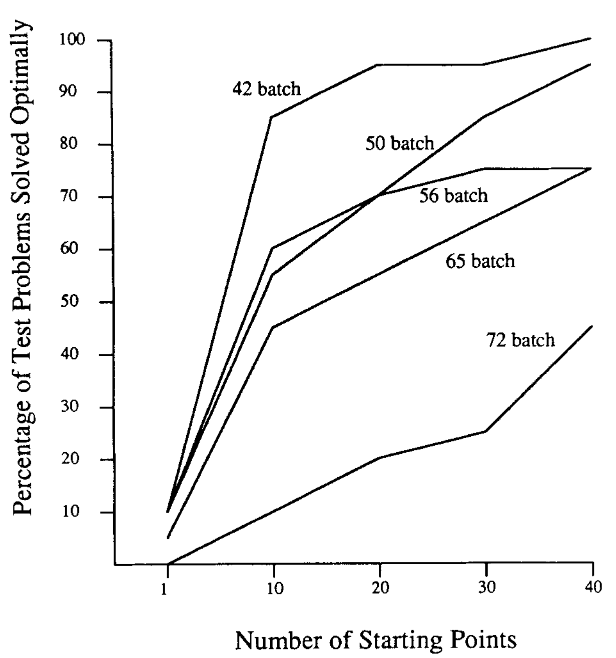

Figure 2. Algorithmic performance vs. computational requirements.

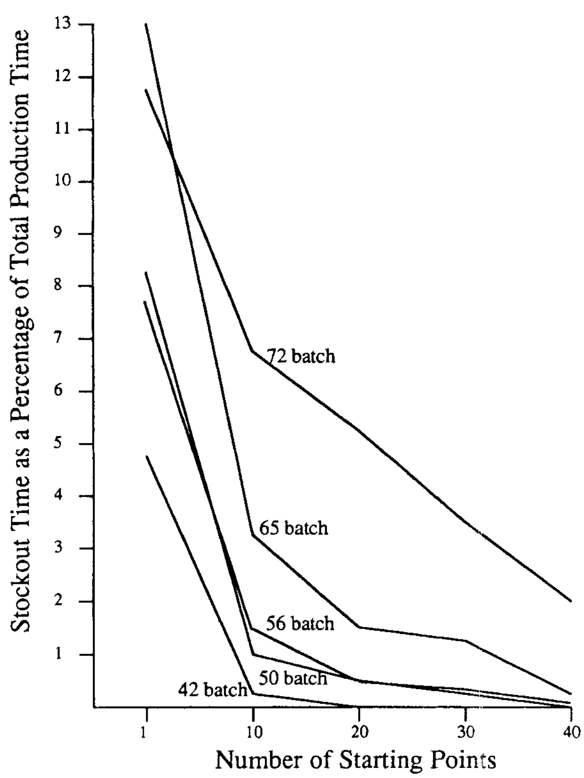

Figure 3. Average performance of the HIM.

all orders may be satisfied on time by the schedule of Table 3. Batch processing times and batch sizes are given in Tables 4 and 5. The cleanout matrix is given in Table 6. Table 7 lists the cumulative production curve corresponding to the schedule of Table 3, while customer orders are given in Table 8. This completes the specification of the problem data.

In Table 9, the minimum batch size, maximum batch size, and the total demand for each product are listed. Based on these, the minimum and maximum numbers of batches of each product are calculated. The algorithm manipulates the maximum number of batches during schedule optimization.

The initial schedule is shown in Table 10, while the net inventory curve corresponding to this schedule is shown in Table 11. The schedule was determined using the BFM with the MULTI sorting. (The batch labels are arbitrary and used for bookkeeping purposes.) Initially there are 42 total days of product stockout: product 400 is out of stock for 13 days while product 700 is out of stock for 29 days. All other products are never out of stock. The total production time is 150-unit days so the value of the objective is 0.28. In other words, the stockout time is 28% of the total production time.

The first pass through the schedule (using EVOLVE-1) reduces the stockout time from 42 to 17 days. The next pass through the schedule (using EVOLVE-2) further reduces the stockout time from 17 to 0 days. The optimal zero stockout solution is shown in Table 12 with the corresponding net inventory curve given in Table 13. Note that its schedule is the same as that in Table 3, out to a time of 30 days. The net inventory is zero out to 31 days and takes on nonzero values for some products thereafter. In this case, the algorithm has found an optimal schedule that is the same as the schedule generated in the formulation of the problem statement. In general, however, these scheduling problems may have multiple, globally-optimal solutions.

Batches that finish processing at times greater than 30 are excess batches and may be pruned from the schedule since they are not required to fill any orders. Recall that inclusion of the batches during the optimization procedure, however, was necessary to maintain the feasibility of the schedule.

Results

Because it is convenient to present the performance of the HIM as a function of the number of batches being scheduled, we define the batch number to be equal to $(N_U + N_L)/2$ rather than N_U. Although N_U batches are manipulated by the algorithm, some of these may be designated as excess batches that are pruned from the final solution. Defining the batch number in this way allows us to avoid overstating the size of the problem being solved.

In Table 14, performance of the HIM is evaluated using the three different sorting criteria for generating the best-fit initial solution. Results are presented for test problems with 42 to 72 batches using the PDD, NAU and MULTI methods. Each point is the average performance on 20 test problems. In Table 14, each entry in parentheses is the percentage of instances in which the specified sorting criterion results in the optimized schedule with the lowest objective value. Initial solutions generated by the BFM using the PDD sorting criterion result in optimized schedules which are superior to those obtained using the NAU and MULTI criteria in 55% to 70% of the test problems. Objective function values ranged from 0.0363 (for 42-batch problems) to 0.0714 (for 72-batch problems). In other words, the total stockout time is equal to 7.14% of the total production time for the largest problems.

For the case in which each customer requires a different product, the objective may be interpreted as the percentage of

Table 15. Computational Requirements

Time Horizon, T	30	35	40	45	50
Avg. No. of Batches, n	42	50	56	65	72
Avg. Comput. Time, t (cpu s)	40	84	141	198	308
Avg. No. of Alternatives Investigated, a	3,377	6,064	9,055	11,085	15,163
Avg. Time per Alternative, t/a (10^{-2} s)	1.19	1.39	1.56	1.79	2.03
Scaled Time per Alternative, $t/(a^*T)$ (10^{-4} s).	3.93	3.96	3.90	3.97	4.07
Scaled No. of Alternatives, $a/(n^*n)$	3.75	4.95	5.66	5.47	6.07

the production time for which the average customer order is late. For example, if the average batch processing time is 1 day, then an objective value of 0.0714 means that each order is 1:41 hours late on average. This is near-optimal performance in an industrial environment. Note that the objective function is not bounded by 1.00, but could in general take on values greater than 100% of the total production time.

Thus far, we have discussed cases in which each test problem was solved from a single initial solution. However, using the MULTI sorting criterion allows the generation of multiple starting points for each set of test data. The use of multiple starting points allows the HIM to converge to multiple local optima, often finding globally optimal zero stockout solutions. The percentage of test problems solved optimally is shown as a function of the number of starting points in Figure 2 for test problems with 42 to 72 batches. These results clearly illustrate the trade-off between computational effort and solution quality. Even though the PDD criterion was the best single-starting-point method, the use of ten starting points (generated using MULTI) resulted in superior schedules in every case. In fact, Figure 2 shows that 45% to 100% of the problems were solved optimally (zero stockout) when 40 starting points were used.

Average performance of the HIM as a function of the number of starting points is shown in Figure 3. Again, performance increases with the number of starting points and generally decreases with increasing problem size. This may be understood qualitatively through the combinatorial analysis of a special case in which any batch may be processed on any unit and all batches are unique. If the number of batches (N) and the number of units (M) are fixed, then the total number of unique schedules (Q) is given by:

$$Q=\frac{(N+M-1)!}{(M-1)!}$$

For plants with eight process units ($M=8$):

No. of Batches (N)	No. of Possible Schedules (Q)
20	2.16×10^{24}
30	2.73×10^{39}
40	5.13×10^{55}
50	8.04×10^{72}
60	7.23×10^{90}

For each ten batch increase in problem size, the number of possible unique schedules increases by more than 15 orders of magnitude. The number of locally optimal solutions will increase as well. Even though multiple solutions may be locally optimal with respect to single-batch repositioning and batch pair interchange, the objective values corresponding to these locally optimal schedules may be very different.

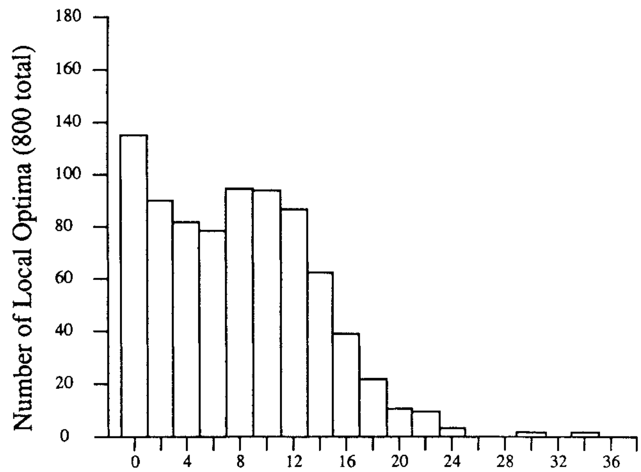

Figure 4. Distribution of local optima for 50 batch problems.

To better understand the nature of this scheduling problem, we have studied the way in which locally optimal solutions are distributed for different-sized problems. Figures 4 and 5 show the frequency distributions of local optima for 20 test problems, each of which is solved using 40 initial solutions—800 local optima in all. In Figure 4 results are plotted for 50-batch problems, while results for 72-batch problems are plotted in Figure 5. The change in the distribution of local optima with increasing problem size is dramatic. In Figure 5 the distribution of local optima resembles a normal distribution with a peak at 14%, while in Figure 4 the distribution is skewed toward lower objective values with a peak at 8% and a large cluster of optimal solutions. Analysis of the 43, 56, and 65 batch

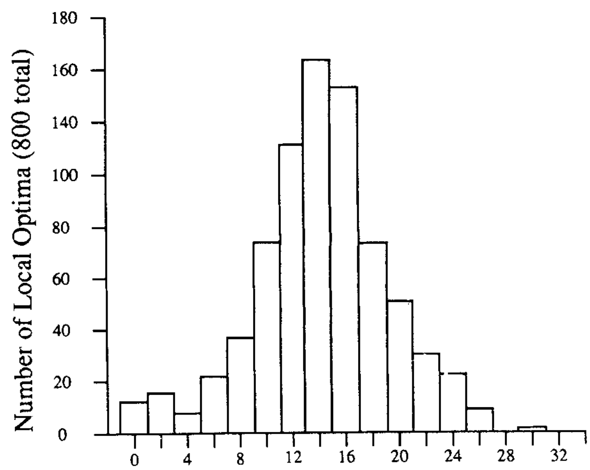

Figure 5. Distribution of local optima for 72 batch problems.

problems is consistent with the trend shown in Figures 4 and 5: As problem size increases, the size of the peak of local optimal solutions decreases, while the size of a second peak increases and becomes skewed toward higher objective values. Thus, the reason that performance of the HIM decreases for large problems is that the distribution of local optima changes. More precisely, a larger fraction of schedules cannot be improved by single-batch repositioning or batch pair interchanges and is also not globally optimal. Since the number of "bad" local optima appears to increase with problem size at a rate faster than that of "good" local optima, a lower probability exists in which one of the 40 local optima obtained by the HIM will correspond to the globally optimal solution.

Results on computation time (using a Digital VAXstation II) and the number of alternative schedules investigated by the HIM are given in Table 15 for problems of various sizes solved from a single initial starting point. Average results are based on 20 test problems solved using 40 starting points for each time horizon.

As expected, computation time increases with increasing problem size. This is due to: (1) an increase in the number of alternative schedules considered and (2) an increase in the computation time to evaluate each alternative. As mentioned previously, the computation time per alternative grows as $O(T)$. This is consistent with the results shown in Table 15, in which the scaled time per alternative is constant. We have also mentioned previously that the number of alternatives grows as $O(n^2)$ for every pass through the schedule using EVOLVE-1 and EVOLVE-2. Table 15 also shows that the scaled number of alternatives is not constant as might be expected, but generally increases slightly with increasing problem size. The reason is that the number of schedule passes required to converge to a local optimum increases with increasing problem size.

Conclusions

The effectiveness of a solution method is demonstrated for the production scheduling of general single-stage process plants, in which simultaneous lot sizing must be performed. The number of production runs of each product, the length of each run, the assignment of production runs to process units, and the sequence of runs on each unit are determined. The method consists of the generation of an initial solution using a best-fit approach followed by an evolutionary improvement strategy. The representation of the optimal scheduling problem in terms of stockout minimization makes possible the efficient solution of the optimization problem. Globally optimal or near-optimal solutions are obtained for industrial-scale test problems that are intractable using exact methods.

Notation

b_k = process unit capacity
c_{ij} = matrix of cleanout times
d_n = order due date
e_k = unit availability time
I_i = initial inventory
M_i = set of allowable units
M = number of process units
N = number of customer orders
o_k = initial product on unit k
p_n = product type for order n
P = number of products
r_i = release time for order i
q_n = order quantity
t_{ik} = process time for i in unit k
T = scheduling horizon
y_{ik} = yield factor

Literature Cited

Musier, R. F. H., and L. B. Evans, "An Approximate Method for the Production Scheduling of Industrial Batch Processes with Parallel Units," *Comput. Chem. Eng.*, **13**, 229 (1989).

Musier, R. F. H., and L. B. Evans, "Batch Process Management," *Chem. Eng. Prog.*, **86**, 66 (1990).

Graham, R. L., E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan, "Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey," *Ann. of Disc. Math.*, **5**, 287 (1978).

Graves, S., "A Review of Production Scheduling," *Ops. Res.*, **29**, 646 (1981).

Manuscript received July 16, 1990, and revision received Apr. 11, 1991.