Continuous-Time Optimization Approach for Medium-Range Production Scheduling of a Multiproduct Batch Plant

Xiaoxia Lin and Christodoulos A. Floudas*

Department of Chemical Engineering, Princeton University, Princeton, New Jersey 08544-5263

Sweta Modi and Nikola M. Juhasz

ATOFINA Chemicals, Inc., 900 First Avenue, King of Prussia, Pennsylvania 19406-0936

The medium-range production scheduling problem of a multiproduct batch plant is studied. The methodology consists of a decomposition of the whole scheduling period to successive short horizons. A mathematical model is proposed to determine each short horizon and the products to be included. Then a novel continuous-time formulation for short-term scheduling of batch processes with multiple intermediate due dates is applied to each time horizon selected, leading to a large-scale mixed-integer linear programming (MILP) problem. Special structures of the problem are further exploited to improve the computational performance. An integrated graphical user interface implementing the proposed optimization framework is presented. The effectiveness of the proposed approach is illustrated with a large-scale industrial case study that features the production of 35 different products according to a basic three-stage recipe and its variations by sharing 10 pieces of equipment.

1. Introduction

In multiproduct batch plants, different products are manufactured via the same or a similar sequence of operations by sharing available pieces of equipment, intermediate materials, and other production resources. They have long been accepted for the manufacture of chemicals that are produced in small quantities and for which the production process or the demand pattern is likely to change. The inherent operational flexibility of this type of plant provides the platform for great savings reflected in a good production schedule.

The research area of production scheduling and planning of multiproduct and multipurpose chemical processes has received great attention in the past decade. One of the most recent reviews of the related works is that of Shah,[1] which first examined different techniques for optimizing production schedules at individual sites, with an emphasis on formal mathematical methods, and then focused on progress in the overall planning of production and distribution in multisite flexible manufacturing systems. In another review, Pekny and Reklaitis[2] discussed the nature and characteristics of the scheduling/planning problems in chemical processing industries and pointed out the key implications for the solution methodology for these problems. Most of the work in this area has dealt with either the long-term planning problem or the short-term scheduling problem. Long-term planning or capacity expansion problems involve identifying the timing and location of additional facilities over a relatively long time horizon.[3] Short-term scheduling models address detailed sequencing of various operational tasks over short time periods. All of the mathematical models in the literature can be classified into two main groups based on the time representations. Early attempts rely on the discretization of the time horizon into a number of intervals of equal duration.[4,5] This approach is a discrete approximation of the time horizon and results in an unnecessary increase of the overall size of the mathematical model. Recent work aims at developing efficient continuous-time models.[6−12] However, it should be pointed out that all slot-based formulations[6−8] restrict the time representation and result by definition in suboptimal solutions. Floudas and co-workers[13−15] proposed a novel true continuous-time mathematical model for the general short-term scheduling problem of batch, continuous, and semicontinuous processes, which is the basis of the work presented in this paper. Lin and Floudas[16] further extended this model to incorporate scheduling issues in the design and synthesis of multipurpose batch processes.

The rest of this paper is organized as follows. We will first present the problem investigated in this work. Then the overall framework is proposed, and detailed formulations of a decomposition model and a short-term scheduling model are discussed. Computational results from an industrial case study are also given. At the end, an integrated graphical user interface implementing the proposed optimization framework is presented.

2. Problem Description

In this work, we investigate the medium-range production scheduling problem of a multiproduct batch plant, which is defined as follows: Given (i) the production recipe (i.e., the processing times for each task at the suitable units and the amount of the materials required for the production of each product), (ii) the available units and their capacity limits, (iii) the available storage capacity for each of the materials, and (iv) the medium-range time horizon under consideration, then the objective is to determine (i) the optimal sequence of tasks taking place in each unit, (ii) the amount of material being processed at each time in each unit, and (iii) the processing time of each task in each unit, so as to satisfy the market requirements expressed as specific amounts of products at given time instances within the time horizon.

* Author to whom all correspondence should be addressed. Tel: (609) 258-4595. Fax: (609) 258-0211. E-mail: floudas@titan.princeton.edu.

10.1021/ie011002a CCC: $22.00
Published on Web 06/22/2002

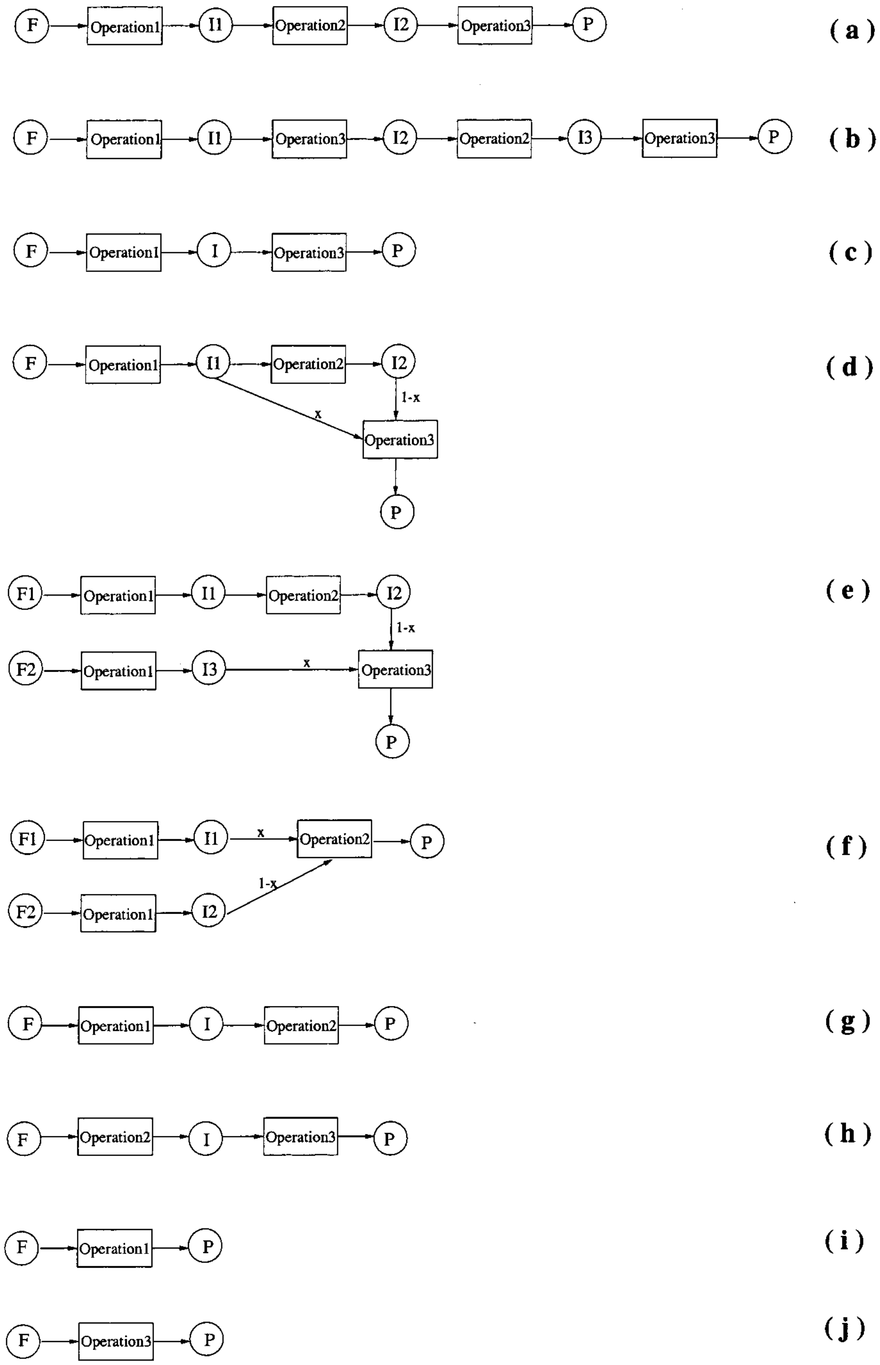

Figure 1. STN of production recipes.

In the batch plant investigated, there are three types of operations: operations 1−3. Up to 60 different products can be produced. For each of them, one of the processing recipes shown in Figure 1 is applied. The recipes are represented in the form of state-task network (STN),[4] in which the *state* node is denoted by a circle and the *task* node by a rectangular box. Some products share the same operation 1 step.

The plant has three types of units: four type 1 units (units 1−4) for operation 1, three type 2 units (units 5−7) for operation 2, and three type 3 units (units 8−10) for operation 3. The information on which units are suitable for each product is given. Type 1 units and type 3 units are utilized in a batch mode, while type 2 units operate in a continuous mode. The capacity limit of each type 1 unit varies from one product to another, while the capacity limit of each type 3 unit is the same for all suitable products. The processing time or processing rate of each task in the suitable units is also specified. When switched from one type of product to another, the units need cleaning up between. The data above are relatively fixed and seldom change over a long period.

The time horizon considered for production scheduling is as long as a whole month. Customer orders are distributed throughout the time horizon with specified amounts, due dates, and priorities. We assume (i) no limitation on raw materials and (ii) unlimited storage capacity for all materials based on an analysis of the specific situation in the plant.

3. Overall Framework

The overall methodology for solving the medium-range production scheduling problem is to decompose

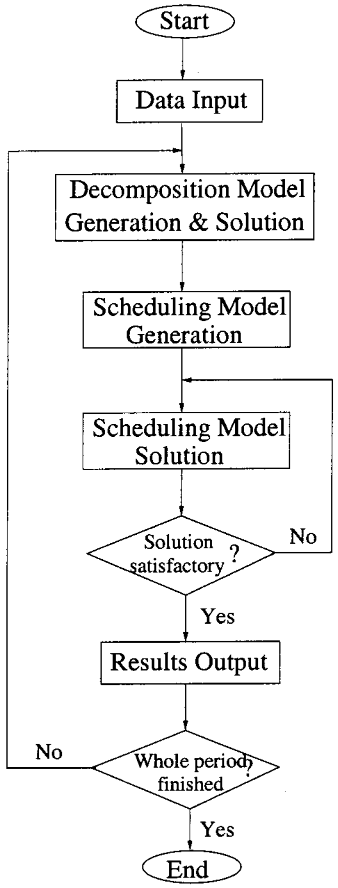

Figure 2. Flowchart of the rolling horizon approach.

the large and complex problem to smaller short-term scheduling subproblems in successive time horizons. The flowchart is shown in Figure 2. The first step is to input relevant data. Then, a mathematical model for the determination of the current time horizon and corresponding products that should be included is formulated and solved. According to the solution of the decomposition model, a short-term scheduling model is formulated using the information on customer orders, inventory levels, and processing recipes. The resulting mixed-integer linear programming (MILP) problem is a large-scale complex problem which requires a large computational effort for its solution and exhibits difficulties in obtaining global optimality. It is solved iteratively by using cutoff values until a satisfactory feasible solution is obtained. Then the solution is output, and the next time horizon is to be solved. The above procedure is applied iteratively until the whole scheduling period under consideration is finished.

4. Decomposition Model

A key issue that arises in the rolling horizon approach described above is the determination of the time horizon and those products that should be considered for each short-term scheduling subproblem. We develop a two-level mathematical formulation that effectively addresses this issue, taking into account the tradeoff between demands satisfaction, unit utilization, and model complexity. In the first level, the time horizon is determined and the main products that should be considered for all processing steps are identified, while in the second level, additional products are identified to go through the operation 1 step if needed.

4.1. Level 1 Formulation. The mathematical model involves the following notations:

Nomenclature

Sets

D = days
P = products
Rg = operation 1 groups
P_r = products in operation 1 group r which share the same operation 1 step
Ud = type 3 units
Ud_p = type 3 units suitable for product p

Parameters

nepd = number of event points per day
$\mathrm{dem}_{p,d}$ = amount of demand for product p due on day d
demt_p = total amount of demand for product p under consideration
suit_r = total number of type 1 units suitable for production of operation 1 group r
suit_p = total number of other units suitable for production of product p
capd_u = capacity of type 3 unit u
$\mathrm{ftd}_{p,u}$ = fixed processing time in type 3 unit u for production of product p
$\mathrm{prior}_{p,d}$ = customer priority of demand for product p due on day d
$\mathrm{ahea}_{p,d}$ = minimum number of days in advance to start processing to satisfy the demand for product p due on day d, which can be derived from the amount of the demand, unit capacities, and task processing times
wt_p = overall weight of product p based on the priority, amount, and due date of its first demand
cmp_r = complexity index for processing of operation 1 group r, >1 for some products with special restrictions and =1 for others
cplm = maximum number of $\mathrm{wv}(i,j,n)$ variables allowed as the complexity limit of the resulting scheduling model
γ_d = upper bound on the ratio of the time for which the type 3 units can be used for selected products over the time horizon
α = coefficient in objective function accounting for relative importance of products inclusion compared to horizon maximization

Variables

$\mathrm{day}(d)$ = binary, whether to include day d
$\mathrm{prod}(p)$ = binary, whether to include product p
$\mathrm{rg}(r)$ = binary, whether to include operation 1 group r

Based on the above definitions of parameters and variables, the following constraints and objective function are formulated:

Horizon continuity

$$\mathrm{day}(d) \geq \mathrm{day}(d+1) \qquad \forall d \in D,\ d \neq d_{\mathrm{last}} \tag{1}$$

These constraints ensure that successive days starting from the first one are selected to form a continuous time horizon.

Inclusion of products with demands in current horizon

$$\mathrm{prod}(p) \geq \mathrm{day}(d) \qquad \forall p \in P,\ d \in D,\ \mathrm{dem}_{p,d} > 0 \tag{2}$$

These constraints state that if there is a demand for product p due on day d and day d is included in the current time horizon, then product p should be considered for this horizon.

Inclusion of products with demands in the future

$$\mathrm{prod}(p) \geq \mathrm{day}(d - \mathrm{ahea}_{p,d}) \qquad \forall p \in P,\ d \in D,\ \mathrm{ahea}_{p,d} > 0 \quad (3)$$

These constraints express the requirement that if a demand for product p due on day d needs $\mathrm{ahea}_{p,d}$ day(s) in advance to start processing in order to satisfy the due date and day $(d - \mathrm{ahea}_{p,d})$ is included in the horizon, then product p has to be taken into account for this horizon.

$$\mathrm{prod}(p) \geq \mathrm{day}(d - 1) \qquad \forall p \in P,\ d \in D,\ \mathrm{prior}_{p,d} \text{ is high} \quad (4)$$

These constraints state that if a demand is of high priority, then it is pushed 1 day forward. These constraints are not necessary; however, this explicit consideration of demand priorities can lead to solutions that satisfy demands with high priorities to a larger extent.

Definition for variables rg(r)

$$\mathrm{rg}(r) \leq \sum_{p \in P_r} \mathrm{prod}(p) \qquad \forall r \in \mathrm{Rg} \quad (5)$$

$$\mathrm{rg}(r) \geq \mathrm{prod}(p) \qquad \forall r \in \mathrm{Rg},\ p \in P_r \quad (6)$$

These constraints relate rg(r) variables with corresponding prod(p) variables. An operation 1 group r is included, that is, rg(r) = 1, if and only if one or more of the products that belong to this operation 1 group are included, that is, prod(p) = 1.

Model complexity limit

$$[\sum_{r \in \mathrm{Rg}} \mathrm{rg}(r)\ \mathrm{suit}_r\ \mathrm{cmp}_r + \sum_{p \in P} \mathrm{prod}(p)\ \mathrm{suit}_p] \cdot \sum_{d \in D} \mathrm{day}(d)\ \mathrm{nepd} \leq \mathrm{cplm} \quad (7)$$

The left-hand side of this constraint gives an estimate of the number of wv(i,j,n) binary variables (see the scheduling model in the next section) in the resulting scheduling problem. Because it can be used to represent the scale and complexity of the scheduling problem, this constraint keeps the scheduling problem under tractable size by imposing an upper bound on the total number of (i, j, n) combinations.

Type 3 unit utilization limit

$$\sum_{p \in P} \mathrm{prod}(p) \frac{\mathrm{demt}_p}{\sum_{u \in \mathrm{Ud}_p} \mathrm{capd}_u} \min_{u \in \mathrm{Ud}_p} \{\mathrm{ftd}_{p,u}\} |\mathrm{Ud}_p| \leq \gamma_d \sum_{d \in D} \mathrm{day}(d) \times 24 |\mathrm{Ud}| \quad (8)$$

The left-hand side represents a lower bound on the total number of hours for which the type 3 units are utilized to satisfy demands of selected products under consideration, for example, within 2 weeks. Thus, this constraint limits the selected products by considering the utilization of type 3 units.

Objective: Maximization of duration of horizon and products included

$$\sum_{d \in D} \mathrm{day}(d) + \alpha \sum_{p \in P} \mathrm{wt}_p\ \mathrm{prod}(p) \quad (9)$$

The first term in the objective maximizes the duration of the time horizon, while the second term aims at including as many products as possible. α is used to balance the relative importance of these two targets.

The formulation described above is a mixed-integer nonlinear programming (MINLP) problem due to the bilinear terms of binary variables in constraint (7). We introduce additional binary variables pd(p,d) and rd(r,d) and the following constraints to replace the bilinear products of prod(p) day(d) and rg(r) day(d).

Definition for variables pd(p,d)

$$\mathrm{pd}(p,d) \leq \mathrm{prod}(p) \qquad \forall p \in P,\ d \in D \quad (10)$$

$$\mathrm{pd}(p,d) \leq \mathrm{day}(d) \qquad \forall p \in P,\ d \in D \quad (11)$$

$$\mathrm{pd}(p,d) \geq \mathrm{prod}(p) + \mathrm{day}(d) - 1 \qquad \forall p \in P,\ d \in D \quad (12)$$

This set of linear constraints is equivalent to pd(p,d) = prod(p) day(d) because pd(p,d) are binary variables.

Definition for variables rd(r,d)

$$\mathrm{rd}(r,d) \leq \mathrm{rg}(r) \qquad \forall r \in \mathrm{Rg},\ d \in D \quad (13)$$

$$\mathrm{rd}(r,d) \leq \mathrm{day}(d) \qquad \forall r \in \mathrm{Rg},\ d \in D \quad (14)$$

$$\mathrm{rd}(r,d) \geq \mathrm{rg}(r) + \mathrm{day}(d) - 1 \qquad \forall r \in \mathrm{Rg},\ d \in D \quad (15)$$

Similarly, these constraints are equivalent to rd(r,d) = rg(r) day(d).

Now, constraint (7) can be reformulated as follows:

Reformulation of constraint on model complexity limit

$$\{\sum_{r \in \mathrm{Rg}} [\mathrm{suit}_r\ \mathrm{cmp}_r \sum_{d \in D} \mathrm{rd}(r,d)] + \sum_{p \in P} [\mathrm{suit}_p \sum_{d \in D} \mathrm{pd}(p,d)]\} \mathrm{nepd} \leq \mathrm{cplm} \quad (16)$$

When constraints (10)−(15) are included and constraint (7) is replaced with constraint (16), the original MINLP problem is transformed to an MILP problem and can be solved to global optimality effectively.

4.2. Level 2 Formulation. After the time horizon and the main products are determined in the first level, a second level mathematical model is formulated to investigate the utilization of the type 1 units, in which the first step in the processing sequence, the operation 1 step, is performed, and to include additional products, if necessary, to go through the operation 1 step to ensure that the type 1 units are utilized efficiently.

The second-level mathematical model involves the following notations:

Nomenclature

Sets

Rg = operation 1 groups
P_r = products in operation 1 group r
Ur = type 1 units
Ur_r = type 1 units suitable for operation 1 group r

Parameters

sl_p = 0−1 parameter to indicate whether product p is selected in the first level

demt_r = total amount of demand for post-operation 1 intermediate material of operation 1 group r

$\mathrm{capr}_{r,u}$ = capacity of type 1 unit u for processing of operation 1 group r

$\mathrm{ftr}_{r,u}$ = fixed processing time in type 1 unit u for processing of operation 1 group r

γ_r = lower bound on the fraction of the time horizon for which the type 1 units are utilized

H = duration of the time horizon determined in the first level

Variables

$\mathrm{rg}(r)$ = binary, whether to include operation 1 group r

The following constraints and objective function are formulated:

Inclusion of products selected in first level

$$\mathrm{rg}(r) \geq \mathrm{sl}_p \qquad \forall r \in \mathrm{Rg},\ p \in P_r \tag{17}$$

These constraints ensure that if a product is selected in the first level, that is, $\mathrm{sl}_p = 1$, then the operation 1 group to which this product belongs is included, that is, $\mathrm{rg}(r) = 1$.

Type 1 unit utilization

$$\sum_{r\in\mathrm{Rg}} \mathrm{rg}(r)\frac{\mathrm{demt}_r}{\sum_{u\in\mathrm{Ur}_r}\mathrm{capr}_{r,u}} \min_{u\in\mathrm{Ur}_r}\{\mathrm{ftr}_{r,u}\}|\mathrm{Ur}_r| \geq \gamma_r H|\mathrm{Ur}| \tag{18}$$

The left-hand side represents a lower bound on the total number of hours for which the type 1 units are utilized to satisfy demands of selected operation 1 groups. Thus, when a lower bound such as the one on the above right-hand side is imposed, this constraint expresses the requirement that enough products should be included to utilize the type 1 units efficiently.

Objective:
Minimization of operation 1 groups included

$$\sum_{r\in\mathrm{Rg}} \mathrm{rg}(r) \tag{19}$$

The objective in the second level is to minimize the total number of products included to limit the size and complexity of the resulting scheduling problem as long as efficient utilization of the type 1 units is ensured.

The second-level mathematical model leads to an MILP problem and can be solved easily.

5. Short-Term Scheduling Formulation

5.1. Basic Formulation. After each time subhorizon and corresponding products to be included are determined with the decomposition model, a continuous-time formulation for short-term scheduling with multiple intermediate due dates is applied. This formulation is based on Floudas et al.'s works,[13−15] featuring the novel concept of event points and formulation of special sequence constraints.

The formulation is presented in detail as follows:

Nomenclature

Sets

I = tasks

I_j = tasks which can be performed in unit j

I_s = tasks which either produce or consume state s

J = units

J_i = units which are suitable for performing task i

N = event points within the time horizon

S = material states

Parameters

$V_{ij}^{\min}$ = minimal capacity allowed of the specific unit j when performing task i

$V_{ij}^{\max}$ = maximal capacity allowed of the specific unit j when performing task i

tav_j = time when unit j starts being available

dend_s = market requirement for state s at the end of the time horizon

ρ_{si}^{p}, ρ_{si}^{c} = proportion of state s produced, consumed from task i, respectively

α_{ij} = constant term of processing time of task i in unit j

β_{ij} = variable term of processing time of task i in unit j expressing the time required by the unit to process one unit of material performing task i

H = time horizon

valm_s = relative value of state s in the sequence of materials for the corresponding product

valp_s = relative value of the corresponding product indicating its priority

vald_s = relative value of the corresponding product indicating its importance to fulfill future demands

$\mathrm{tcl}_{ii'}$ = cleanup times of units when switched from task i' to task i

dint_{sn} = demand for state s at event point n, which is specified based on the relative time at which the demand has to be fulfilled, the number of stages required to produce the final product, and the number of other tasks that may take place in the same unit

due_{sn} = due time for demand for state s at event point n

pri_{sn} = priority of demand for state s at event point n

γ = constant coefficient in objective function balancing meeting demands with intermediate due dates and overall production

Variables

$\mathrm{wv}(i,j,n)$ = binary variables that assign the beginning of task i in unit j at event point n

$\mathrm{yv}(j,n)$ = binary variables that assign the utilization of unit j at event point n

$B(i,j,n)$ = amount of material undertaking task i in unit j at event point n

$\mathrm{STI}(s)$ = initial amount of state s

$\mathrm{ST}(s,n)$ = amount of state s at event point n

$\mathrm{STF}(s)$ = amount of state s at the end of the horizon

$D(s,n)$ = amount of state s delivered at event point n

$\mathrm{SL}(s,n)$ = slack variable for the amount of state s not meeting the demand at event point n

$T^{\mathrm{s}}(i,j,n)$ = time that task i starts in unit j at event point n

$T^{\mathrm{f}}(i,j,n)$ = time that task i finishes in unit j while it starts at event point n

Allocation constraints

$$\sum_{i\in I_j}\mathrm{wv}(i,j,n) = \mathrm{yv}(j,n) \qquad \forall j \in J,\ n \in N \tag{20}$$

These constraints express that in each unit j and at an event point n only one of the tasks that can be performed in this unit (i.e., $i \in I_j$) should take place. If unit j is utilized at event point n, that is, yv(j,n) equals 1, then one of the wv(i,j,n) variables should be activated. If unit j is not utilized at event point n, then all wv(i,j,n) variables take zero values, that is, no assignments of tasks are made.

Material balances

$$\mathrm{ST}(s,n_{1st}) = \mathrm{STI}(s) + \sum_{i\in I_s}\rho^{c}_{si}\sum_{j\in J_i}B(i,j,n_{1st}) \qquad \forall s \in S \tag{21}$$

$$\mathrm{ST}(s,n) = \mathrm{ST}(s,n-1) - D(s,n) + \sum_{i\in I_s}\rho^{p}_{si}\sum_{j\in J_i}B(i,j,n-1) + \sum_{i\in I_s}\rho^{c}_{si}\sum_{j\in J_i}B(i,j,n) \qquad \forall s \in S,\ n \in N \tag{22}$$

$$\mathrm{STF}(s) = \mathrm{ST}(s,n_{last}) + \sum_{i\in I_s}\rho^{p}_{si}\sum_{j\in J_i}B(i,j,n_{last}) \qquad \forall s \in S \tag{23}$$

where $\rho^{c}_{si} \le 0$ and $\rho^{p}_{si} \ge 0$ represent the proportion of state s consumed by or produced from task i, respectively. According to these constraints, the amount of material of state s at event point n is equal to that at event point $n - 1$ adjusted by any amounts delivered at event point n and produced or consumed between the event points $n - 1$ and n.

Capacity constraints

$$B(i,j,n) \ge V^{\min}_{ij}\mathrm{wv}(i,j,n) \qquad \forall i \in I,\ j \in J_i,\ n \in N \tag{24}$$

$$B(i,j,n) \le V^{\max}_{ij}\mathrm{wv}(i,j,n) \qquad \forall i \in I,\ j \in J_i,\ n \in N \tag{25}$$

These constraints express the minimal and maximal allowed capacity of a unit j, respectively, when performing task i. If wv(i,j,n) equals 1, then constraints (24) and (25) correspond to lower and upper bounds on the batch size, $B(i,j,n)$. If wv(i,j,n) equals zero, then $B(i,j,n)$ becomes zero.

In the multiproduct plant that we study, there is no physical restriction on the minimal capacity of units and parameters $V^{\min}_{i,j}$ are set to zero. However, if it is not allowed or not suitable to operate with small batch sizes because of other considerations not included in this model, appropriate artificial values can be incorporated. There are also cases in which some units, for example, the type 1 units, are always operated in full capacities to produce as much as possible. Then, the two inequality constraints (24) and (25) are combined and give the following equality constraint:

$$B(i,j,n) = V^{\max}_{ij}\mathrm{wv}(i,j,n) \qquad \forall i \in I_r,\ j \in J_i,\ n \in N \tag{26}$$

where I_r is the set of operation 1 tasks.

Duration constraints

$$T^{\mathrm{f}}(i,j,n) = T^{\mathrm{s}}(i,j,n) + \alpha_{ij}\mathrm{wv}(i,j,n) + \beta_{ij}B(i,j,n) \qquad \forall i \in I,\ j \in J_i,\ n \in N \tag{27}$$

where α_{ij} are the fixed processing times for batch tasks (operations 1 and 3) and zero for continuous tasks (operation 2) and β_{ij} are the inverse of processing rates for continuous tasks and zero for batch tasks, respectively. The duration constraints express the dependence of the time duration of task i in unit j at event point n on the amount of material being processed. If wv(i,j,n) equals 1, then the last two terms in constraints (27) are added to $T^{\mathrm{s}}(i,j,n)$. If wv(i,j,n) equals zero, then the last two terms become zero because of the capacity constraints (24) and (25) and hence $T^{\mathrm{f}}(i,j,n) = T^{\mathrm{s}}(i,j,n)$.

Sequence constraints

Same task in the same unit

$$T^{\mathrm{s}}(i,j,n+1) \ge T^{\mathrm{f}}(i,j,n) \qquad \forall i \in I,\ j \in J_i,\ n \in N,\ n \ne n_{last} \tag{28}$$

The sequence constraints (28) state that task i starting in unit j at event point $n + 1$ should start after the end of the same task performed in the same unit which has already started at event point n.

Different tasks in the same unit

The following set of constraints (29) establishes the relationship between the starting time of a task i at event point $n + 1$ and the end time of task i' at event point n when these tasks take place in the same unit j.

$$T^{\mathrm{s}}(i,j,n+1) \ge T^{\mathrm{f}}(i',j,n) + \mathrm{tcl}_{ii'}\mathrm{wv}(i,j,n) - H(1 - \mathrm{wv}(i',j,n)) \qquad \forall j \in J,\ i \in I_j,\ i' \in I_j,\ i \ne i',\ n \in N,\ n \ne n_{last} \tag{29}$$

Constraints (29) are written for tasks i and i' that are performed in the same unit j. If both tasks are performed in the same unit, they should be at most consecutive. This is expressed by constraints (29) because if wv(i',j,n) = 1, which means that task i' takes place in unit j at event point n, then the last term of constraints (29) becomes zero, forcing the starting time of task i in unit j at event point $n + 1$ to be greater than the ending time of task i' in unit j at event point n plus the required cleanup time; otherwise, the right-hand side of constraints (29) becomes negative, and the constraint is trivially satisfied.

Different tasks in different units

$$T^{\mathrm{s}}(i,j,n+1) \ge T^{\mathrm{f}}(i',j',n) - H(1 - \mathrm{wv}(i',j',n)) \qquad \forall j,\ j' \in J,\ i \in I_j,\ i' \in I_{j'},\ i \ne i',\ n \in N,\ n \ne n_{last} \tag{30}$$

Constraints (30) are written for different tasks i and i' that are performed in different units j and j' but take place consecutively according to the production recipe. Note that if task i' takes place in unit j' at event point n (i.e., wv(i',j',n) = 1), then we have $T^{\mathrm{s}}(i,j,n+1) \ge T^{\mathrm{f}}(i',j',n)$ and hence task i in unit j has to start after the end of task i' in unit j'. Otherwise, the right-hand side becomes negative and the constraint is trivially satisfied.

Constraints for demands with intermediate due dates

$$D(s,n) + \mathrm{SL}(s,n) = \mathrm{dint}_{sn} \qquad \forall s \in S,\ n \in N \tag{31}$$

These constraints represent that products are delivered at event points when demands exist. The slack variables, SL(s,n), are introduced to give more flexibility to the model in handling partial fulfillment of demands. Under feasible conditions, some or all of these variables can be fixed to zero to ensure that some or all of the demands within the time horizon are met.

Due dates constraints

$$T^{s}(i,j,n) \le \mathrm{due}_{sn} \qquad \forall s \in S,\, i \in I_s,\, j \in J_i \quad (32)$$

These constraints ensure the satisfaction of product demand by the corresponding due date.

Constraints for demands at the end of the time horizon

$$\mathrm{STF}(s) \ge \mathrm{dend}_s \qquad \forall s \in S,\, n \in N \quad (33)$$

These constraints ensure the satisfaction of demands which should be met at the end of the time horizon.

Unit available time constraints

$$T^{s}(i,j,n) \ge \mathrm{tav}_j - H(1 - \mathrm{wv}(i,j,n)) \qquad \forall i \in I,\, j \in J_i,\, n \in N \quad (34)$$

These constraints represent the requirement of not starting any task until the unit is available. When wv(i,j,n) equals zero, which means the task is not activated, the constraint is relaxed and becomes trivial.

Time horizon constraints

$$T^{f}(i,j,n) \le H \qquad \forall i \in I,\, j \in J_i,\, n \in N \quad (35)$$

$$T^{s}(i,j,n) \le H \qquad \forall i \in I,\, j \in J_i,\, n \in N \quad (36)$$

The time horizon constraints represent the requirement that every task start and end within the time horizon H.

Objective: Maximization of production

$$-\sum_s \sum_n \mathrm{pri}_{sn}\, \mathrm{SL}(s,n) + \gamma \sum_s \mathrm{vald}_s\, \mathrm{valp}_s\, \mathrm{valm}_s\, \mathrm{STF}(s) \quad (37)$$

The objective shown in eq 37 is the maximization of production in terms of the relative value of all states minus the penalty term for not meeting demands at intermediate due dates.

5.2. Additional Constraints. The mathematical formulation described above results in an MILP problem, which can be solved by a commercial MILP solver such as CPLEX. Mainly because of its original concept of event points, the proposed formulation outperforms some other existing discrete-time and pseudo-continuous-time formulations in terms of reducing significantly the size of the resulting mathematical programming problem and thus the required computational resources. However, solving the resulting MILP problems is still very challenging (e.g., the solutions require considerable CPU time for proof of global optimality), which reflects the inherent complexity of the specific physical conditions. To improve the modeling and solution, the following constraints are incorporated:

Tighter Sequence Constraints for Operation 1. Because operation 1 is the first step in the task sequence for all products and we want to maximize the overall production in principle, the timing of the operation 1 tasks can be enforced to be as tight as possible.

Same task in the same unit

$$T^{s}(i,j,n+1) \le T^{f}(i,j,n) + H(2 - \mathrm{wv}(i,j,n) - \mathrm{wv}(i,j,n+1)) \qquad \forall i \in I_r,\, j \in J_i,\, n \in N,\, n \ne n_{\mathrm{last}} \quad (38)$$

These additional sequence constraints for the same operation 1 task in the same type 1 unit, combined with constraints (28), enforce the "zero-wait" condition on the task taking place at two consecutive event points. Namely, if wv(i,j,n) = wv(i,j,n+1) = 1 (that is, task i takes place in unit j at both event point n and n + 1), then $T^{s}(i,j,n+1) = T^{f}(i,j,n)$, which states that in unit j task i starting at event point n + 1 starts immediately after the end of the same task which has already started at event point n; otherwise, these constraints are relaxed.

Different tasks in the same unit

$$T^{s}(i,j,n+1) \le T^{f}(i',j,n) + \mathrm{tcl}_{ii'} + H(2 - \mathrm{wv}(i',j,n) - \mathrm{wv}(i,j,n+1)) \qquad \forall j \in J_r,\, i \in I_j,\, i' \in I_j,\, i \ne i',\, n \in N,\, n \ne n_{\mathrm{last}} \quad (39)$$

where J_r is the set of type 1 units.

According to these new constraints and constraints (29), if wv(i',j,n) = wv(i,j,n+1) = 1 (that is, task i' takes place in type 1 unit j at event point n and task i takes place in the same type 1 unit at event point n + 1), then $T^{s}(i,j,n+1) = T^{f}(i',j,n) + \mathrm{tcl}_{ii'}$ requiring that task i in type 1 unit j at event point n + 1 starts immediately after task i' in the same type 1 unit at event point n ends and necessary cleaning is done for the type 1 unit. Otherwise, these constraints are trivial.

Restrictions on Binary Variables. For each product, based on the information of the overall amount of demands and maximal batch sizes of related tasks performed in suitable units, lower bounds on the total number of activated tasks can be specified for operations 1−3, respectively,

$$\sum_{j \in J_i} \sum_{n \in N} \mathrm{wv}(i,j,n) \ge m_i \qquad \forall i \in I \quad (40)$$

where m_i are parameters calculated based on relevant data.

These constraints reduce the combinatorial complexity of the MILP problems and improve the computational performance.

Special Restrictions.

Restriction 1. It is given that operation 1 for a specific product should run in one of the type 1 units in campaign mode; namely, a prespecified minimum number, c, of batches needs to be performed consecutively once started. This arrangement is due to the comparatively long cleanup time for the type 1 unit required to switch from this product to others. The corresponding constraints can be formulated as follows:

$$\mathrm{wv}(i,j,n) \ge \mathrm{wv}(i,j,n-1) \qquad \forall i \in I_{rE},\, j \in J_{rE},\, n \in N,\, 2 \le n \le c \quad (41)$$

$$\mathrm{wv}(i,j,n) \ge \mathrm{wv}(i,j,n-1) - \frac{1}{c-1} \sum_{n-c \le n' \le n-2} \mathrm{wv}(i,j,n') \qquad \forall i \in I_{rE},\, j \in J_{rE},\, n \in N,\, n > c \quad (42)$$

$$\mathrm{wv}(i,j,n) \le \mathrm{wv}(i,j,n-1) \qquad \forall i \in I_{rE},\, j \in J_{rE},\, n \in N,\, n \ge n_{\mathrm{last}} - c + 2 \quad (43)$$

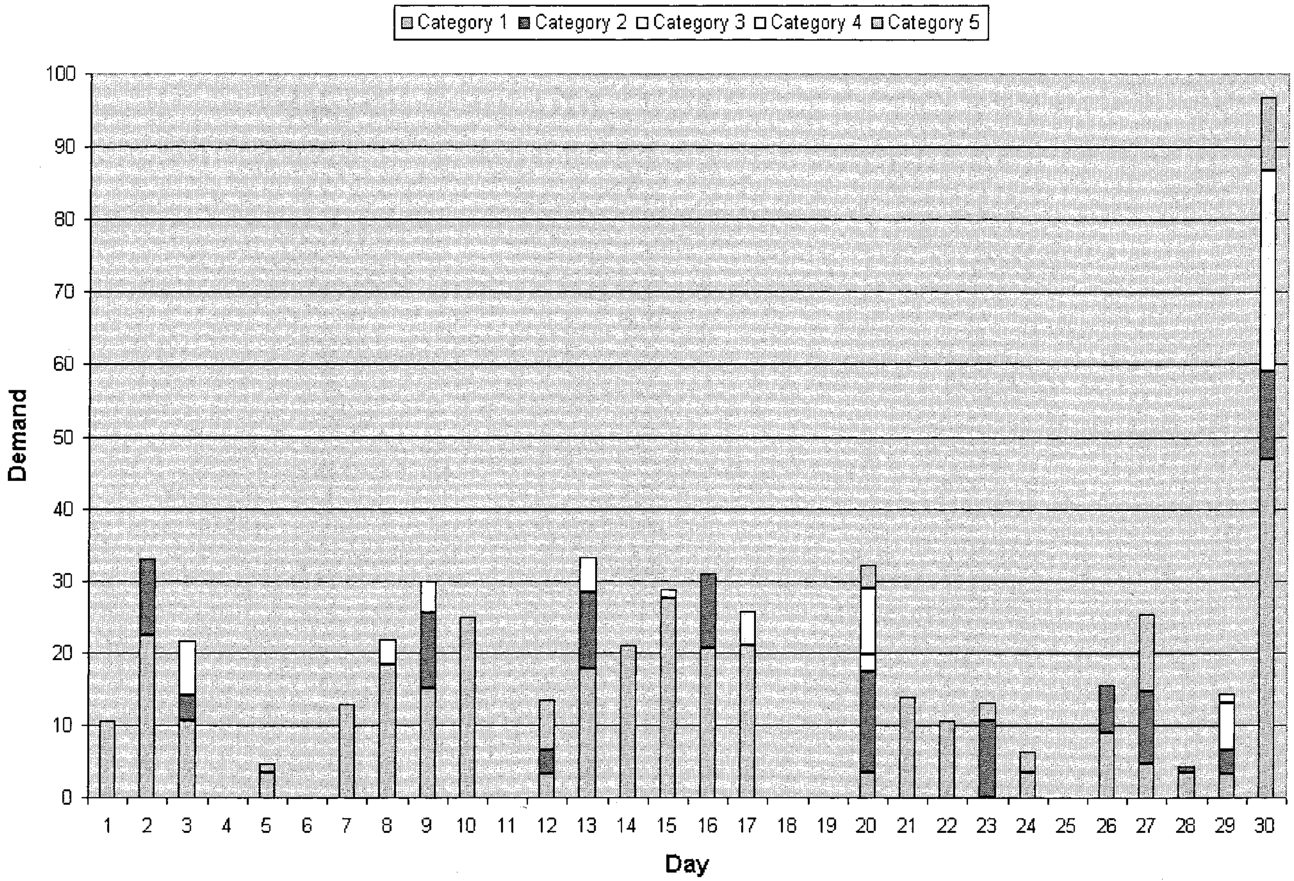

Figure 3. Distribution of demands.

where I_{rE} is the set of operation 1 tasks and J_{rE} is the set of type 1 units dedicated to them.

Constraints (41) and (42) state that if wv(i,j,n−1) equals 1 and wv(i,j,n−2) ∼ wv(i,j,n−c), which do not all exist for n less than c, are not all equal to 1, which means task i has taken place in unit j at event point n − 1 and there have been less than c batches performed, then wv(i,j,n) equals 1, that is, task i should take place in unit j at event point n; otherwise, these constraints become trivial. Constraint (43) ensures that the operation 1 task does not get started after event point $n_{\text{last}} - c + 1$ because the type 1 unit will not be able to perform c batches in the remaining period.

Restriction 2. A subset of the products are described as black. They are also required to be put together for operations 2 and 3 so as to avoid cleanup as much as possible

$$\mathrm{wv}(i,j,n) \le 2 - \sum_{i' \in I_{edB} \cap I_j} [\mathrm{wv}(i',j,n') + \mathrm{wv}(i',j,n'')]$$
$$\forall i \in I_{edNB},\ j \in J_i,\ n,\ n',\ n'' \in N,\ n' < n < n'' \quad (44)$$

where I_{edB} and I_{edNB} are the set of operation 2 and 3 tasks for black products and nonblack products, respectively. If type 2 unit or type 3 unit j is utilized for black products at both event points n' and n'', then the right-hand side of constraint (44) is zero, which states that no task for nonblack products is allowed in the same unit at any event point n between n' and n''. Otherwise, these constraints become trivial.

Restriction 3. Production of a special category of products only needs to go through operations 1 and 3. It is required that these operation 1 and 3 tasks run for 1−2 weeks in a campaign mode. Therefore, demands for this category of products within a period of time (e.g., 1 month) are put together. A type 1 unit and a type 3 unit are then dedicated to the “combined” operation 1 task and operation 3 task, respectively. The relative order of original tasks for different products can be simply based on the due dates of the demands. Production of this category of products is treated separately. The dedicated type 1 and 3 units are excluded from available resources for other products during the dedicated time intervals. This approach will be illustrated through the computational study in the next section.

6. Computational Study

6.1. Problem Overview. The proposed rolling horizon approach is applied to an industrial case study in which detailed production schedules are to be determined to satisfy customer orders for various products distributed within a whole month.

The distribution of the demands throughout the whole month under consideration is plot in Figure 3. There are five main categories of products, and 35 different products are required to be produced in this month. It is assumed that no final product is available at the beginning of the month. However, lower bounds on the amounts of initially available intermediate materials are provided.

The processing recipes to make these products are shown in Figure 1. The operation 1 and 3 steps are performed in a batch mode, while the operation 2 step is performed in a continuous mode. The processing time or processing rate of each step is dependent on both the product and the unit, with operations 1−3 in the ranges of 6−11 h, 0.15−0.25 units/h, and 12−16 h, respectively. Capacities of the four type 1 units vary from 1.125 to 3.5 units/batch, while capacities of the three type 3 units are either 4.5 or 3.5 units/batch. The cleanup time required ranges from 2 to 36 h, depending on the unit and the product sequence involved.

6.2. Detailed Schedules.

"Campaign Mode" Production. We first consider scheduling of the special category of products that require operation in the "campaign mode" (category 4 in Figure 3). Demands for all products in this category in the whole month are grouped together. One type 1 unit and one type 3 unit are dedicated to the production of these products, and the detailed schedules of the operations 1 and 3 tasks for different products are determined based on their relative due dates. The starting time of production of all of these products is determined so that all of the due dates of the demands for these products can be satisfied. Then, the units and time intervals which are used are excluded from available resources for the production of other products. The relative schedule obtained for production of this category of products is shown in Figure 4.

The rolling horizon approach is then applied for the production of the remaining products to break down the large scheduling problem into several short-term scheduling subproblems in successive time horizons. There are mainly two types of connections between consecutive time horizons: initial available time of units and intermediate materials. The decomposition and short-term scheduling models are implemented with MINOPT[17] and solved with CPLEX, a commercial MILP solver. The CPU time required to obtain each solution ranges from 15 min to about 7 h on an HP J-2240 workstation.

Horizon 1. With parameter cplm of 1500, the first horizon is determined to be the first 5 days of the month and eight main products are identified to be included in this horizon according to level 1 of the decomposition model. No additional product is identified to undergo the operation 1 step from level 2 of the decomposition model.

A total of 20 event points are used in the short-term scheduling model for this horizon, which leads to 1320 binary variables, 3968 continuous variables, and 21 912 constraints. Two feasible solutions are generated, and the detailed schedule accepted is shown in Figure 5.

Horizon 2. The second horizon is determined to be the next 5 days of the month, that is, from the 6th day to the 10th day, and six main products are identified to be included according to level 1 of the decomposition model with cplm of 1000. No additional product is identified from level 2 of the decomposition model.

A total of 23 event points are used in the short-term scheduling model for this horizon, which leads to 897 binary variables, 2576 continuous variables, and 14 260 constraints. One feasible solution is obtained and accepted. The detailed schedule is shown in Figure 6. Note that the starting times of the type 1 units correspond to the finishing times of the same units in the previous horizon.

Horizon 3. The third horizon is determined to be from the 11th day to the 14th day of the month, and 10 main products are identified to be included according to level 1 of the decomposition model with cplm of 1500. No additional product is identified from level 2 of the decomposition model.

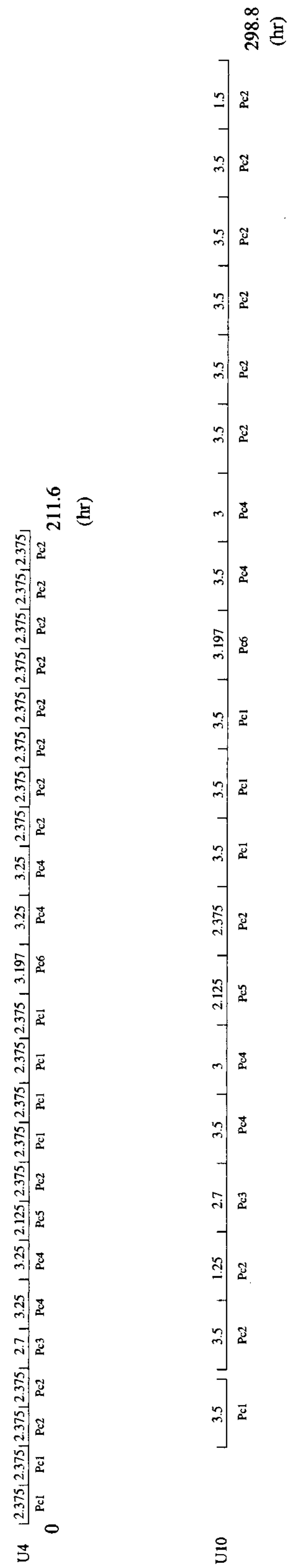

Figure 4. Production schedule for a special category of products in campaign mode (U4 and U10, units; Pc1−Pc6, products).

A total of 19 event points are used in the short-term scheduling model for this horizon, which leads to 1216 binary variables, 3918 continuous variables, and 22 086 constraints. Three feasible solutions are obtained before the last one is accepted. The detailed schedule is shown in Figure 7.

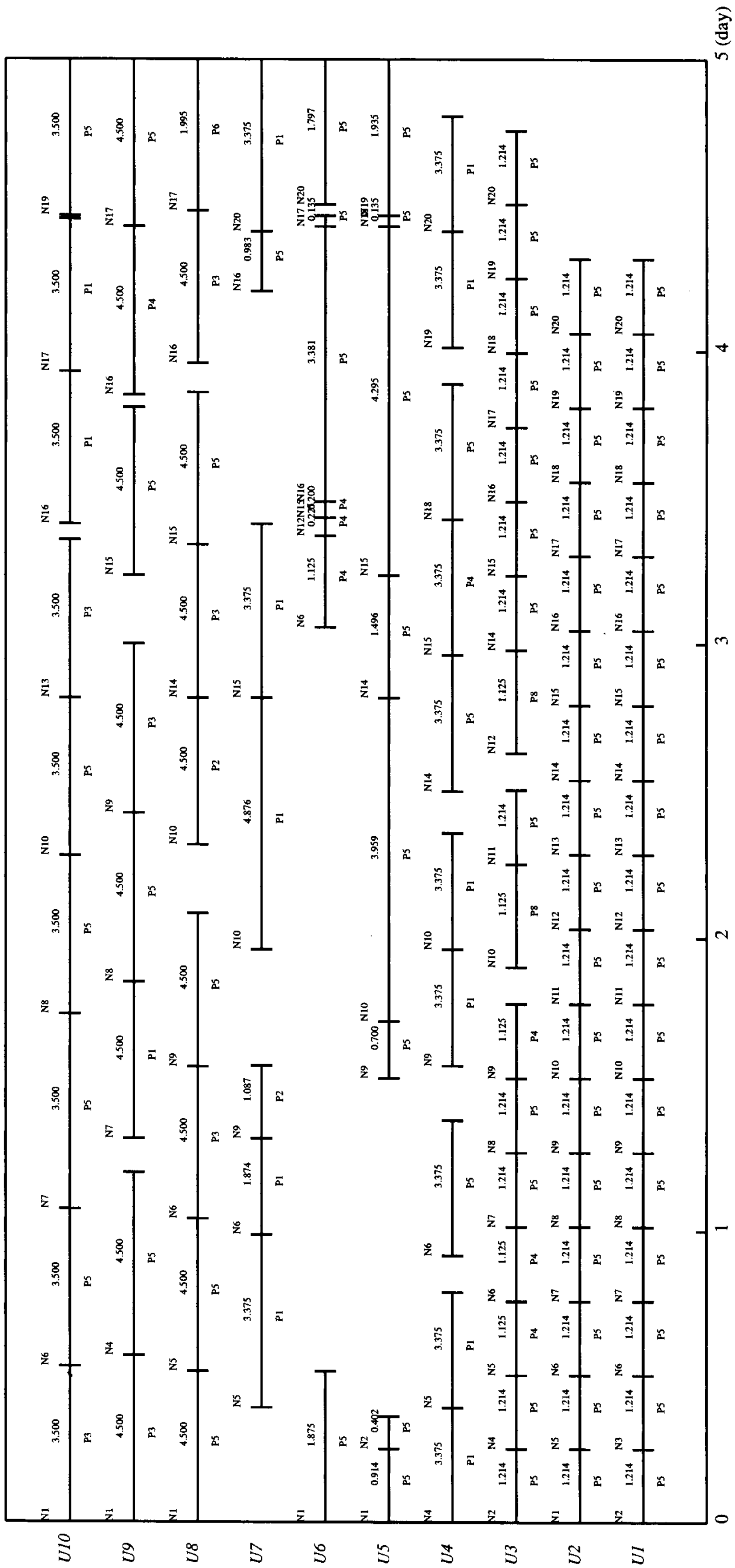

Figure 5. Detailed schedule for time horizon 1 (U1−U10, units; P1−P8, products).

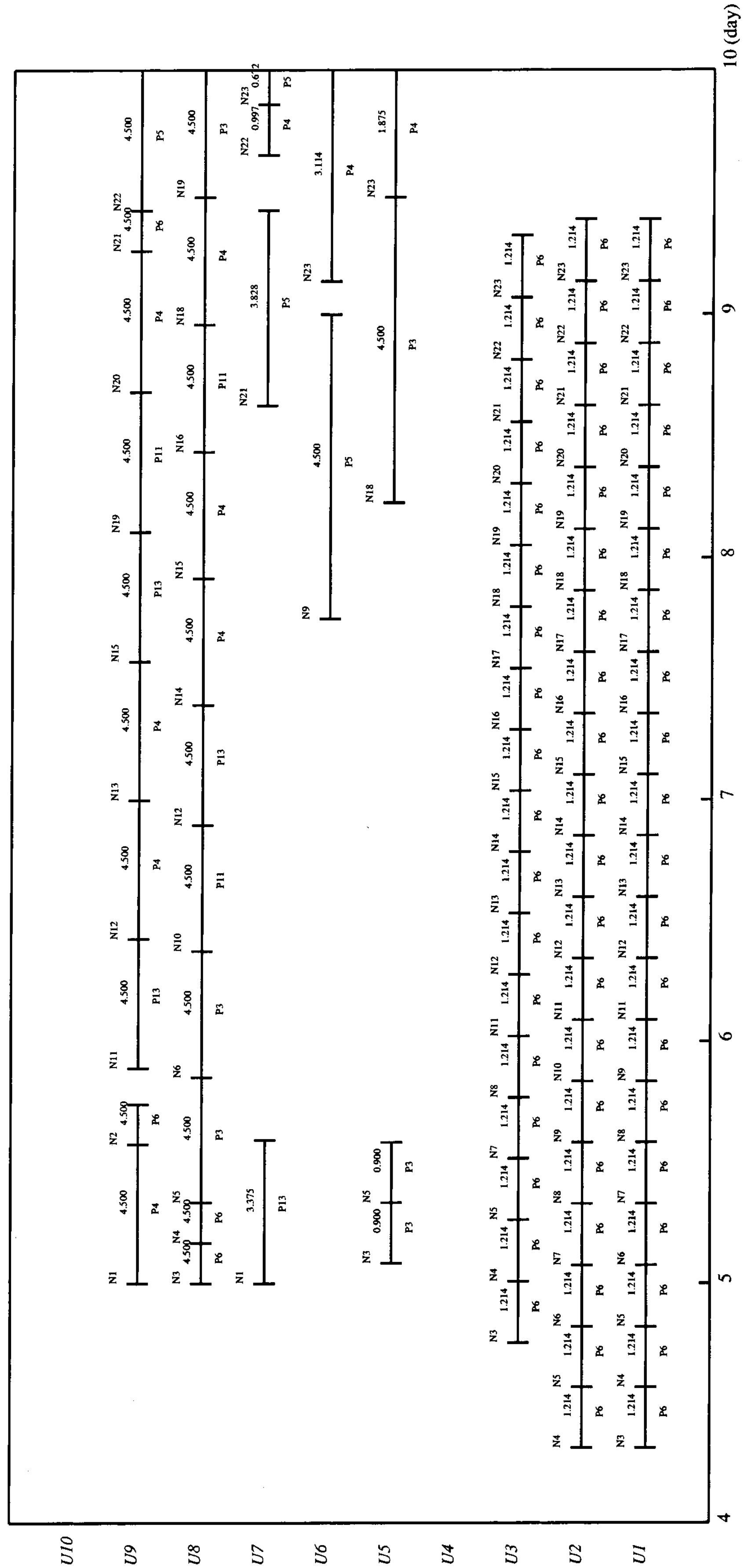

Figure 6. Detailed schedule for time horizon 2 (U1−U10, units; P3−P13, products).

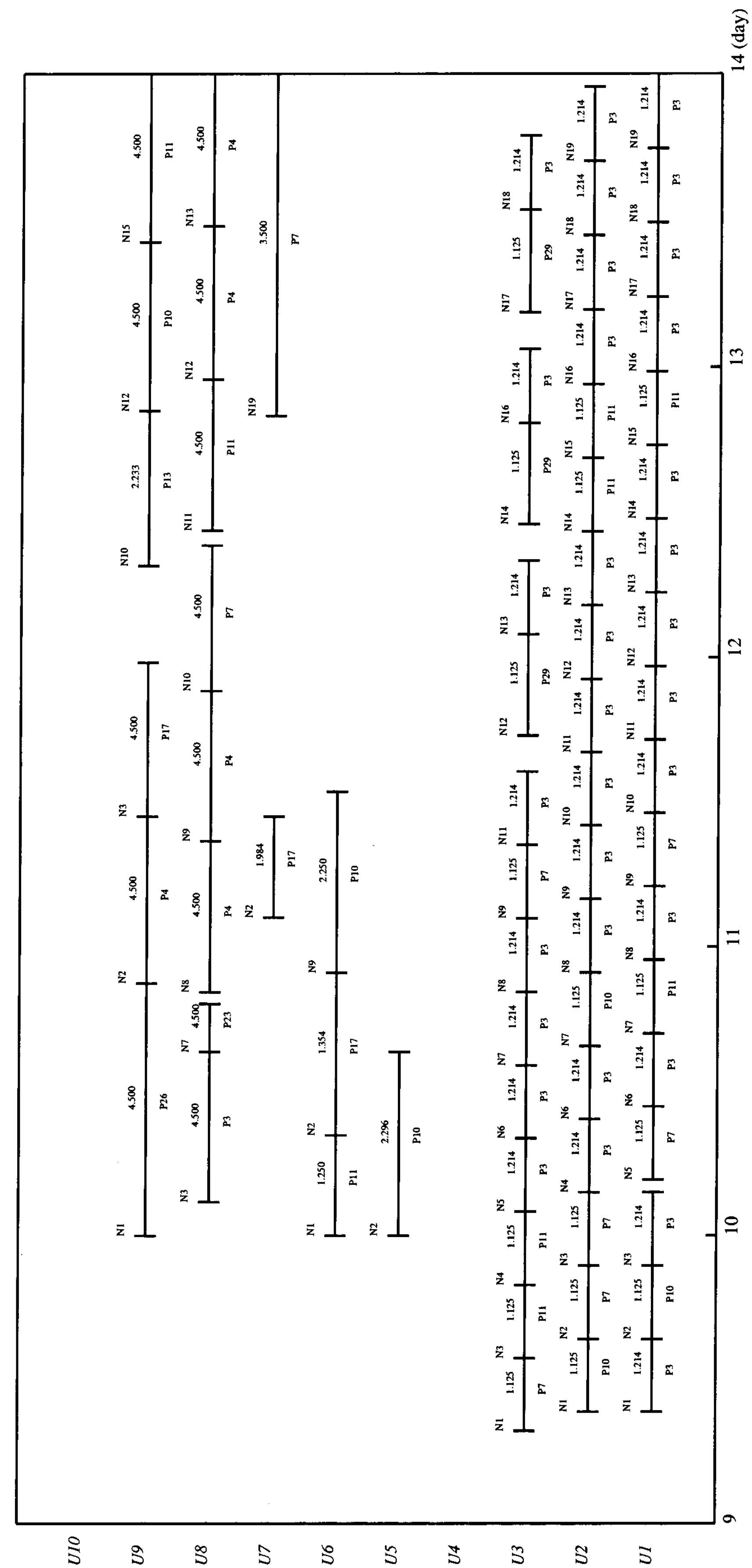

Figure 7. Detailed schedule for time horizon 3 (U1−U10, units; P3−P29, products).

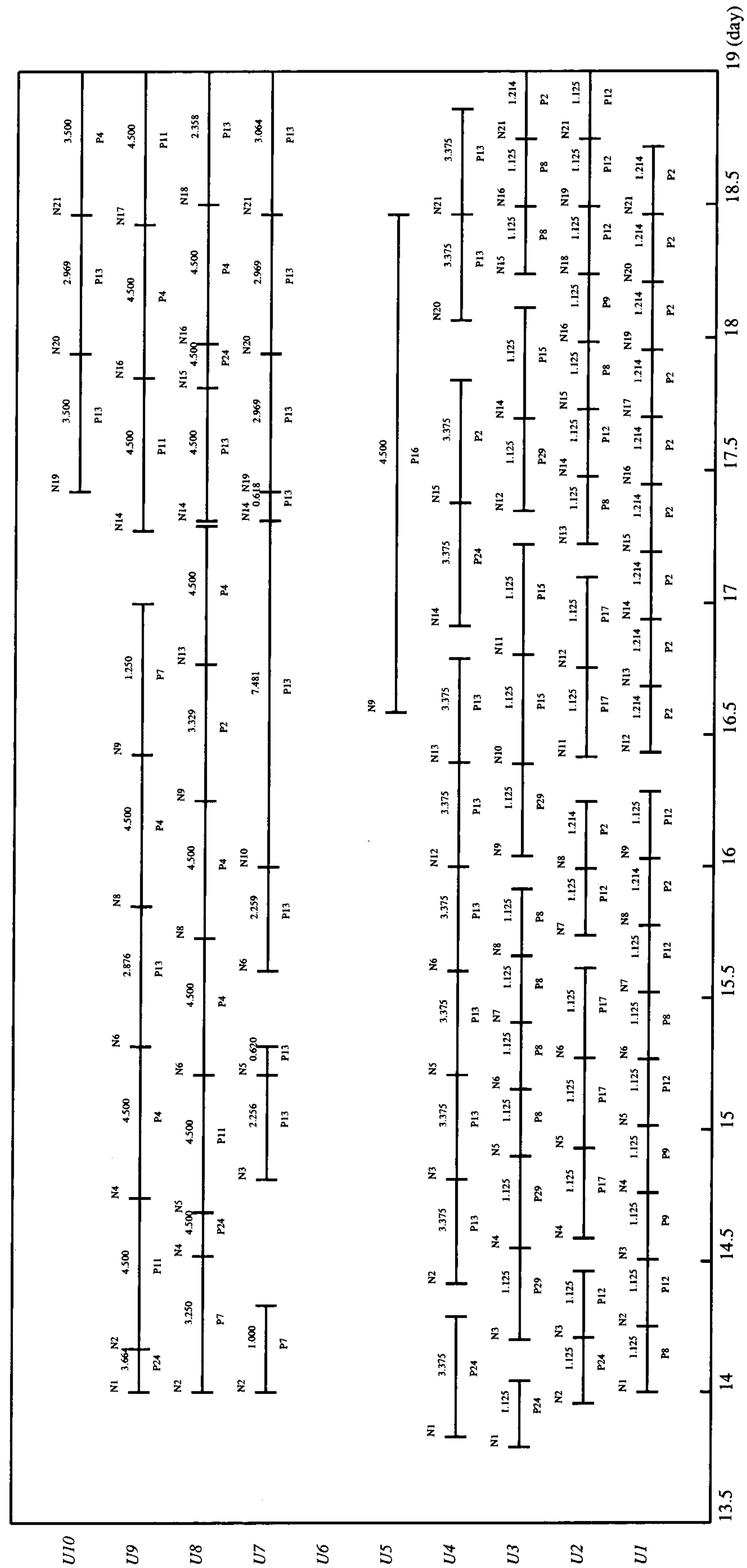

Figure 8. Detailed schedule for time horizon 4 (U1−U10, units; P2−P29, products).

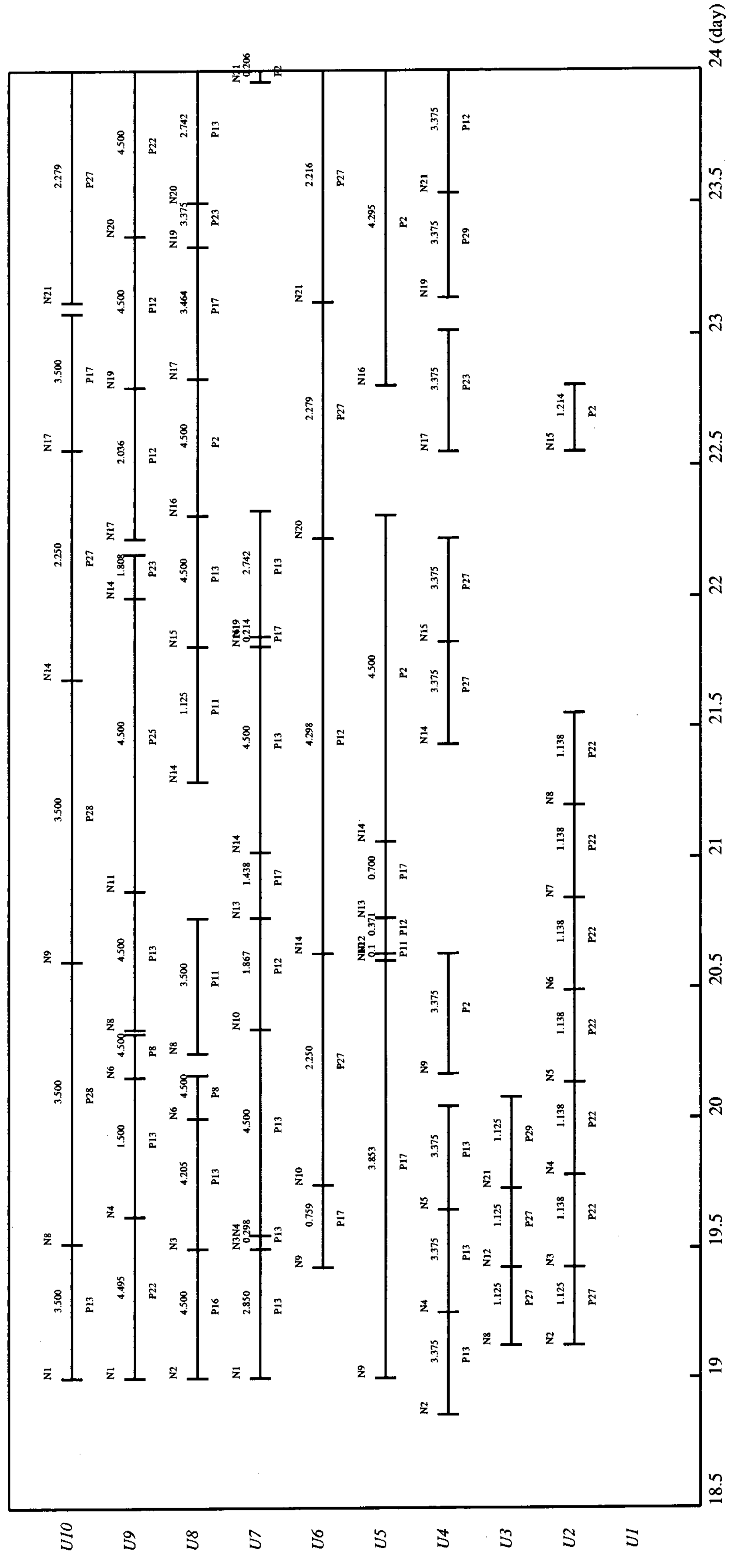

Figure 9. Detailed schedule for time horizon 5 (U1−U10, units; P2−P29, products).

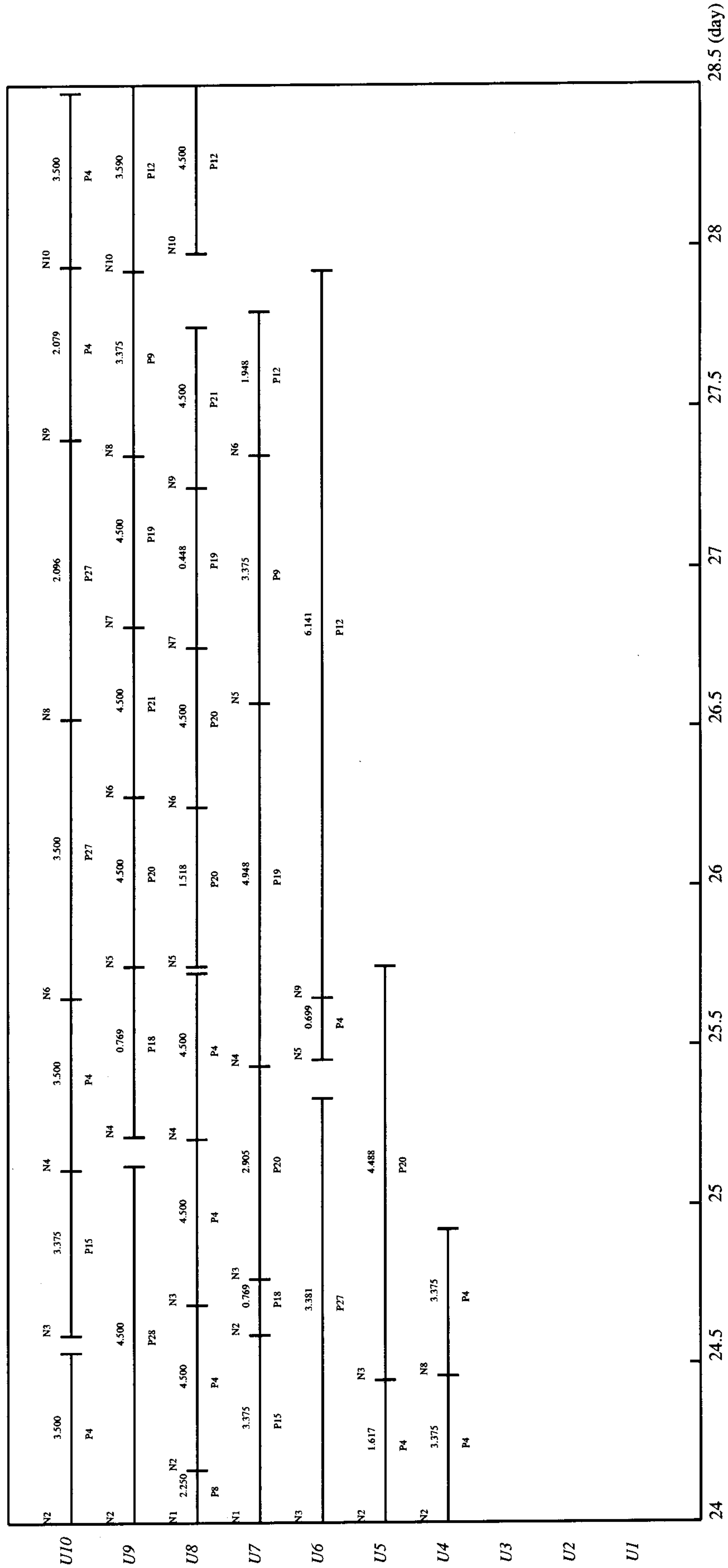

Figure 10. Detailed schedule for time horizon 6 (U1−U10, units; P4−P28, products).

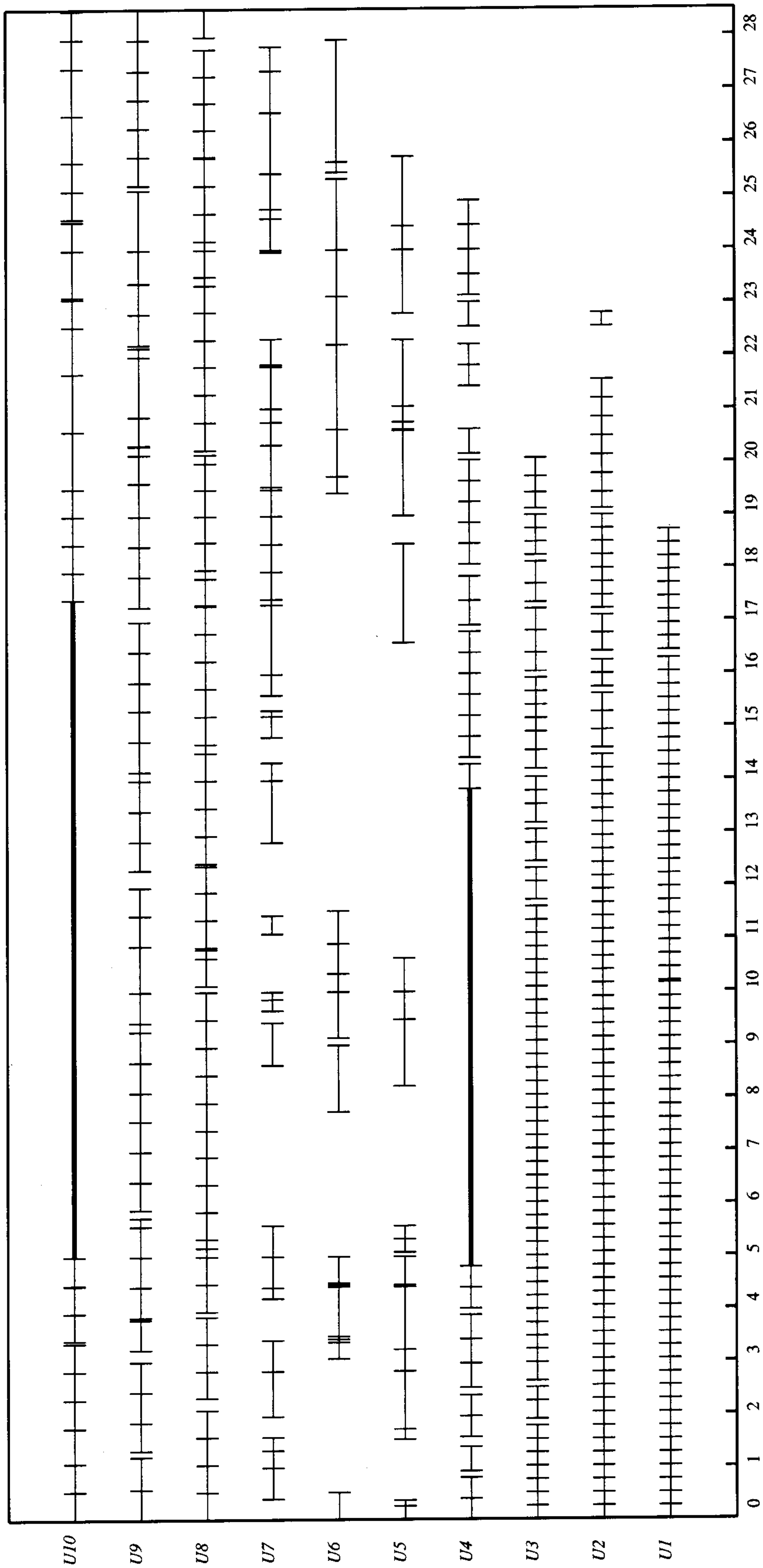

Figure 11. Sketch of the production schedule for the whole month (U1−U10, units).

Horizon 4. The fourth horizon is determined to be from the 15th day to the 19th day of the month, and eight main products are identified to be included according to level 1 of the decomposition model with cplm of 2000. Seven additional products are identified to undergo the operation 1 step from level 2 of the decomposition model with γ_r of 80%.

A total of 21 event points are used in the short-term

Table 1. Decomposition of the Whole Scheduling Period into Successive Horizons

time horizon	1	2	3	4	5	6
no. of days	5	5	4	5	5	4.5
no. of main products	8	6	10	8	9	11
no. of additional products	0	0	0	7	5	0

Table 2. Comparisons of Demands and Production through Proposed Schedules

product	demand	production
category 1	325.8	350.6
category 2	105	113.5
category 3	16.5	17.0
category 4	54.5	61.1
category 5	36.9	45.3
overall	538.7	587.5 (+9.1%)

scheduling model for this horizon, which leads to 1827 binary variables, 5831 continuous variables, and 37 298 constraints. Three feasible solutions are obtained before the last one is accepted. The detailed schedule is shown in Figure 8.

Horizon 5. The fifth horizon is determined to be from the 20th day to the 24th day of the month, and nine main products are identified to be included according to level 1 of the decomposition model with cplm of 2000. Five additional products are identified from level 2 of the decomposition model with γ_r of 30%.

A total of 21 event points are used in the short-term scheduling model for this horizon, which leads to 2016 binary variables, 6471 continuous variables, and 41 866 constraints. One feasible solution is obtained and accepted. The detailed schedule is shown in Figure 9.

Horizon 6. There are only 6 days remaining in the month. It is found that the demands in this remaining period can be fulfilled very easily, and only 4.5 days is actually needed. Therefore, the last horizon is chosen to be 4.5 days starting from the 25th day of the month, and all of the 11 products for the remaining demands are included.

A total of 10 event points are used in the short-term scheduling model for this horizon, which leads to 580 binary variables, 1820 continuous variables, and 6723 constraints. Five feasible solutions are obtained before we accept the last one as a satisfactory schedule. The detailed schedule is shown in Figure 10.

6.3. Summary of the Results. As described in detail in the previous section, the decomposition model and the scheduling model are used iteratively, moving forward the scheduling horizon. In summary, the whole scheduling period is decomposed into six time horizons, each varying from 4 to 5 days long and including 6−15 products, as shown in Table 1.

A sketch of the schedules obtained for the whole month is given in Figure 11. It should be noted that the type 1 units are mostly idle toward the end of the whole period because no demands are specified for the coming period. The production schedules obtained for 28.5 days not only satisfy all demands in all categories, though some of the due dates are relaxed, but also produce 9.1% more than the demands in overall (see Table 2).

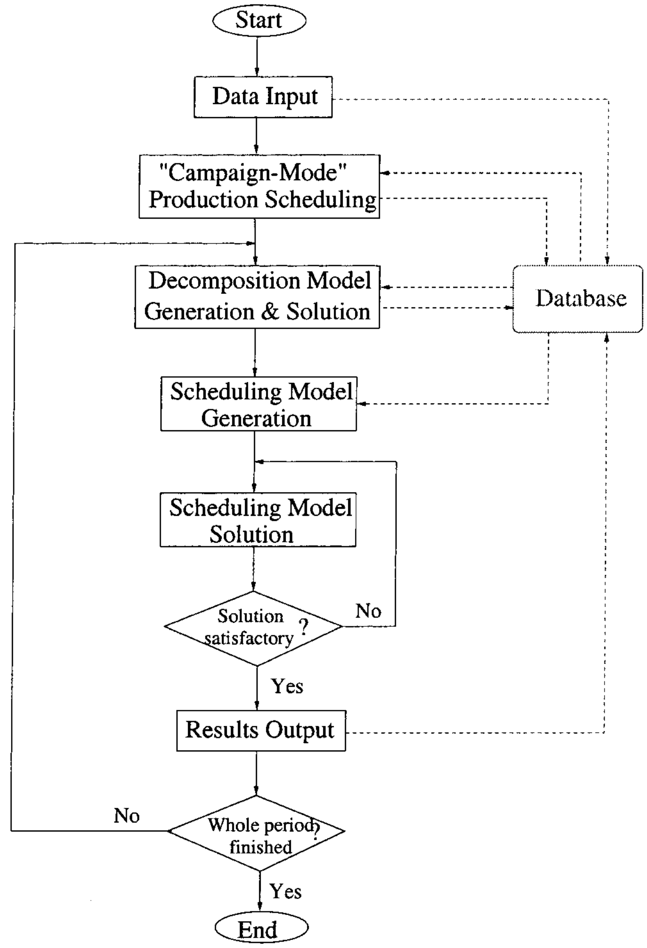

Figure 12. Flowchart of the integrated user interface.

Another important criterion for judging the production schedule is the efficiency of unit utilization. As shown in Table 3, the schedules obtained in this work ensure that the units, especially the type 3 units, are utilized efficiently. The type 3 units are utilized intensively throughout the whole period, which indicates that they are bottlenecks of the overall production as far as the demand structure in this case study is concerned.

7. Integrated Graphical User Interface

A graphical user interface has been developed to integrate various components required to apply the proposed optimization framework to the medium-range production scheduling problem systematically and effectively. The flowchart is shown in Figure 12. After the user inputs all relevant data, which is stored in a database, the scheduling for products that require production in the campaign mode over a relatively long period is performed first. Then, based on the information in the database, the two-level decomposition mode is generated and then solved with an MILP solver, CPLEX. Next, the short-term scheduling model is formulated. The resulting MILP problem is solved iteratively by using cutoff values until a satisfactory feasible solution is obtained. Then the solution is output in readable

Table 3. Unit Utilization

unit	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10
time used (h)	444.6	507.3	457.1	534.8	357.2	346.9	433.8	650.0	652.4	663.8
time used/28.5 days (%)	65.0	74.2	66.8	78.2	52.2	50.7	63.4	95.0	95.4	97.1

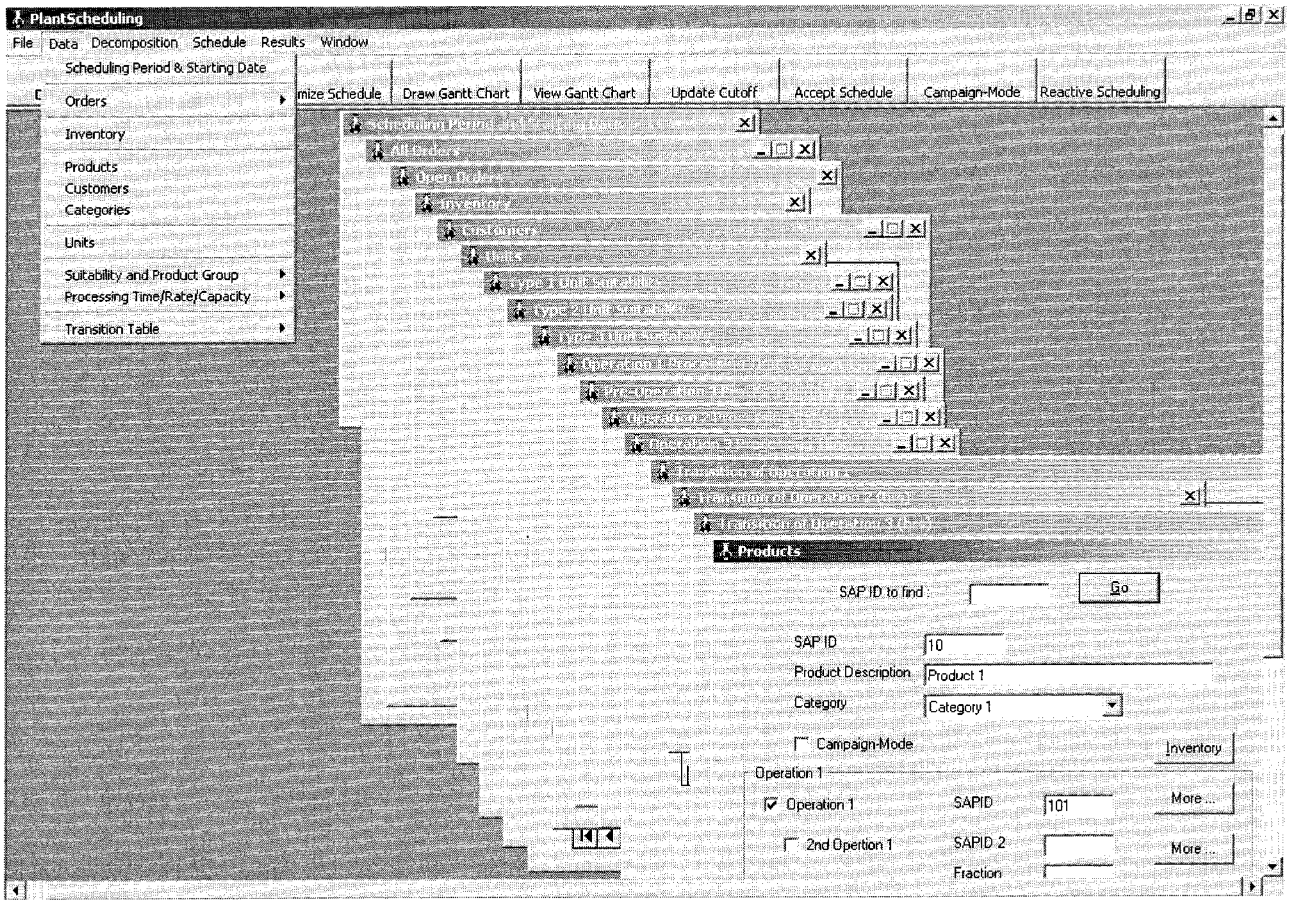

Figure 13. Data forms in the integrated graphical user interface.

formats, for example, the Gantt chart, and the database is updated according to the solution to move toward the next time horizon. The above procedure is applied iteratively until the whole scheduling period is finished. The software is developed in extensive Visual Basic with M. S. Access supporting the database. It consists of the following five main functional modules.

7.1. Data Manipulation. All of the information is stored in a background database, and can be entered, changed, or deleted through the user interface. Figure 13 shows the main window of the user interface with forms opened to access various data.

The important data include (i) scheduling period, (ii) orders, as shown in Figure 14, the amount, due date, and priority for each customer order, (iii) inventories of both final products and intermediate materials, (iv) processing recipes, as shown in Figure 15, with the processing steps required for making each product and other related information needing to be specified, (v) information of the units, as shown in Figure 16, (vi) unit-product suitabilities, for example, Figure 17 showing the type 1 unit suitability form, (vii) processing times/rates, for example, Figure 18 showing the operation 3 processing time form, and (viii) sequence-dependent cleanup requirements, for example, Figure 19 showing the transition table for the type 1 units.

7.2. Campaign-Mode Production Scheduling. Products that are required to go through the campaign mode are handled separately. The appropriate products and suitable units are identified, and a certain period of time is dedicated to the campaign-mode production of these products. Figure 20 shows schedule tables for the campaign-mode production generated in the user interface.

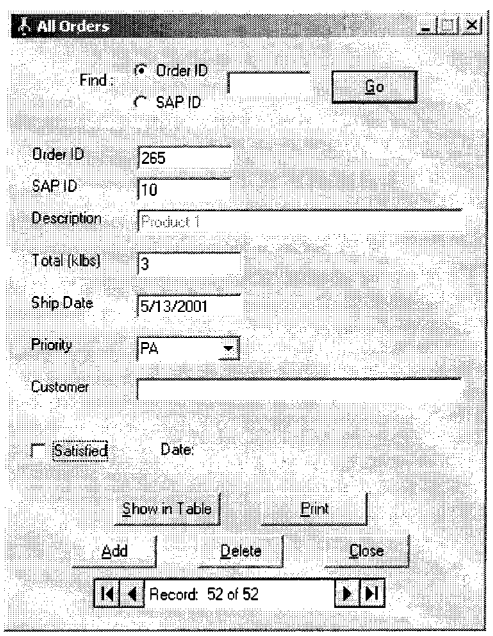

Figure 14. Orders form in the integrated graphical user interface.

7.3. Decomposition. Based on the information in the database, the decomposition model is generated and then solved with an MILP solver, to determine the current time horizon and to identify products to be included for the scheduling problem. Figure 21 shows CPLEX solving the decomposition model, and the results are presented in Figure 22.

7.4. Short-Term Scheduling. According to the solution of the decomposition model, the short-term sched-

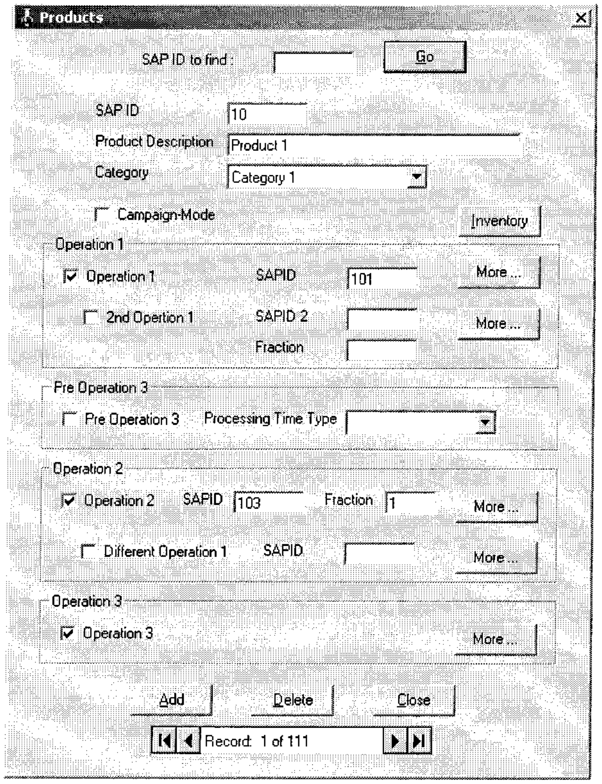

Figure 15. Products form in the integrated graphical user interface.

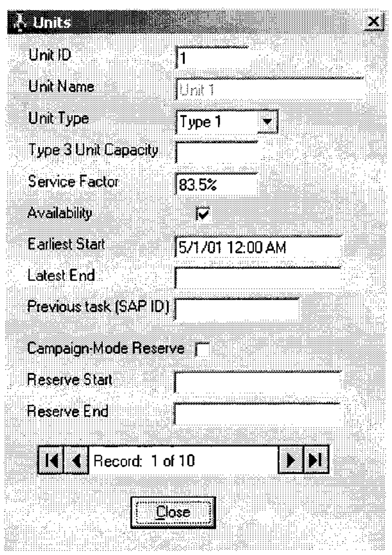

Figure 16. Units form in the integrated graphical user interface.

uling model is formulated to determine the detailed schedule for the current horizon. The resulting large-scale complex MILP problem is solved iteratively by using cutoff values until a satisfactory feasible solution is obtained. Figure 23 shows CPLEX solving the scheduling model.

7.5. Results Output and Database Update. The solution to the short-term scheduling model is organized in readable formats, including the schedule table and the Gantt chart, and the database is updated accordingly. Figure 24 shows the Gantt chart representation of a schedule obtained in the user interface.

Furthermore, a wide variety of additional features are incorporated to make the graphical user interface as user-friendly as possible. For example, the user can

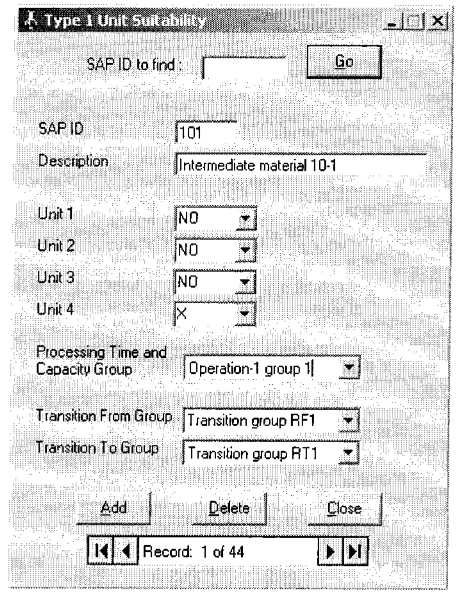

Figure 17. Type 1 unit suitability form in the integrated graphical user interface.

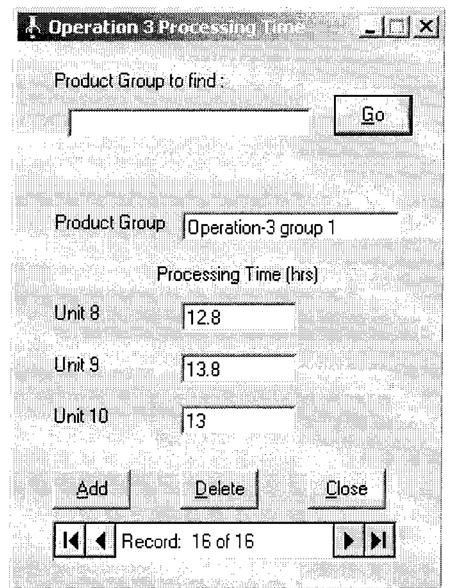

Figure 18. Operation 3 processing time form in the integrated graphical user interface.

request search functions on various data forms to access specific information efficiently.

8. Conclusions

In this paper, the medium-range production scheduling problem of a multiproduct batch plant is investigated. Three basic types of operation are involved, and 10 pieces of equipment are shared to produce up to 60 different products. The scheduling horizon considered is 1 month, even though longer horizons can be addressed with the proposed framework. The overall approach is to decompose the large and complex problem for the whole scheduling period into smaller short-term scheduling subproblems in successive time horizons. A two-level decomposition model is proposed to determine the current horizon and identify those products to be included. Then a continuous-time formulation for short-term scheduling of batch processes with multiple inter-

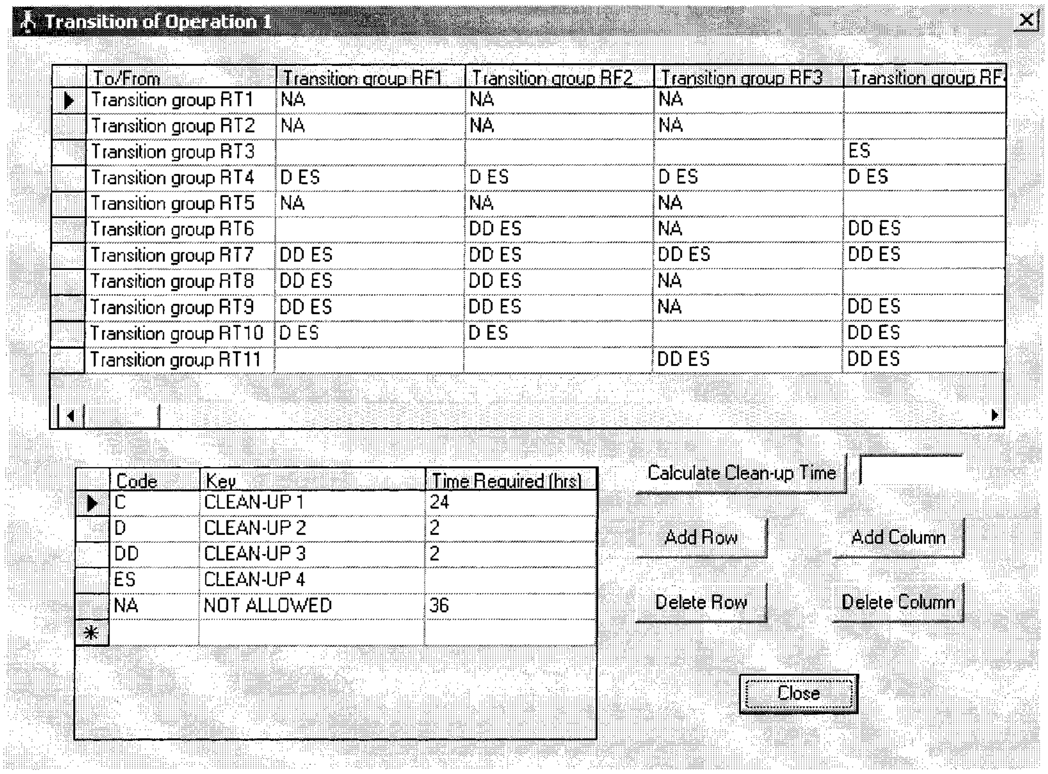

Figure 19. Type 1 unit transition table in the integrated graphical user interface.

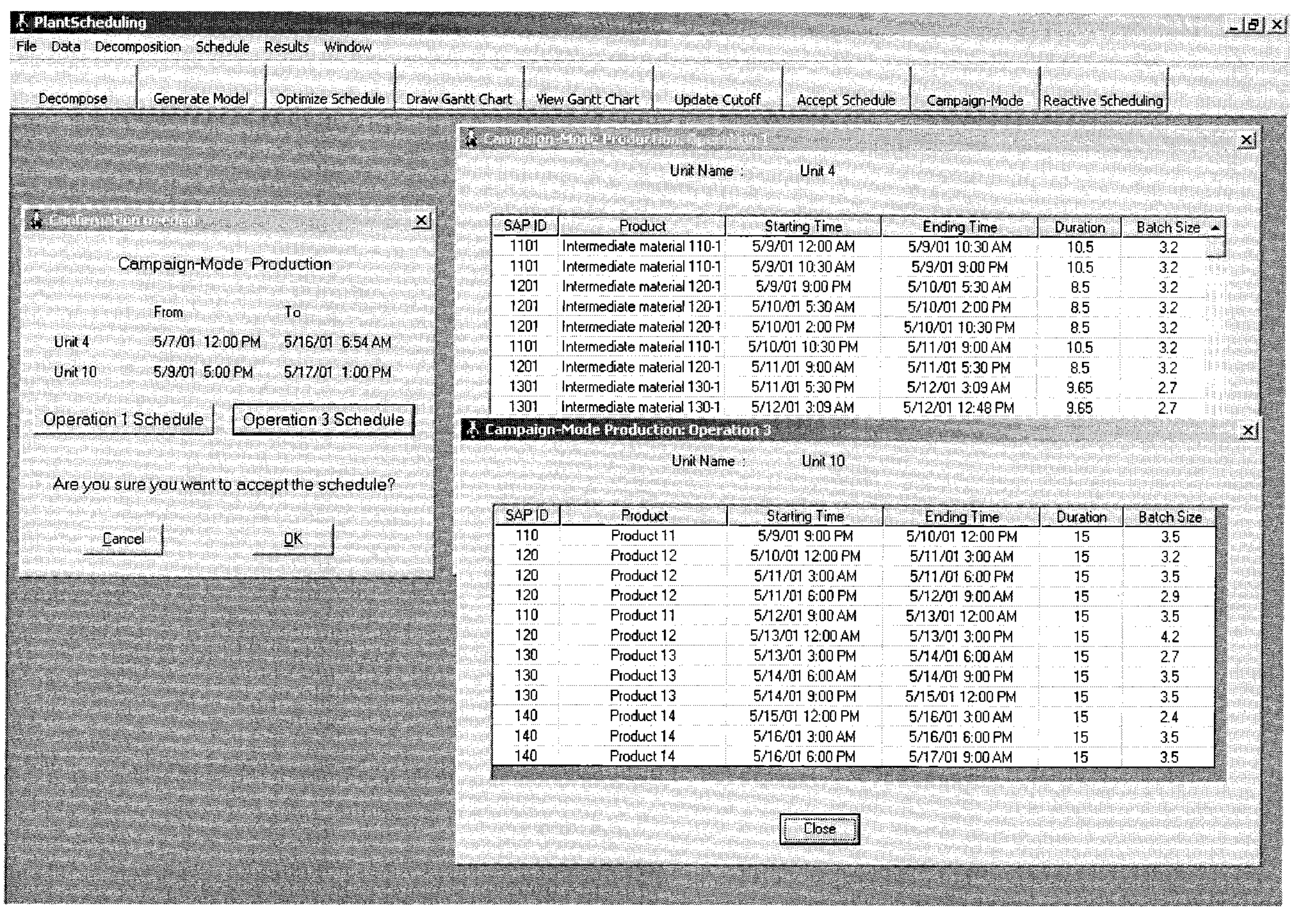

Figure 20. Schedule tables for campaign-mode production obtained in the integrated graphical user interface.

mediate due dates is introduced. This procedure is applied iteratively until the whole scheduling period is completed. The effectiveness of this proposed rolling horizon approach is illustrated with computational results from an industrial case study. A graphical user interface developed to integrate various components to

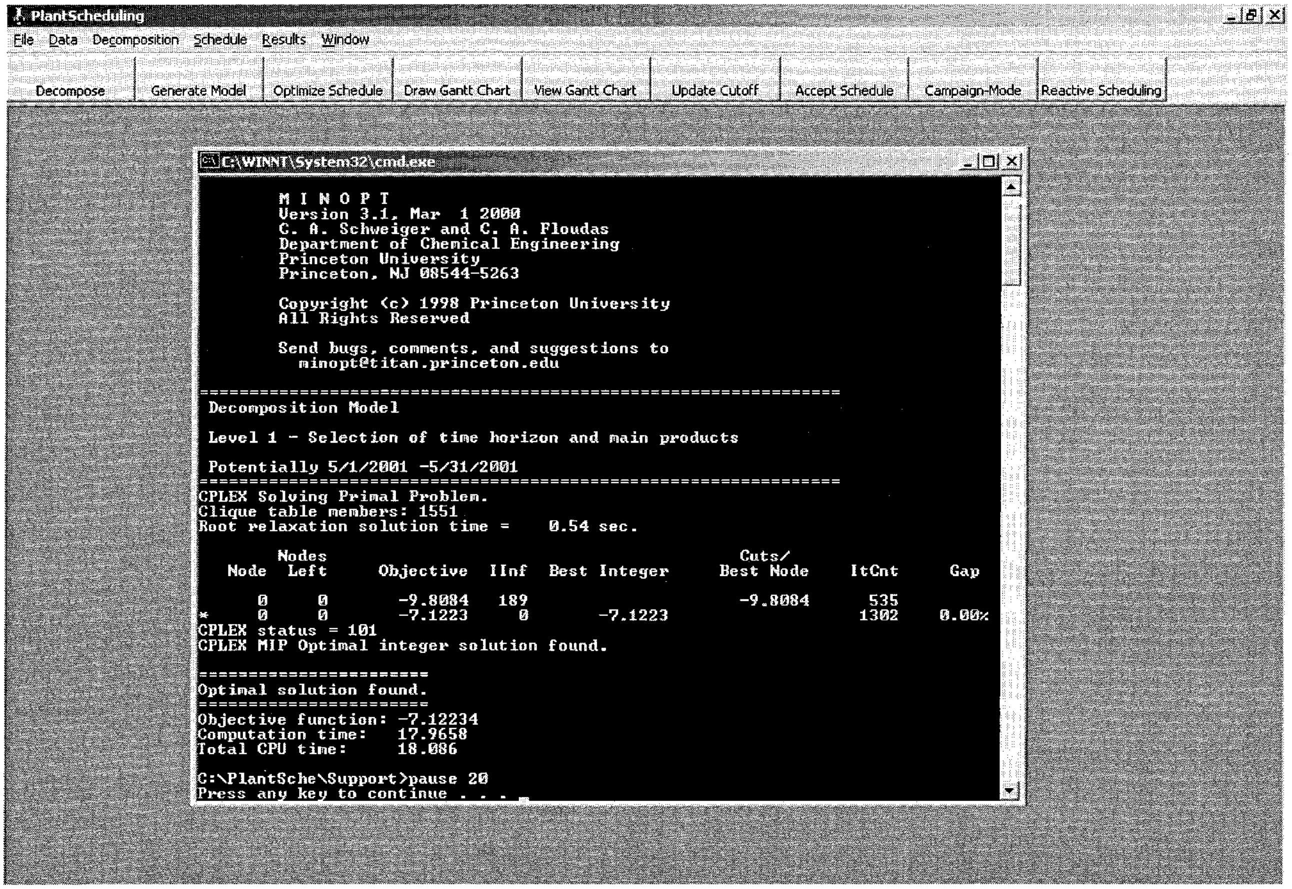

Figure 21. CPLEX solving the decomposition model in the integrated graphical user interface.

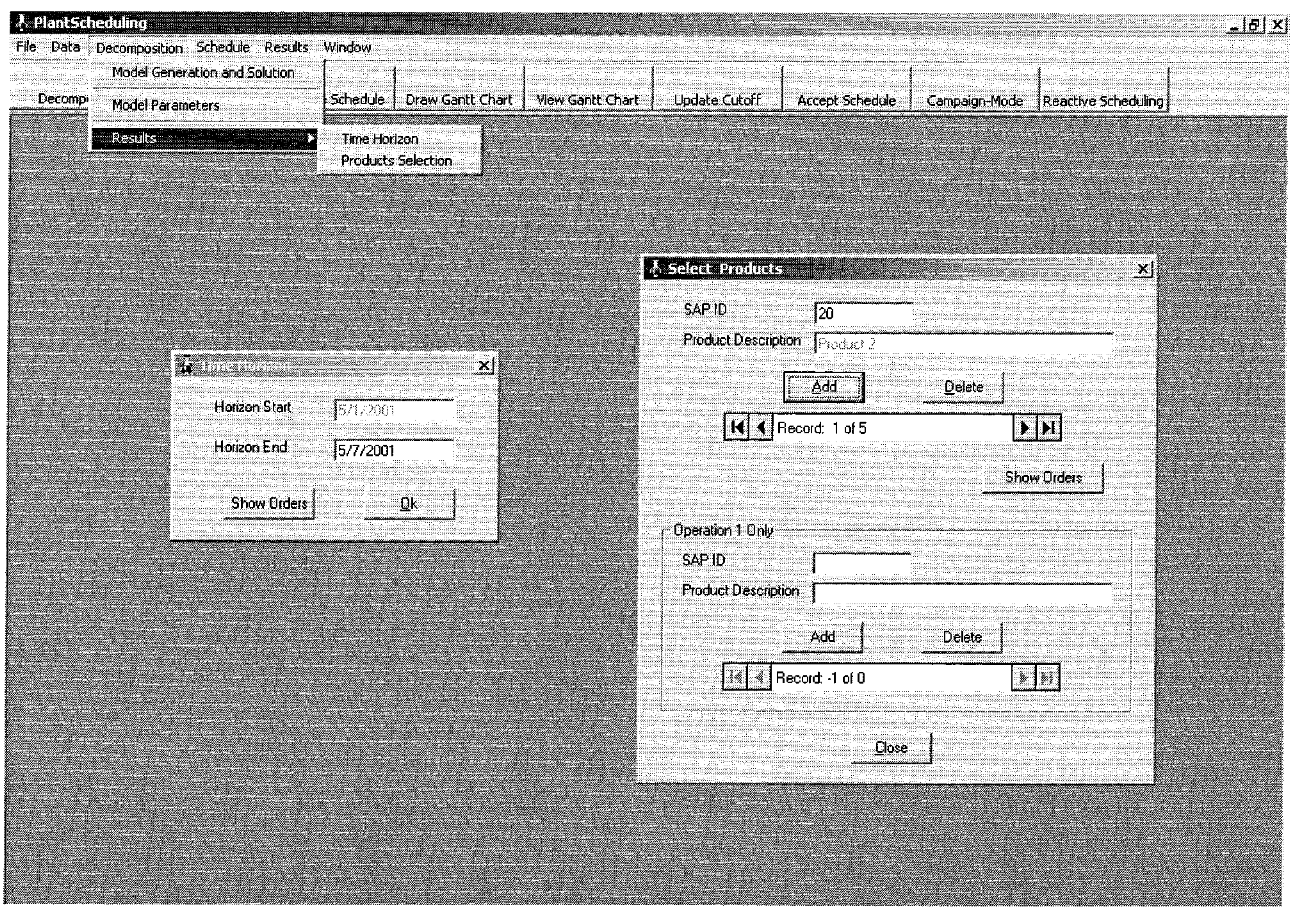

Figure 22. Decomposition results shown in the integrated graphical user interface.

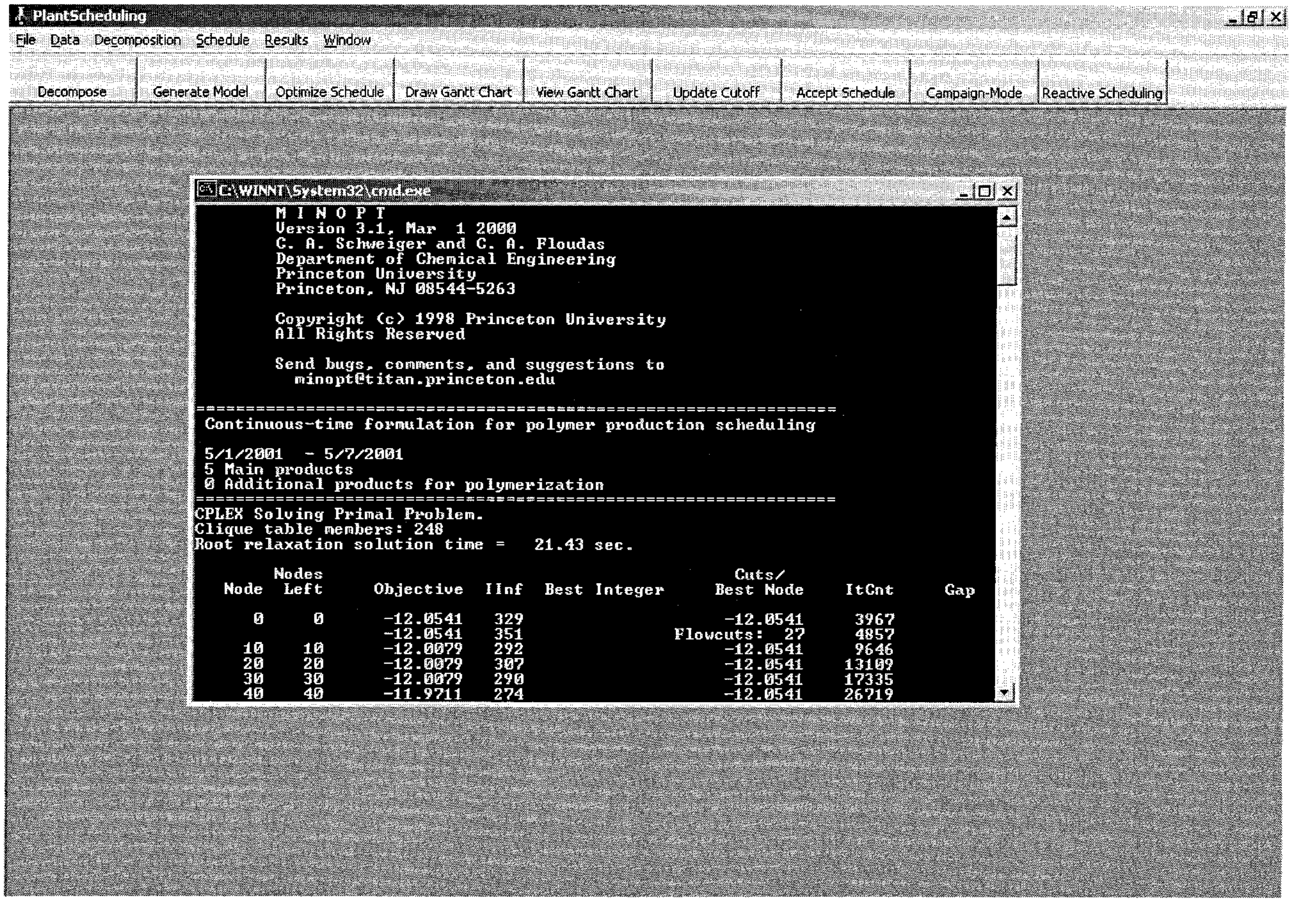

Figure 23. CPLEX solving the short-term scheduling model in the integrated graphical user interface.

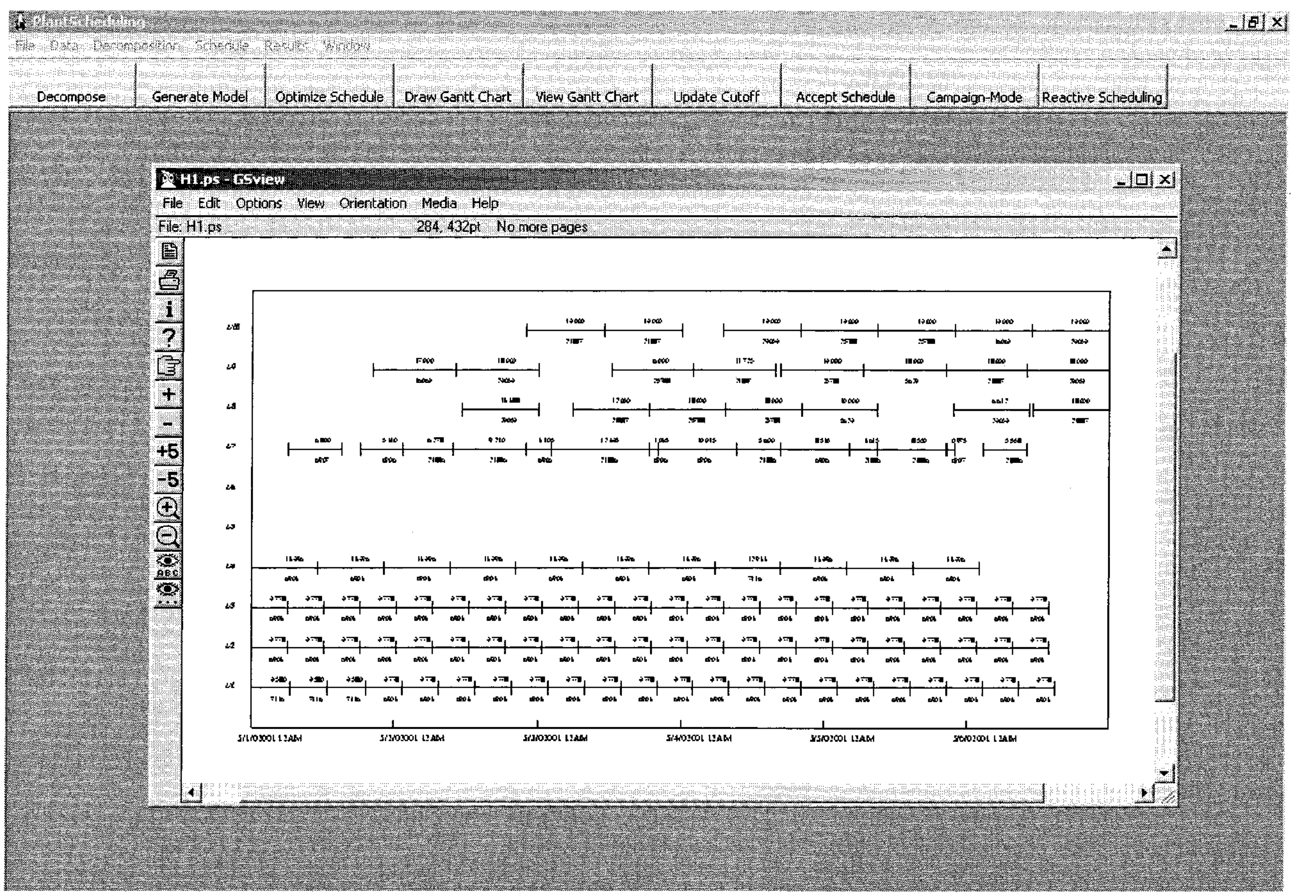

Figure 24. Schedule obtained in the integrated graphical user interface.

apply the proposed optimization framework systematically is also presented.

Acknowledgment

The authors gratefully acknowledge support from the National Science Foundation and from ATOFINA Chemicals, Inc.

Literature Cited

(1) Shah, N. Single- and Miltisite Planning and Scheduling: Current Status and Future Challenges. In *Proceedings of the Third International Conference on Foundations of Computer-Aided Process Operations*, Snowbird, UT, July 5−10, 1998; Pekny, J. F., Blau, G. E., Eds.; CACHE−AIChE: New York, 1998; pp 75−90.

(2) Pekny, J. F.; Reklaitis, G. V. Towards the Convergence of Theory and Practice: A Technology Guide for Scheduling/Planning Methodology. In *Proceedings of the Third International Conference on Foundations of Computer-Aided Process Operations*, Snowbird, UT, July 5−10, 1998; Pekny, J. F., Blau, G. E., Eds.; CACHE−AIChE: New York, 1998; pp 91−111.

(3) Iyer, R. R.; Grossmann, I. E. A Bilevel Decomposition Algorithm for Long-Range Planning of Process Networks. *Ind. Eng. Chem. Res.* **1998**, *37*, 474.

(4) Kondili, E.; Pantelides, C. C.; Sargent, R. W. H. A General Algorithm for Short-Term Scheduling of Batch Operations—I. MILP Formulation. *Comput. Chem. Eng.* **1993**, *17*, 211.

(5) Shah, N.; Pantelides, C. C.; Sargent, R. W. H. A General Algorithm for Short-Term Scheduling of Batch Operations—II. Computational Issues. *Comput. Chem. Eng.* **1993**, *17*, 229.

(6) Pinto, J. M.; Grossmann, I. E. Optimal Cyclic Scheduling of Multistage Continuous Multiproduct Plants. *Comput. Chem. Eng.* **1994**, *18*, 797.

(7) Pinto, J. M.; Grossmann, I. E. A Continuous Time Mixed Integer Linear Programming Model for Short-Term Scheduling of Multistage Batch Plants. *Ind. Eng. Chem. Res.* **1995**, *34*, 3037.

(8) Pinto, J. M.; Grossmann, I. E. A Logic-Based Approach to Scheduling Problems with Resource Constraints. *Comput. Chem. Eng.* **1997**, *21*, 801.

(9) Schilling, G.; Pantelides, C. C. A Simple Continuous-Time Process Scheduling Formulation and a Novel Solution Algorithm. *Comput. Chem. Eng.* **1996**, *20*, S1221.

(10) Zhang, X.; Sargent, R. W. H. The Optimal Operation of Mixed Production Facilities—General Formulation and Some Solution Approaches for the Solution. *Comput. Chem. Eng.* **1996**, *20*, 897.

(11) Mockus, L.; Reklaitis, G. V. Mathematical Programming Formulation for Scheduling of Batch Operations Based on Nonuniform Time Discretization. *Comput. Chem. Eng.* **1997**, *21*, 1147.

(12) Karimi, I. A.; McDonald, C. M. Planning and Scheduling of Parallel Semicontinuous Processes. 2. Short-Term Scheduling. *Ind. Eng. Chem. Res.* **1997**, *36*, 2701.

(13) Ierapetritou, M. G.; Floudas, C. A. Effective Continuous-Time Formulation for Short-Term Scheduling: 1. Multipurpose Batch Processes. *Ind. Eng. Chem. Res.* **1998**, *37*, 4341.

(14) Ierapetritou, M. G.; Floudas, C. A. Effective Continuous-Time Formulation for Short-Term Scheduling: 2. Continuous and Semi-continuous Processes. *Ind. Eng. Chem. Res.* **1998**, *37*, 4360.

(15) Ierapetritou, M. G.; Hené, T. S.; Floudas, C. A. Effective Continuous-Time Formulation for Short-Term Scheduling: 3. Multiple Intermediate Due Dates. *Ind. Eng. Chem. Res.* **1999**, *38*, 3446.

(16) Lin, X.; Floudas, C. A. Design, Synthesis and Scheduling of Multipurpose Batch Plants via an Effective Continuous-Time Formulation. *Comput. Chem. Eng.* **2001**, *25*, 665.

(17) Schweiger, C.; Floudas, C. A. *MINOPT: A Modeling Language and Algorithmic Framework for Linear, Mixed-Integer, Nonlinear, Dynamic, and Mixed-Integer Nonlinear Optimization*; Computer-Aided Systems Laboratory, Department of Chemical Engineering, Princeton University: Princeton, NJ, 1997.

Received for review December 11, 2001
Revised manuscript received April 25, 2002
Accepted April 26, 2002

IE011002A