Multiscale Planning and Scheduling in the Secondary Pharmaceutical Industry

Hlynur Stefansson and Nilay Shah
Centre for Process Systems Engineering, Dept. of Chemical Engineering, Imperial College London, London SW7 2AZ, U.K.

Pall Jensson
Faculty of Engineering, University of Iceland, 107 Reykjavik, Iceland

DOI 10.1002/aic.10989
Published online October 26, 2006 in Wiley InterScience (www.interscience.wiley.com).

Most sophisticated planning and scheduling approaches for the process industry consider a fixed time horizon and assume that all data is given at the time of application. Planning and scheduling approach for a continuous and dynamic decision process where decisions have to be made before all data are available is proposed. As an inspiration we have a real world problem originating from a complex pharmaceutical enterprise. The approach is based on a hierarchically structured moving horizon framework. On each level optimization models are proposed to provide support for the relevant decisions. The levels differ regarding the time scope, aggregation, update rate and availability of data at the time applied. The framework receives input data piece by piece and has to make decisions with only a partial knowledge of the required input. Solution procedures have been developed and the optimization models have been validated and tested with data from the real world problem. The solution procedures were able to obtain good solutions within acceptable computational times. It is believed that multiscale dynamic online procedures are more suitable than traditional offline procedures for many specific types of planning and scheduling problems found in the process industry and should be explored further. © 2006 American Institute of Chemical Engineers *AIChE J,* 52: 4133–4149, 2006
Keywords: production planning, scheduling, real-time optimization, mixed integer programming, integrated multiscale approach

Introduction

The pharmaceutical industry has become a very competitive and unpredictable industry where customers constantly seek low price as well as high service levels and flexibility. Flexible multi–product production processes have become commonly used as they help companies to respond to changing customer demand and increase plant utilization, but the greater complexity of these processes together with the altered market conditions have rendered the relatively simple planning and scheduling techniques previously used insufficient. It is, thus, very important to improve production plans and schedules in order to strive for superior utilization of resources, increased flexibility and reduced response time at the same time as cutting down the cost of production. The general objective of planning and scheduling is to decide what to produce, where and when and how to produce it. The planning activity aims to optimize the economic performance of the enterprise and match production to demand in the best possible way, while the production scheduling translates the eco-

H. Steffansson is also affiliated with the School of Science and Engineering, Reykjavik University, 107 Reykjavik, Iceland.
Correspondence concerning this article should be addressed to N. Shah at n.shah01@imperial.ac.uk.

nomic imperatives of the plan into a sequence of actions to be executed on the plant. It can be hard to distinguish between planning and scheduling problems and problems are often a combination of both as in the case we consider here.

There is an extensive body of literature dealing with planning and scheduling problems in the process industry. Recent reviews of planning and scheduling problems have, for example been written by Floudas and Lin,[1] Kallarath,[2] Pinto and Grossman,[3] and Shah.[4,5] The problem under consideration in this article is composed of two main components which are to create a campaign plan and schedule the orders within the campaigns. Examples of campaign planning problems are presented by Grunow et al.,[6] Papageorgiou and Pantelides,[7,8] Shah and Pantelides,[9] and Tsirukis et al.[10] and examples of scheduling problems are presented by Christofides et al.,[11] Dematta and Guignard,[12] Harjunkoski and Grossmann,[13] Hui and Gupta,[14] Jain and Grossman,[15] Kondili et al.,[16] Mendez and Cerda,[17] Neumann and Schwindt,[18] Pinto and Grossman,[19] and Schwindt and Trautmann.[20]

The approaches found in the literature on planning and scheduling may be differentiated by the representation of time (see Figure 1). Some of the research uses a discrete time axis while some uses a continuous time axis. With a discrete time axis the horizon is divided into a number of equally spaced intervals, but with a continuous time axis the horizon is divided into fewer intervals, and the spacing decided as part of solving the problem. The discrete approaches generally have a common grid used for all resources in the plant but the continuous approaches have either an individual resource grid or a common grid. The choice of time representation is an important factor in the structure and character of a model, and can be used for classification.

The models based on a discrete time representation are in general known to be very flexible and capable of accounting for many planning and scheduling features and different plant layouts. The drawbacks are the approximate time domain representations, and the very large number of binary variables and constraints that result when actual industrial problems are modeled. An approach based on a continuous time representation generally results in a lower number of binary variables and constraints compared to traditional approaches based on a discrete time representation. The constraints required in the continuous time approaches can, however, be more complicate as they do not rely on an uniform time grid and often require additional variables to express temporal relations of tasks, for example, if order 1 is produced before or after order 2. Because of the more complex structure the continuous time approaches can result in greater computational complexity despite having a lower number of variables and constraints compared to the discrete time approaches. So far there is no general opinion in the literature suggesting that either continuous time approaches or discrete time approaches are more computationally efficient or suitable for the real world planning and scheduling problems found in the process industry.

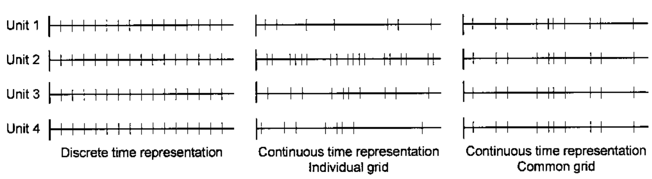

Figure 1. Different time grids used in planning and scheduling models.

The majority of the approaches found in the literature have a drawback for real world applications in order driven production plants such as in the secondary pharmaceutical manufacturing plant under consideration in this paper. The approaches can all be classified as "offline" approaches where the ideal case is assumed and regarded that all data of the problem are given or that the data is subjected to given stochastic uncertainties. In these approaches a fixed time horizon is also assumed instead of considering the problem as a continuous dynamic process where new information can become available throughout the process. The ideal case is of course not very common, and in this article we propose a planning and scheduling approach for a continuous and dynamic decision process where decisions have to be made before all data are available. As an inspiration we have a real world problem originating from a complex pharmaceutical enterprise. Online scheduling as a special case of (combinatorial) online optimization, makes decisions, based on past events and current data without information about future events relevant for the current decision problem and many decisions have to be made before all data are available, and decisions once made cannot be changed.[2] Order-driven production planning and scheduling in the pharmaceutical industry is a critical example of an online optimization problem. Current process data, updated demand forecasts and customer orders are available, but customer orders that will enter the system in the near future, and within the horizon of the current schedule to be determined are not available. The schedule has to be improved step by step where the decision maker does not have access to the whole input instance, unlike the offline case. Instead he or she sees the input piece by piece and has to react to the new requests with only a partial knowledge of the input. The major difference between the online and offline approaches is not in the basic mathematical structure of the models, but rather in the way they are embedded in the overall approach. With some modifications most of the offline models can be used in an online fashion by revealing data periodically and resolving the model with some parts of the schedule frozen to respect previously made decisions. These modifications would involve changes or additions in mathematical formulation, most likely increasing the size and complexity of the models, and as a result affecting the proposed solution procedures.

The literature on online planning and scheduling problems is not extensive. A review of heuristics for theoretically oriented online scheduling problems is presented by Sgall.[21] In the review, problems are classified into four different categories, based on which part of the problem is given online. The chosen categories are: the problem of scheduling jobs one by one, the problem with unknown running times, the problem where jobs arrive over time, and the problem where interval scheduling is involved.[21] Real world applications can belong to one of these categories, be a combination of more than one of these problems or even possess more complicated characteristics than the categories described.

Some practical online scheduling approaches are also described in the literature. Sand and Engell[22] and Sand et al.[23] describe online scheduling algorithms from an industrial multiproduct batch plant producing expandable polystyrene, where demand and yields are subjected to stochastic uncertainties.

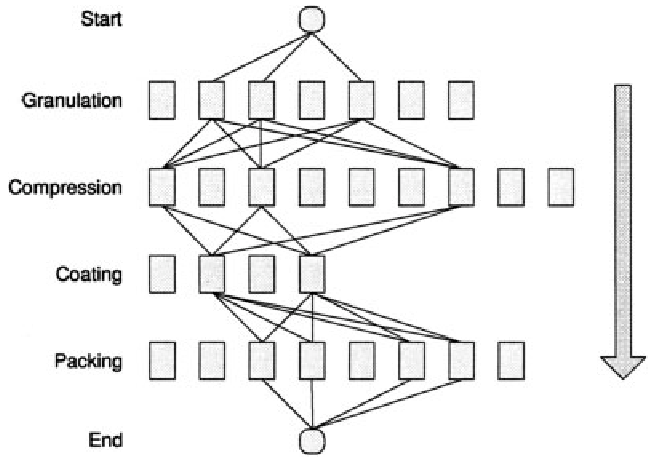

Figure 2. Production process and the possible production routes for one single product through the process.

Engell et al.[24] structure the overall, problem hierarchically into a "long-term" stochastic linear planning problem, and a "short-term" deterministic nonlinear scheduling problem subject to uncertainty in demand and yield. Sand and Engell[22] use a rolling horizon, two-stage stochastic programming approach to schedule the production subject to uncertainty in processing times, yields, capacities and demands. The uncertainty in processing times and yields is considered "short-term", and the uncertainty in capacities and demands is considered "medium-term". A hierarchical scheduling technique is used where a master schedule deals with medium-term uncertainties, and a detailed schedule with the short-term ones. The uncertainties are represented through discrete scenarios and the two-stage problem solved using a decomposition technique. Engell et al.[24] also introduce conceptual considerations concerning online scheduling and suggest an interesting telescopic decomposition approach with a number of layered submodels of different degrees of temporal aggregation for solving online scheduling problems in multiproduct batch plants.

Dynamic scheduling in multiproduct batch plants is considered by Mendez and Cerda,[25] and resource constraints also included by Mendez and Cerda.[26] Their approach is based on a continuous time model that takes into account the schedule currently in progress, the updated information on old production batches still to be processed and new-order arrivals, the present plant status and the availability of renewable discrete resources. The approach is based on a model presented by Mendez et al.[27] To avoid full-scale rescheduling, the approach only allows partial modifications to the schedule in progress by iteratively solving the MILP problem until no further improvement on the current schedule is obtained, and, hence, does not guarantee the optimal solution although a good one should be possible in many cases. This approach is able to cope with some of the dynamic features involved in the order-driven production process. However the number of orders and the complexity of our secondary pharmaceutical planning and scheduling processes with its campaign production make the approach unsuitable for the problem under consideration here. The difference here between dynamic and online scheduling is the attempt to see beyond the existing data.

Problem Description

Our study is based on a real world problem originating from a pharmaceutical enterprise whose competitive advantage relies on cutting down time to market and delivering a fast response to customers with high-service level and reliability. At the same time the manufacturer strives to reduce production costs and maximize the utilization of assets in order to increase margins. We focus on single plant production planning and scheduling for a secondary production facility with order-driven multistage, multi-product flowshop production. The plant consists of a large number of multipurpose production equipment items at each production stage, operated in batch mode. When switching between batches containing products from different product families large sequence dependent setup and cleaning times are required but these are much smaller when switching between batches containing products within the same product family. The plant, therefore, uses campaign production, where a campaign is an ordered set of batches containing products from the same product family produced consecutively by the same machine. Each product has a number of different feasible production routes through the plant as shown in Figure 2, and as the number of product families is over 40, and product variations more than 1,000, the process of planning and scheduling the production in an optimal way becomes extremely complicated.

The overall goal of the problem is to determine a campaign plan and to schedule customer orders within the campaigns. The customers request certain delivery dates for their orders and the plant attempts to meet those requests. In order to make the plans and schedules feasible the most important constraints from the operational environment must be respected. These constraints include allocation constraints, sequencing constraints, delivery constraints, capacity constraints, campaign constraints, mutual exclusivity constraints, and so on. The general objective of the plans and schedules is to meet the quantity and delivery date of customer orders and minimize the unproductive production time in order to maximize the economic performance of the company. For a further description of pharmaceutical production plants and supply chains see Shah.[28]

Modeling Approach

Online and dynamic characteristics of the problem

It is our goal to develop a state of the art approach including models that are realistic and can be applied in actual circumstances. To do that it is necessary to acknowledge and cope with the continuous dynamic decision process involved, and the online characteristics of the problem under consideration. Order driven production planning and scheduling in the phar-

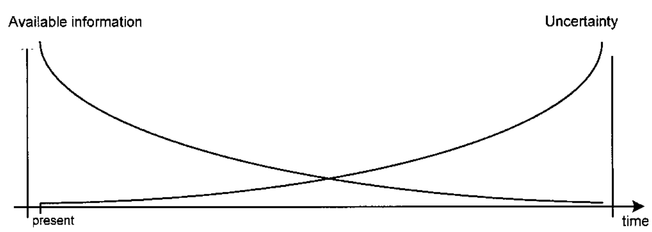

Figure 3. Explanation of how information becomes available as the time horizon moves closer to the present time point and as a result the uncertainty in the decision problem decreases.

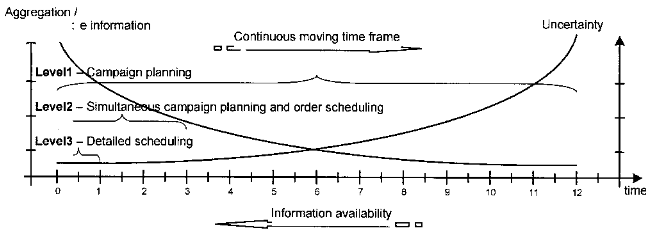

Figure 4. Hierarchical framework drawn together with the representation of uncertainty and availability of information with regard to time.

maceutical industry is a critical example of a continuous and dynamic online optimization problem. The production is an ongoing process that is affected by several uncertain inputs; the most important one is the demand from customers.

The first task of the planning process is to create a campaign plan for long-term planning purposes. The campaign structure has to be decided on earlier than when the orders become available and demand predictions are used instead of the actual orders. The material requirement planning and purchasing of raw materials from suppliers is based on the campaign plan, and needs to be performed before the orders become available as the suppliers' lead-times are often longer than the promised lead-time of the production. Also maintenance, shifts, launches of new products and other events must often be planned in advance of orders being placed.

Then the plant receives new customer orders each week with a requested delivery date, and the orders need to be scheduled and feedback given to customers with a confirmed delivery date. When the orders are scheduled within the campaigns the decision maker does not have the information about other orders that will be added to the campaigns but have not been received yet. The current orders can change, and other unanticipated orders may appear later with earlier due dates, larger quantities or higher priorities which can make it necessary to change the current schedule. However the confirmed delivery dates that have already been promised to customers must be respected if at all possible.

The decision maker does not have access to the whole input instance at the time decisions need to be made and instead he or she sees the input piece by piece and has to react to the new requests with only a partial knowledge of the input. Once decisions have been made it can be difficult to change them later on. As more information becomes available the uncertainty of the decision problem decreases (Figure 3), and as a result all decisions should generally be made as late as possible though respecting interactions and relationships with other decisions and requirements. In addition to the uncertainty involved in the demand, there are some other sources of uncertainty, such as uncertainty in processing times, setup and cleaning times and uncertainty caused by machine breakdowns, employees' absence, quality failures etc. However, the uncertainty caused by the demand is the most significant factor in this case.

Modeling techniques

We apply mathematical programming techniques in our approach. One of the major difficulties in building mathematical programming models is to keep the size of the models, that is, the number of variables and constraints, within reasonable limits. When the model size becomes very large it can be difficult to obtain results in an acceptable computational time. However, the accuracy of the models should not be sacrificed, and the models must capture the required details to make valid decisions for each optimization problem. Realworld scheduling tasks belong to the class of NP-hard problems, that is, there are no known solution algorithms of polynomial complexity in the problem size. To solve real-world scheduling problems efficiently, simplification, approximation or aggregation strategies are most often necessary.[6,24]

We propose an aggregation strategy with a hierarchically structured approach for planning the campaigns and scheduling the orders for the real-world problem under consideration. The decisions that need to be made differ by level of detail, scope and time horizon and the relevant data differ by its degree of certainty and aggregation and, hence, a hierarchically structured approach as explained in the next section should be well suited for the problem.

Hierarchically structured framework

To cope with the continuous dynamic decision process we propose a multiscale hierarchically structured framework shown in Figure 4. On each level of the hierarchically structured framework we propose optimization models to provide support for the relevant decisions. The levels differ regarding the scope and the availability of information at the time when they are applied.

This is a continuously moving framework as the nature of the actual decision problem, and the models are used with a different frequency depending on their level of aggregation and the update rate of their input factors. The most detailed decisions are made as late as possible when more information is available and the uncertainty has decreased. It does not make sense in general to create detailed decisions over the full horizon since uncertainties make detailed decisions obsolete soon after they are created, and as a result will they not be used. By solving problems and obtaining results that will not be used, optimization effort is wasted which reduces the overall efficiency in real-time applications.

Levels in the Framework

Campaign planning at level 1

At the top level in the hierarchical framework an aggregated optimization model is used to optimize the long-term

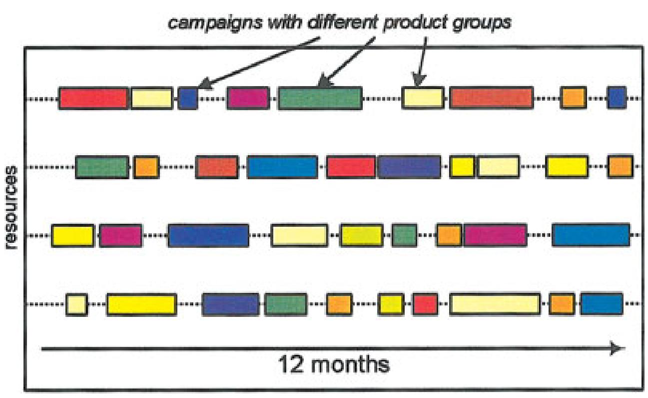

Figure 5. Campaign plan for level 1.

[Color figure can be viewed online in the online issue, which is available at www.interscience.wiley.com.]

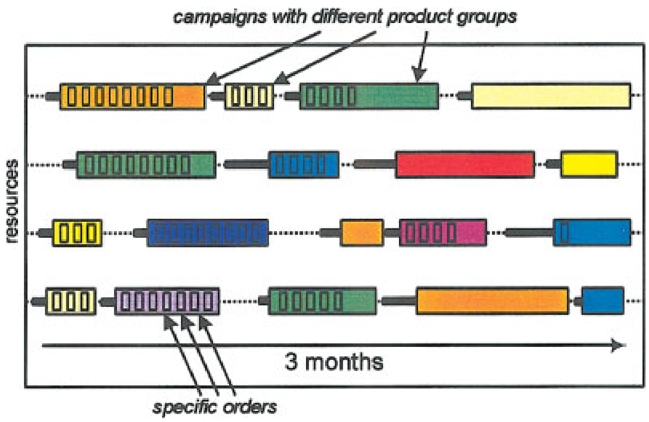

Figure 6. Campaign plan and order allocation for level 2.

[Color figure can be viewed online in the online issue, which is available at www.interscience.wiley.com.]

campaign plan with the objective to fulfill demand and minimize cost of production. This level can be regarded as the first step in the planning process. The input for this model is mainly based on a combination of *sales forecasts*, and long-term customer orders, as well as information regarding products, production process, performance and current status of production. The output from the model is a campaign plan that includes all machines on all production stages and specifies the starting time of campaigns, the product family to be produced in the campaign, the quantity, the duration of the campaign and the machine where the campaign will be operated (see simplified campaign plan in Figure 5). The campaign plan is used for making raw material procurement plans and for other long-term planning purposes.

The time horizon of this model is 12 months, and it is updated every 3 months or more often if new sales forecasts or other aggregated high-level information becomes available. We use mathematical programming techniques to formulate the model, and propose a novel and efficient mixed integer linear program with a discrete representation of time.

Campaign planning and order scheduling at level 2

At the middle level in the hierarchical framework an optimization model is used to revise the campaign plan and allocate orders and other production tasks, within the campaigns. The campaign plan is now based on more accurate input information compared to the earlier level as the *customer orders* are more or less available here. The output from the model is a revised campaign plan and a production schedule for the orders (see Figure 6). The production schedule specifies in which campaign each order is produced on every production stage, and it also specifies the latest allowed completion time for the order which is used to report to customers the confirmed delivery date of the order. After the delivery dates have been confirmed it is important to respect them although the production schedule can be changed in many other ways.

The time horizon for this level is three months, and the optimization model is used once every week when customer orders are scheduled and their delivery dates confirmed, or when needed as new information becomes available. For this level we also propose a novel and efficient mixed integer linear program with a discrete representation of time.

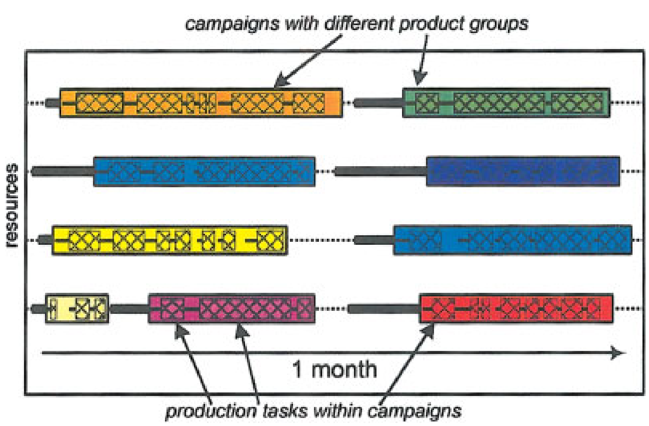

Figure 7. Campaign plan for level 1.

[Color figure can be viewed online in the online issue, which is available at www.interscience.wiley.com.]

Detailed scheduling at level 3

At the lowest level in the hierarchical framework is an optimization model used for the detailed scheduling of production tasks. The optimization is based on *confirmed-customer orders* together with the newest possible information each time it is used, and the resulting production schedule specifies on which machine and in which campaign each order is produced on each production stage, the production sequence of the orders, the precise start time and duration of processing tasks and the setup tasks required between orders (see Figure 7). The model accounts for sequence dependent setup times between the campaigns, as well as between the orders within the campaigns.

The time horizon is one month and the model should be used every day if some new information has become available that affects the feasibility or optimality of the last production schedule. For this level we propose a continuous-time model which allows a detailed resolution of time.

Integration of levels

There is a directed flow of information between levels. Results from the higher level models are transferred and used as input for the lower level models as can be seen in Figure 8.

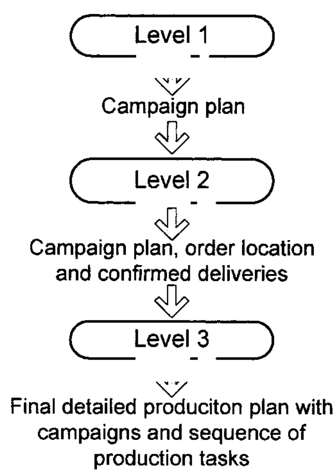

Figure 8. Directed flow of information in the hierarchical structure.

From the first level the campaign plan is transferred and used as input for the second level. The second level is, however, not required strictly to respect the campaign plan, and is allowed to restructure it if it results in an improved schedule. The campaign plan and confirmed delivery times are transferred to the third-level together with allocation of orders in the campaigns, which the third level model can change to improve the schedule and react to more detailed information.

When the flow of information between hierarchically structured levels is directed it is very important to ensure that the results transferred from higher-levels provide feasible input for lower-levels. If that is ensured, monodirectional integration should be sufficient. However if not, it may be necessary to transfer information from the lower levels and develop bidirectional integration of levels through some feedback loops. The feedback loops can in principle deliver first and second-order information about infeasibilities, constraints that are violated, sensitivity information and other knowledge that will make the framework consistent.

Optimization Models

Campaign planning for level 1

Here, we employ a discrete-time model at a weekly level of resolution.

Variables

Continuous Variables. $\varepsilon_i =$ Continuous variable for delay of order *i,* equal to 0 if no delay will occur for order *i,* else equal to the difference between estimated completion time of order *i* and the due date for the order ($\varepsilon_i \geq 0$)

Binary Variables. $X_{i,w,j} =$ decision variable equal to 1 if order *i* is processed in week *w* on machine *j,* else equal to 0.

$Z_{w,j} =$ decision variable equal to 1 if setup work is needed in week *w* for machine *j,* else equal to 0.

$Y_{f,w,j} =$ decision variable equal to 1 if product family *f* is produced in week *w* on machine *j,* else equal to 0.

Constraints

Allocation Constraints. Every order should be allocated to exactly one machine on each production stage $s \in S_i$

$$\sum_{j \in J_{i,s}} \sum_{w \in W_{i,j}} X_{i,w,j} = 1 \qquad \forall \; i \in I, s \in S_i \qquad (1)$$

where $X_{i,w,j} = 1$ if production of order *i* takes place in week *w* on machine *j,* else 0. $X_{i,w,j}$ is only defined for weeks where the correct product family will be produced and only for machines *j* that can produce order *i* on each production stage.

Sequencing Constraints. The production activities on each production stage must be performed according to a correct production route and sequence, and production of an order must not start on certain stage before it is finished on the preceding stage

$$\sum_{j \in J_{i,j}} \sum_{w \in W_{i,j}} X_{i,w,j} \cdot (w + p_{i,j} + q_{i,s}) \leq \sum_{j \in J_{is+1}} \sum_{w \in W_{ij}} X_{i,w,j} \cdot w \quad \forall \; i \in I, s \in S_i \backslash s_i^{last} \quad (2)$$

where *w* is the week number, $p_{i,j}$ is the production time of order *i* on machine *j,* and $q_{i,s}$ is the time that needs to pass after each production stage for example, to allow for quality inspections. To ensure the correct sequence of production stages we assume that each order can only be processed at one production stage in each time bucket (week). A campaign containing one or more orders of the same product family *f* can however take place at stages *s* and *s* + 1 in the same week *w*. To reduce the possible minimum time span of production when an order requires more than one week of production time at some or all stages, the order is divided into two or more linked orders that each require less than one week of production time. The resolution of the time scale for the long term planning at level 1 was chosen to be one week which limits the accuracy of the model but is an appropriate compromise between computational efficiency and accuracy. Further explanation of campaigns, orders and time related parameters can be found in Figure 9.

The production of orders cannot start before the orders have been released or before the production process is ready, specified by the earliest start time r_i

$$r_i \leq \sum_{j \in J_{i,s}} \sum_{w \in W_{i,j}} X_{i,w,j} \cdot w \qquad \forall \; i \in I, s = s_i^{first} \qquad (3)$$

Other sequence related constraints are not included at this level of the hierarchical structure.

Delivery Constraints. The production plan is supposed to respect the due dates specified by the demand forecast. If that is not possible the variable ε_i in constraint (Eq. 4) measures the delay.

$$\sum_{j \in J_{i,s}} \sum_{w \in W_{i,j}} X_{i,w,j} \cdot (w + p_{i,j} + q_{i,s}) \leq d_i + \varepsilon_i \qquad \forall \; i \in I, s = s_i^{last} \quad (4)$$

d_i is the delivery time of order *i,* ε_i is lateness of the delivery, and $q_{i,s}$ here represents the time required for quality checks and other mandatory activities before the order can be shipped to the customer.

Capacity Constraints. The structure of the models allows straightforward formulation of various capacity constraints.

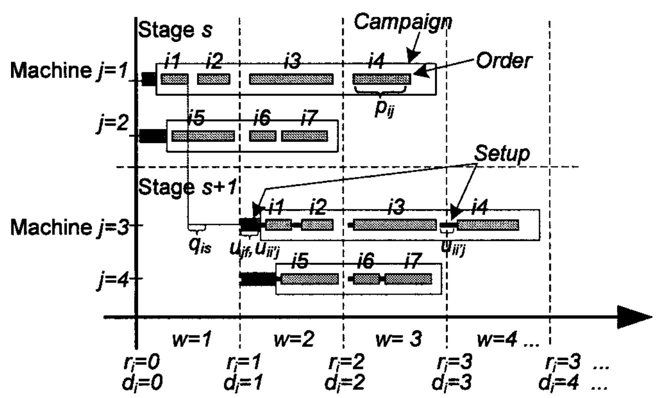

Figure 9. The structure of campaigns and orders at different production stages and also explains the use of various parameters in the models.

Setup times between campaigns are included in the models proposed for level 1 and 2, and in addition to that are setup times between individual orders within each campaign included in the model proposed for level 3.

Here we include constraints for limited capacity of each machine and limited capacity of the factory as whole

$$\sum_{i \in I_j} X_{i,w,j} \cdot p_{i,j} \leq pt_{j,w}^{\max} \quad \forall\, w \in W, j \in J_w \tag{5}$$

$pt_{j,w}^{\max}$ is the total time available for production in each week w on machine j

$$\sum_{j \in J_w} \sum_{i \in I_j} X_{i,w,j} \cdot p_{ij} \leq pt_w^{\max} \quad \forall\, w \in W \tag{6}$$

$pt_w^{\max}$ is the total production time available in the factory each week. If a single order requires more than one week of production time at some or all stages then the order is divided into two or more linked orders that require less than one week of production time and constraints Eqs. 5 and Eq. 6 will still be feasible.

Campaign Constraints. To define the binary variable $Y_{f,w,j}$, indicating that product family f is produced on machine j in week w we use constraint (Eq. 7).

$$Y_{f,w,j} \geq X_{i,w,j} \quad \forall\, f \in F, i \in I_f, j \in J_f, w \in W_{i,j} \tag{7}$$

Constraint (Eq. 8) activates the binary variable $Z_{w,j}$ indicating that a new campaign is starting and setup work is needed in week w on machine j.

$$Z_{w,j} \geq Y_{f,w,j} - Y_{f,w-1,j} \quad \forall\, f \in F_j, j \in J_f, \quad w \in W \backslash w^{first} \tag{8}$$

It is often required that only one product family is produced in each week on each machine

$$\sum_{f} Y_{f,w,j} \leq 1 \quad \forall\; w \in W, j \in J_w \tag{9}$$

A constraint can also be used to avoid intermediate weeks in campaigns with no production. If nothing is produced in the week then the campaign will finish, and a setup will be needed to start the production again. If this constraint is not included and the capacity is not restricted then the model may suggest long campaigns with sparse production in order to reduce setup cost.

$$Y_{f,w,j} \leq \sum_{i \in I_f} X_{i,w,j} \quad \forall\; f \in F, w \in W, j \in J_f \tag{10}$$

A maximum and/or minimum number of campaigns on each production stage should be respected

$$z_s^{\min} \leq \sum_{w} \sum_{j \in J_s} Z_{w,j} \leq z_s^{\max} \quad \forall\, s \in S \tag{11}$$

where $z_s^{\min}$ and $z_s^{\max}$ are, respectively, the minimum and maximum number of campaigns on stage s.

Mutual Exclusivity Constraints. It may be forbidden to produce a certain pair of products (orders) at the same time on the same production stage

$$\sum_{J_s} X_{i,w,j} + \sum_{Js} X_{i',w,j} \leq 1 \quad \forall\, i \in I, i' \in IS_{i,i'}, s \in S_i, w \in W \tag{12}$$

where $IS_{i,i'}$ is a set of orders that are not allowed to produced at the same time on the same production stage. It can also be forbidden to produce a certain pair of products at the same time in the entire plant

$$\sum_{J} X_{i,w,j} + \sum_{J} X_{i',w,j} \leq 1 \quad \forall\, i \in I, i' \in IE_{i,i'}, w \in W \tag{13}$$

where $IE_{i,i'}$ is a set of orders that are not allowed to produced at the same time in the entire plant.

Objective Function

The objective function includes minimization of delayed deliveries of orders and the setup time needed for starting new campaigns

$$\min \alpha \cdot \sum_{i \in I} \varepsilon_i + \beta \cdot \sum_{j \in J} \sum_{w \in W} u_j \cdot Z_{w,j} \tag{14}$$

where α and β are parameters to adjust the weight of each criteria in the objective function, and u_j is a parameter for the average setup time for machine j.

Campaign Planning and Order Scheduling for Level 2

Here, we employ a similar model to that at level 1, but with actual individual customer orders and more accurate timing constraints.

Variables

Continuous variables

ε_i = continuous variable for delay of order i, equal to 0 if no delay will occur for order i else equal to the difference between estimated completion time of order i, and the due date for the order ($\varepsilon_i \geq 0$).

Binary variables

$X_{i,w,j}$ = decision variable equal to 1 if order i is processed in week w on machine j, else equal to 0.

$Z_{f,w,j}$ = decision variable equal to 1 if setup work is needed in week w for machine j, else equal to 0. A new dimension has been added to the variable compared to the formulation of the model at level 1.

$Y_{f,w,j}$ = decision variable equal to 1 if product family f is produced in week w on machine j, else equal to 0.

Constraints

For the constraints that are identical to those in the model used on level 1 we refer to the number of the corresponding equations in the previous section.

Allocation Constraints. The allocation constraint is identical to Eq. 1, stating that every order should be allocated to one single machine on each production stage $s \in S_i$. The difference is in the data used because here are orders from customers used instead of the demand forecast which is more aggregated.

Sequencing Constraints. Constraints 2 and 3 are used for ensuring the correct sequence of production stages, and the release time of orders. As an addition to the sequencing constraints used in the model on level 1 the production of an order should not end too early with respect to the delivery date for example, because of products' maximum lifetime.

$$(d_i + \varepsilon_i) - \sum_{j \in J_{i,s}} \sum_{w \in W_{ij}} X_{i,w,j} \cdot (w + p_{i,j} + q_{i,s}) \leq e_i \quad \forall\; i \in I, s = s_i^{last} \tag{15}$$

e_i is the maximum duration from the end of producing order i to the actual delivery calculated by $(d_i + \varepsilon_i)$.

Also as an addition to the model of level 1 the "flow time" of orders is limited by a specified maximum value ft_i.

$$\sum_{j \in J_{i,\max last_i}} \sum_{w \in W_{ij}} X_{i,w,j} \cdot w - \sum_{i \in J_{i,s=max first_i}} \sum_{w \in W_{ij}} X_{i,w,j} \cdot w \leq ft_i \quad \forall\ i \in I \quad (16)$$

Delivery Constraints. Since the model structure involves an overlapping moving horizon time framework, customer orders may either be "old" or "new", that is, their delivery time may have already been confirmed to the customer or not. When orders are first added to the production schedule it should be tried to respect the requested delivery date but if that is not possible a reply is given with a new delivery date to the customer. However, once the order delivery dates have been confirmed it is very important to respect them. The production of the order can be moved if the change results in an earlier or the same finishing time but not a later finishing time

$$\sum_{j \in J_{i,s}} \sum_{w \in W_{i,j}} X_{i,w,j} \cdot (w + p_{ij} + q_{i,s}) \geq d_i + \varepsilon_i \quad \forall\ i \in I_{\text{new}}, s = s_i^{\text{last}} \quad (17)$$

$$\sum_{j \in J_{i,s}} \sum_{w \in W_{i,j}} X_{i,w,j} \cdot (w + p_{ij} + q_{i,s}) \geq dT_i \quad \forall\ i \in I_{old}, s = s_i^{last} \quad (18)$$

I_{new} and I_{old} are, respectively, the sets of new orders and the set of orders that have already been scheduled but not yet been produced.

Capacity Constraints. In this model we increase the accuracy of the capacity constraints used earlier. Here we include constraints for limited capacity of each machine and limited capacity of factory, as whole and now we include the time used for work on setup

$$\sum_{i \in I_j} X_{i,w,j} \cdot p_{i,j} + \sum_{f \in F_j} Z_{f,w,j} \cdot u_{j,f} \leq pt_{j,w}^{\max} \quad \forall w \in W, j \in J_w \quad (19)$$

$pt_{j,w}^{\max}$ is the total time available for production in each week w on machine j

$$\sum_{j \in J_w} \sum_{i \in I_i} X_{i,w,j} \cdot p_{i,j} + \sum_{j \in J_w} \sum_{f \in F_j} Z_{f,w,j} \cdot u_{j,f} \leq pt_w^{\max} \quad \forall w \in W \quad (20)$$

$pt_w^{\max}$ is the total production time available in the factory each week.

In this model we also include a constraint for renewable resources such as number of employees available each week

$$\sum_{j \in J_w} \sum_{j \in I_{w,j}} X_{i,w,j} \cdot p_{i,j} \cdot ne_{i,j} + \sum_{j \in J_w} \sum_{f \in F_{w,j}} Z_{f,w,j} \cdot pu_{f,j} \cdot neu_{f,j} \leq tne_w \cdot wh_w \quad \forall\ w \in W \quad (21)$$

($ne_{i,j'}\ p_{i,j}$) is the number of working hours (number of employees times the production time) needed for producing order i on machine j, ($pu_{f,j} - neu_{f,j}$) is the number of working hours (number of employees times the production time) for setup of machine j for product family f, and ($tne_w \cdot wh_w$) is the total number of available working hours each week. This constraint can be broken up in order to allow for specific employees, that is, specific employees for setup operations (mechanics), and specific employees for production (machine operators).

Campaign Constraints. To activate the binary variable $Y_{f,w,j}$, indicating that product family f is produced on machine j in week w we use constraint (Eq. 7). The variable $Z_{f,w,j}$ has received an additional dimension as it now also defines the product family f for which a new campaign is intended. Constraint (Eq. 22) activates the binary variable $Z_{f,w,j}$ indicating that a new campaign of product family f is beginning and setup is needed in week w on machine j.

$$Z_{f,w,j} \geq Y_{f,w,j} - Y_{f,w-1,j} \quad \forall f \in F_j, j \in J_f, w \in W/w^{first} \quad (22)$$

It is often required that only one product family is produced in each week on each machine; this can be ensured with constraint (Eq. 9). To avoid empty campaigns we refer to Eq. 10.

A maximum and/or minimum number of campaigns on each production stage should be respected

$$Z_{s,f}^{\min} \leq \sum_{w} \sum_{j \in Js} Z_{f,w,j} \leq \mathrm{z}_{\mathrm{s,f}}^{\max} \quad \forall\ s \in S, f \in F \quad (23)$$

where $z_{s,f}^{\min}$ and $z_{s,f}^{\max}$ are respectively the minimum and maximum number of campaigns on stage s.

For the occasional instances when it is required that some certain orders are produced during the same time period we have constraint Eq. 24

$$\sum_{j \in Js} X_{i,w,j} = \sum_{j \in Js}, X_{i',w,j}\ \forall\ i \in ISC_{i,i'}, i' \in ISC_{i,i'}, s \in S_i, w \in W \quad (24)$$

Mutual exclusivity constraints

The mutual exclusivity constraints are the same as in the level one model apart from the orders used. Equation 12 prevents the production of certain pairs of orders at the same time on the same production level, and Equation 13 prevents production of certain pairs of orders at the same time in the entire plant.

Objective Function

The objective includes minimization of delays, and setup time for each product family f on each machine

$$\min \alpha \cdot \sum_{j \in I} \varepsilon_i + \beta \cdot \sum_{F} \sum_{w,j\, j \in J_w} \sum_{w \in W} u_{j,f} \cdot Z_{f,w,j} \quad (25)$$

α and β are parameters to adjust the weight of each criterion in the objective function, and $u_{j,f}$ is a parameter for the average setup time for family f on machine j.

Detailed Scheduling of Production Tasks for Level 3

Here, we utilize a continuous time model because we are interested in the actual timing of activities. Our formulation is based on a formulation proposed by Mendez, Henning and Cerda.[27]

Variables

Continuous variables

$ST_{i,s}$ = The start time for production of order i on stage $s \in S_i$ ($ST_{i,s} \geq 0$).

T_i = the difference between estimated completion time of production of order i, and the delivery date for order i ($T_i \geq 0$).

Binary variables

$X_{i,j}$ = binary variable = 1 if order i is processed on machine j, else = 0.

$Y_{i,i',s}$ = Binary variable = 1 if order i is processed before order i' on production stage s, else = 0, $s \in S_{i,i'}$.

$Q_{i,c}$ = Binary variable = 1 if order i is produced in campaign c, $c \in C_i$.

$W_{i,s,i',s'}$ = Binary variable = 1 if stage s' of order i' has been completed after starting stage s of order i.

Constraints

Allocation Constraints. Each order i should be allocated to exactly one machine j on each production stage s

$$\sum_{j \in J_{i,j}} X_{ij} = 1 \quad \forall\, i \in I, s \in S_i \tag{26}$$

Sequencing Constraints. Production of order i' on stage s cannot start before order i is finished or conversely, if both orders i and i' use the same machine j. It must also be ensured that there is sufficient time for setup work to be done in between orders

$$\left.\begin{aligned} &ST_{i,s} + p_{i,j} + u_{i,i',j} \leq ST_{i',s} + M\cdot(1 - Y_{i,i',s}) + M\cdot(2 - X_{i,j} - X_{i,j}) \\ &ST_{i',s} + p_{i',j} + u_{i',i,j} \leq ST_{i,s} + M\cdot Y_{i,i',s} + M\cdot(2 - X_{i,j} - X_{i,j}) \end{aligned}\right\} \quad \forall i, i' \in I, s \in S_{i,i'}, j \in J_{i,i',s} \tag{27}$$

These constraints are only active if order i and i' use the same machine (then the term $(2 - X_{i,j} - X_{i',j})$ becomes equal to zero). The first constraint is for the case when order i is scheduled earlier than order i' ($Y_{i,i'} = 1$) while the second constraint is for the case when order i is scheduled later than order i'.

The correct sequence of production stages for each order must be respected and it must be ensured that a production stage cannot start before its predecessor has finished

$$ST_{i,s} + \sum_{j \in J_{i,s}} X_{i,j}\cdot p_{i,j} + q_{i,s} \leq ST_{i,s+1} \quad \forall\, i \in I, s \in S_i \backslash s_i^{last} \tag{28}$$

Production cannot start earlier than the first possible start time of production on the first production stage, for example, because of availability of material, documentation or equipment.

$$r_i \leq ST_{is} \quad \forall\, i \in I, s = s_i^{first} \tag{29}$$

r_i is the earliest start time for production of order i.

The production of an order must not end too early with respect to delivery date for example, because of products' maximum lifetime.

$$\left(ST_{i,s} + \sum_{j \in J_{is}} X_{i,j}\cdot p_{i,j} + q_{i,s}\right) - \left(dT_i + \max\{0, T_i\}\right) \leq e_i \quad \forall\, i \in I, s = s_i^{last} \tag{30}$$

e_i is the maximum duration from the start of producing order i to the actual delivery calculated by $(d_i - \varepsilon_i)$. This constraint is rarely needed on this level as the time horizon is short and the orders will all be delivered within a relatively short period. The value of the variable T_i is calculated by Eq. 32.

The flow time of orders is limited with a specified maximum value ft_i

$$\left(ST_{i,s=s_i^{last}} + \sum_{j \in J_{i,s=s_i^{last}}} X_{ij}\cdot p_{ij} + q_{i,s=s_i^{last}}\right) - ST_{i,s=s^{first}} \leq ft_i \quad \forall i \in I \tag{31}$$

Delivery Constraints. A state variable T_i is used in the model for the difference between the planned completion time of production of order i, and the confirmed delivery date for order i. If the variable is positive then the confirmed delivery will be delayed

$$T_i = ST_{i,s} + \sum_{j \in J_{i,j}} X_{i,j}\cdot p_{i,j} + q_{i,s} - dT_i \quad \forall\, i \in I, s = s_i^{last} \tag{32}$$

$ST_{i,s}$ is the starting time of order i on stage s, $X_{i,j}$ is a binary variable that takes the value 1 if order i is allocated to machine j, $p_{i,j}$ is the production time of order i if machine j is used, $q_{i,s}$ is the time that needs to pass after each production stage and dT_i is the confirmed delivery date for order i.

Capacity Constraints. The limited capacity of the individual machines is already included in the sequencing constraints (Eq. 27) and (Eq. 28) where it is ensured that each task must finish before the next can start, and, thus, always at most one task can be processed on a particular machine at a particular time.

The number of employees is limited, and here it is assumed that production of order i on production stage s, needs a certain amount of employees, and this does not depend on the machine used for the production.

$$W_{i,s,i',s'}\cdot M > \left(ST_{i',s'} + \sum_{j \in J_{i',s'}} X_{i',j}\cdot p_{i',j}\right) - ST_{i,s} \quad \forall\, i, i' \in I, i' \neq i, s \in S_i, s \in S_{i'} \tag{33}$$

$$ne_{i,s} + \sum_{i' \in I, i \neq i'} \sum_{s' \in S_{i'}} ne_{i',s'}(W_{i,s,i',s'} + W_{i',s',i,s} - 1) \leq tne \quad \forall\, i \in I, s \in S_i \tag{34}$$

Constraint (33) activates the binary variable $W_{i,s,i',s'}$, which takes a value of 1 if stage s' of order i' is completed after starting stage s of order i. For each order i and stage s, constraint (34) sums the number of employees used simultaneously to produce all other orders i' at the corresponding stages s', and adds to $ne_{i,s}$ which is the number of employees needed for producing order i on stage s. On the right hand side is tne, which is the total number of available employees and the constraint, thereby ensures that the available number of employees is respected. Orders i and i' are simultaneously processed to some extent only if $W_{i,s,i',s'} + W_{i',s',i,s} = 2$, and $W_{i,s,i',s'} + W_{i',s',i,s}$ is never less than 1.

Campaign Constraints. The campaigns generated at level 2 are used as input for this model. Constraints (35) to (37) ensure that order i is produced in an appropriate campaign c on each production stage s. It is assumed that a setup can be started and finished before the start of the campaign if the machine is not being used for some other production task. If there is insufficient time between campaigns to do the required cleaning and other setup activities on the machine, then the sequencing constraints guarantee that the setup can be done

$$Q_{ic} \cdot sc_c \leq ST_{is} \quad \forall\, i \in I, c \in C_{is}, s \in S_i \tag{35}$$

$$ST_{is} + \sum_{j \in J_{is}} X_{ij} \cdot p_{ij} \leq ec_c + M \cdot (1 - Q_{ic}) \;\; \forall i, c \in C_{is}, s \in S_i \tag{36}$$

$$\sum_{C_{is}} Q_{ic} = 1 \quad \forall\, i, s \in S_i \tag{37}$$

$Q_{i,c}$ is a binary variable that takes the value of 1 if order i is allocated in campaign c, sc_c and ec_c are respectively the starting and finishing time for campaign c.

The campaign constraints described here can in general be considered as optional. If the campaign constraints are disregarded the model will in fact also decide the structure of the campaigns.

Mutual Exclusivity Constraints. As mentioned earlier, the campaign plan used in this model is created by the model on level 2 in the hierarchical framework. When the campaign plan is created it is ensured that pair of orders (or campaigns) with certain properties will not be produced at the same time on the same production level or even not at the same time in the entire plant and hence mutual exclusivity constraints are not needed here.

Objective Function

The objective used in this model is to minimize late deliveries, that is, the priority weighted difference between completion time of orders and the confirmed delivery date of orders

$$\text{Min} \sum_{i=1}^{n} pr_i \max\left\{0, T_i\right\} \tag{38}$$

where pr_i is a priority weight of orders. If the production of the order is completed later than the confirmed delivery date the value of T_i is positive, however, if there is no delay it will be negative and the term max(0, T_i) will give the value zero.

Solution Approaches

Planning and scheduling problems, in particular scheduling problems, are known to be very difficult to model and solve in an efficient manner.[2] Our objective is to provide realistic and accurate models solvable within acceptable computational times. This is very difficult in practice for complex and comprehensive real-world problems such as the one we are working with.

Decomposition Algorithm

Problem dynamics

When decomposition methods are applied the problem is broken into smaller parts that are solved separately although possible interactions between the subproblems should be taken into account. The campaign planning problems found in the literature are in some instances decomposed into two parts; the first one deciding the length and number of campaigns (campaign sizing), and the second one assigning machines to the campaigns and deciding the sequence of the campaigns on the assigned machines. This approach has limitations which may cause infeasible solutions or compromise the optimality of the solution obtained. The main limitation with this kind of decomposition is that the two different problems (or even three) are highly connected, and the divided decisions highly influence each other. It decreases the quality of the solution to solve the campaign sizing problem without assigning products to machines as the machines have different capacities and availability. It is also not promising to solve the assignment problem without sequencing, because of changeovers or to solve the sequencing without the campaign sizing because there are complex temporal relationships between campaigns, especially in multistage production facilities. It is, therefore, desirable to solve the campaign planning problem without this kind of decomposition, that is, solving first the problem of deciding the length and the number of campaigns and then the problem of assigning machines to the campaigns and deciding the sequence of the campaigns on the assigned machines. In the case under consideration in this research the capacity is highly restricted. Due to the high demand the factory has to utilize its capacity as well as possible, and the utilization is more important than keeping the inventory levels of intermediate or finished products as low as possible, as is often considered important in the literature. This fact makes it even more desirable to solve simultaneously the different parts of the campaign planning problem.

Decomposition of Production Stages

After standard solution methods failed to provide solutions to the problem at level 1 and 2, we developed and implemented a decomposition heuristic. Problem sizes are given in Table 1.

The decomposition heuristic is inspired from the production performance problems found in the plant we are working with. When we started analyzing production data and schedules from the plant, it was obvious that there were two bottlenecks in the overall production process, at the first production stage (granulation) and at the last production stage (packing). The capacity on those stages was barely sufficient for the scheduled production and the schedule was, therefore, volatile due to unseen events. The capacity on the two intermediate production stages did not restrict the overall production capacity.

Because of the limited capacity at the first and the last production stage we decomposed the problem into two main components:

1. First, solve the problem for stage 1 and stage 4
2. Then, solve the problem for stage 2 and stage 3

The underlying assumption is that the production stages causing the bottlenecks were most important to plan in an optimal way and should, therefore, have higher-priority in the planning process, while the production stages with sufficient or superfluous capacity should have lower-priority, and be planned with regard to the requirements from the others. This follows the ideas of the "theory of constraints" (see, for example,

Table 1. Maximum Number of Variables, Continuous Variables, Integer Variables and Constraints for the Largest Test Cases Experienced with Standard Solution Methods at Each Level of the Hierarchical Framework

Level	Total Variables	Cont. Variables	Int. Variables	Constraints
1	51738	9894	41844	47468
2	39524	7798	31726	44478
3	9117	300	8817	22258

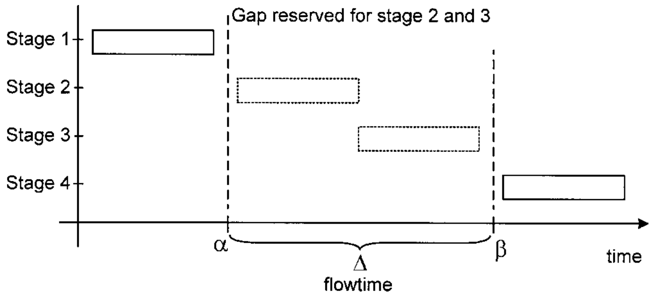

Figure 10. Illustration of the intended gap between production of stages 1 and 4 where production on stages 2 and 3 is fitted in.

Luebbe and Finch[29] for a comparison between this and optimization).

Structure and Implementation

When decomposition methods are applied and the problem components solved separately, it is important to take possible interactions among them into account. In order to make the production schedule for production stages 1 and 4 feasible as input for solving the problem for production stages 2 and 3, we defined an "expected flow time" for stages 2 and 3. The flow time is the expected time required for setup, production, waiting and all other tasks required from finishing production on stage 1 until starting production on stage 4. By including the flow time while solving for production on stage 1 and 4, the model should be able to account for the intermediate production tasks. When the problem is solved for stages 2 and 3, the finishing times from stage 1 and starting times from stage 4 create a gap or time interval defining when the production is allowed to take place as shown in Figure 10.

The flow time for stage 2 and 3 was estimated from the production data as the minimum time period possible for the production of each order. The flow time depends on the products being produced as well as on the machine selected for the production because some of the products do not need coating on stage 3 and the machine capacity depends on the product. The actual machine for each order will not have been selected when the flow time is estimated, and, thus, an average value must be used. The flow time also depends on the quantity specified by customer orders. However, if the machines on stage 2 and 3 are heavily loaded it may not be feasible to let the production go directly through the stages without waiting and being finished in the minimum possible time. In such cases the flow time must be readjusted according to the results from infeasible production on stage 2 and 3, and the problem solved again for stage 1 and 4. This is thus an iterative process which should not be terminated unless the flow times have been correctly adjusted and the production schedule becomes feasible for all production stages. After that has been achieved the order allocation and sequence is solved simultaneously for all production stages with the campaign structure fixed according to the results obtained from solving the previous models. If the flow time becomes greater than the planning horizon then there is either no feasible solution or the algorithm is off track and should be terminated. The algorithm can be summarized in the following steps.

Step 1

Initialize values for flow time and solve campaign planning and scheduling problem for production stage 1 and 4.

Step 2

Calculate earliest possible starting and latest possible finishing times for production on stage 2 and 3 from the results obtained in step 1.

Step 3

Initialize values for starting and finishing times and solve campaign planning and scheduling problem for production stage 2 and 3.

Step 4

If the result from step 3 is infeasible increase flow time and repeat step 1 and 2, else go to step 5. If the flow time for any of the orders is larger than the planning horizon then stop the algorithm.

Step 5

Solve the campaign planning and scheduling problem for all stages with fixed campaign structure from step 1 and 2.

The algorithm is illustrated with a flowchart shown in Figure 11.

When the decomposition algorithm is used the number of variables and constraints is greatly reduced in the individual models and thereby large savings made regarding computational effort when solving the individual models. There are however some evident drawbacks of the decomposition algorithm. The major drawback is involved in the integration between the two main components of the algorithm. If the gap provided for production of order i on stages 2 and 3 is not large enough for obtaining feasible solutions then the algorithm increases the flow time for the order and makes iteration as shown in Figure 11. This should in principle be an effective method for guaranteeing a feasible solution for each order i, however, if and only if it is the only order to be produced on certain machine. When the flow time is increased for one order it can have an effect on all other orders that share the same machines on one or more of the production stages. If the gap is increased for one order it may open some new sequencing options for other orders that may already have an increased gap, and, thus, the overall flow time for those orders can not be decreased back to the original value. The algorithm can, thus, make the gap for orders unnecessarily large and thus obtain solutions far from optimal.

Lower bounds improved with valid inequalities

When the models were tested for the real world case under consideration it became evident that they suffered from a poor LP relaxation, and, therefore, an increased solution time. The same occurrence has been reported in some other research available in the literature on resource constrained scheduling problems and in some cases this has made them very difficult

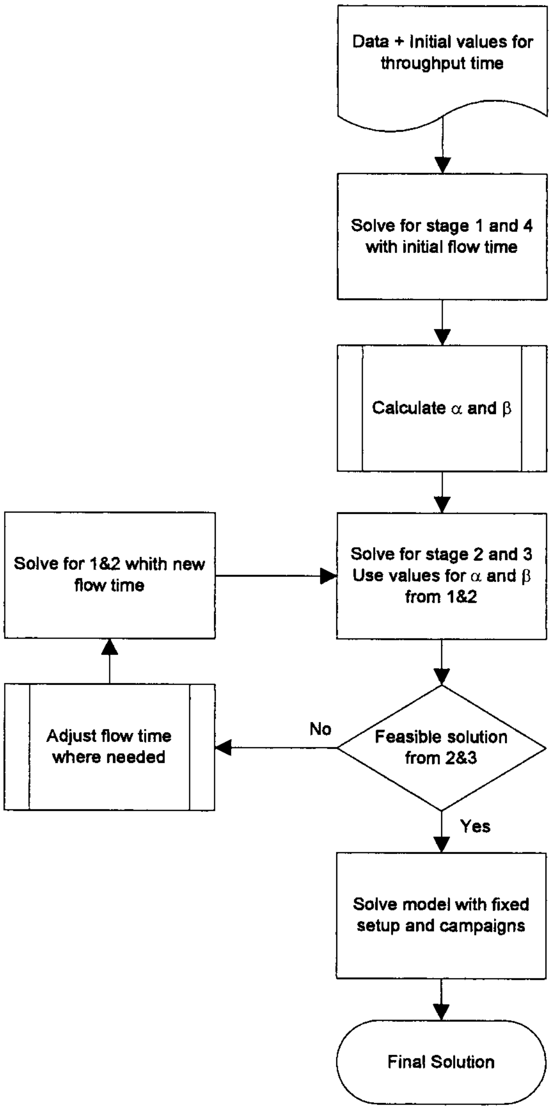

Figure 11. Flowchart illustrating the structure of the decomposition algorithm.

or even impossible to solve with MILP methods.[2] The reason for this poor relaxation is mainly because resource constraints can easily be fulfilled with fractional values of the binary variables used in the discrete time formulations, and as a result the lower bounds become very weak.

To improve the lower bounds two kinds of valid inequalities where added to the models as described in the next sections.

Minimum Number of Campaigns Required

The most effective method to improve the lower bounds for the models at level 1 and 2 is to add a constraint for the minimum number of campaigns needed for feasible solutions. The minimization of setup cost is one of the criteria in the objective function as can be seen in Eqs. 14 and 25. Introducing a minimum value for the number of campaigns, that is, the number of times a setup is needed, directly creates a lower bound for the objective function and improves the LP relaxation. The constraint specifies the minimum number of campaigns needed for each product family on each production stage over the entire planning horizon

$$z_{s,f}^{\min} \leq \sum_{w} \sum_{j \in J_s} Z_{f,w,j} \quad \forall\, s \in S, f \in F \tag{39}$$

where $z_{s,f}^{min}$ is the minimum number of campaigns for product family f on stage s. The minimum number of campaigns depends on the production data for the case under consideration but it is not always easy to estimate it. Two methods were considered for the estimation of the minimum number of campaigns.

Simple data analysis to estimate minimum number of campaigns

Data analysis was conducted to estimate the minimum number of campaigns required for fulfilling the product demand defined by customer orders. The data analysis routine counts the number of product families included in the order domain. If the feasible production intervals for all the orders of the same product family intersect then the data analysis routine concludes that one campaign on each production stage is sufficient, otherwise more than one campaign is needed. The data analysis routine also uses the maximum amount of product that can be produced in each campaign to decide the minimum number of campaigns.

Heuristic method to estimate minimum number of campaigns

A simple heuristic was also developed for the estimation of the minimum number of campaigns required. The heuristic is based on the following steps.

Step 1

Calculate $\hat{D}_f = \frac{\sum_{j \in I_f} D_i \cdot M_i}{\sum_{j \in I_f} M_i}$ as the mass weighted due date for each family f and T_{fs}^{P} as the production time (duration) for all orders in each family f on the first production stage. M_i is the amount in each order.

Step 2

Select the family with the lowest $\hat{D}_f$ value as the one to be produced first. The start time of that family will be $T_{fs} = 0$.

Step 3

Select the family with the lowest $\hat{D}_f$ value from the families that have not already been selected as the one to be produced next. The start time for that family will be $T_{fs} = T_{fs}^{last} + T_{fs}^{p}$ where T_{fs}^{last} is the starting time of the last family that was selected before this one (the predecessor on the same machine).

Step 4

Repeat step 3 until all families on first production stage have been located.

Step 5

Locate campaigns on stage 2–4. Calculate the start time for the family with the lowest $\hat{D}_f$ value as $T_{fs} = T_{f(s-1)} + T_{f(s-1)}^{p}$ and for other families selected in a sequence corresponding to the $\hat{D}_f$ value T_{fs} is the larger of $T_{fs} = T_{f(s-1)} + T^p_{f(s-1)}$ or $T_{fs} = T^{last}_{fs} + T^p_{fs}$. Also calculate the end time T^{finish}_f for the last stage of each family.

Step 6

Calculate the difference between the end time and the mass weighted due date of family f $T^\Delta_f = T^{finish}_f - \hat{D}_f$

Step 7

If $T^\Delta_f > 0$ then the production of family f needs to begin earlier. If the campaign of the predecessor is larger than the minimum time length then split that campaign into two halves on each stage and put the orders with the earliest due dates in the first half. Check this for all families and split if necessary.

Step 8

If $T^\Delta_f < 0$ for all families in step 7 then this algorithm has finished, otherwise proceed to next step.

Step 9

Repeat steps 1–5 to relocate all the campaigns.

Step 10

Repeat step 6–8 to check if there are delays.

Step 11

If $T^\Delta_f < 0$ for all families in step 7 or all the campaigns have been broken into minimal sizes then this algorithm has finished.

This algorithm is also illustrated in the flowchart shown in Figure 12.

The heuristic is, however, not guaranteed to provide strict lower bounds for the minimum number of campaigns. The heuristic can result in too many campaigns as its only objective is to avoid delays and it does not consider the cost of setups. Since the setup cost is also part of the objective function this heuristic can in some cases give a higher number of campaigns than the actual optimization model will deliver.

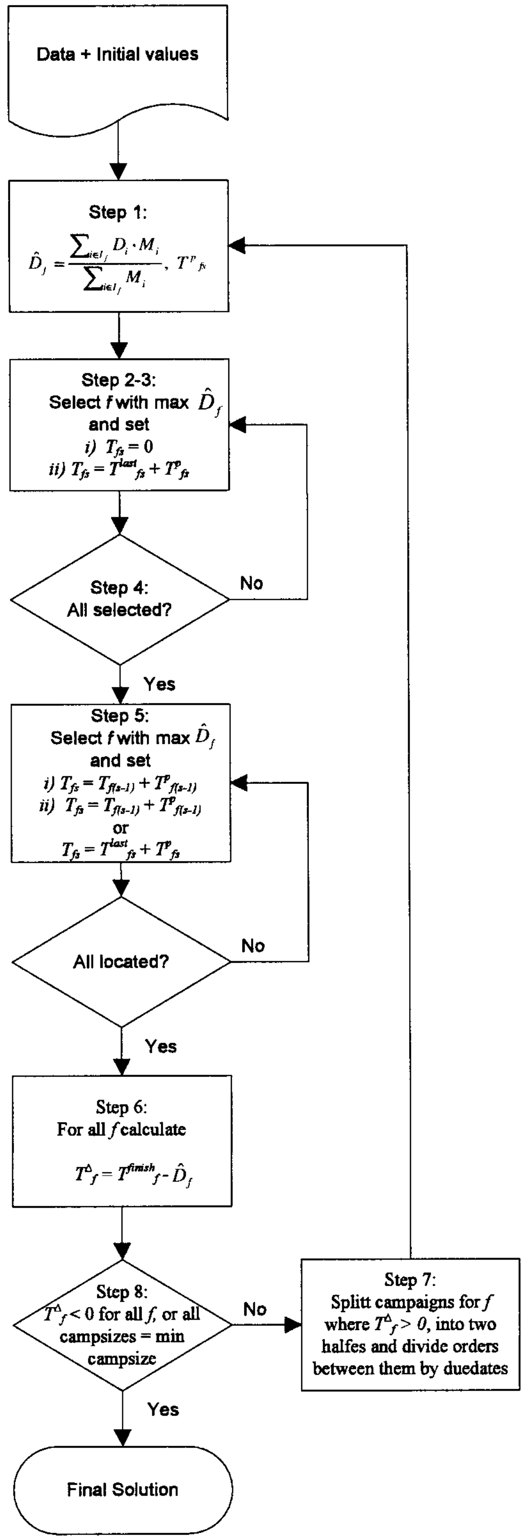

Figure 12. A flowchart illustrating the structure of the heuristic for estimating minimum number of campaigns.

Minimum Unavoidable Delays

The lower bounds for the models on level 1 and 2 in the hierarchical framework can also be improved by adding constraints for the minimum delays that will occur in feasible solutions. Minimization of delays is one of the criteria in the objective functions seen in Eqs. 14 and 25. Introducing a minimum value for the delays creates a lower bound for the objective function and improves the LP relaxation.

To obtain the value for the minimum unavoidable delays a simple algorithm is used based on first solving a relaxed model where constraint (9) is relaxed. The constraint ensures that only one campaign can be produced on each machine in each week. From the output of the relaxed model we obtained the value for the minimum delays. This constraint was relaxed because it makes the solution of the model considerably more difficult for the test cases, as there are some orders constituting of low-quantities of products that do not share product families with any of the other orders (products). Therefore, a campaign needs to be constructed entirely for each of these orders and a machine reserved for the whole week as one week is the minimum time unit used. This reduced the utilization of the machines and resulted in more delays and thus by

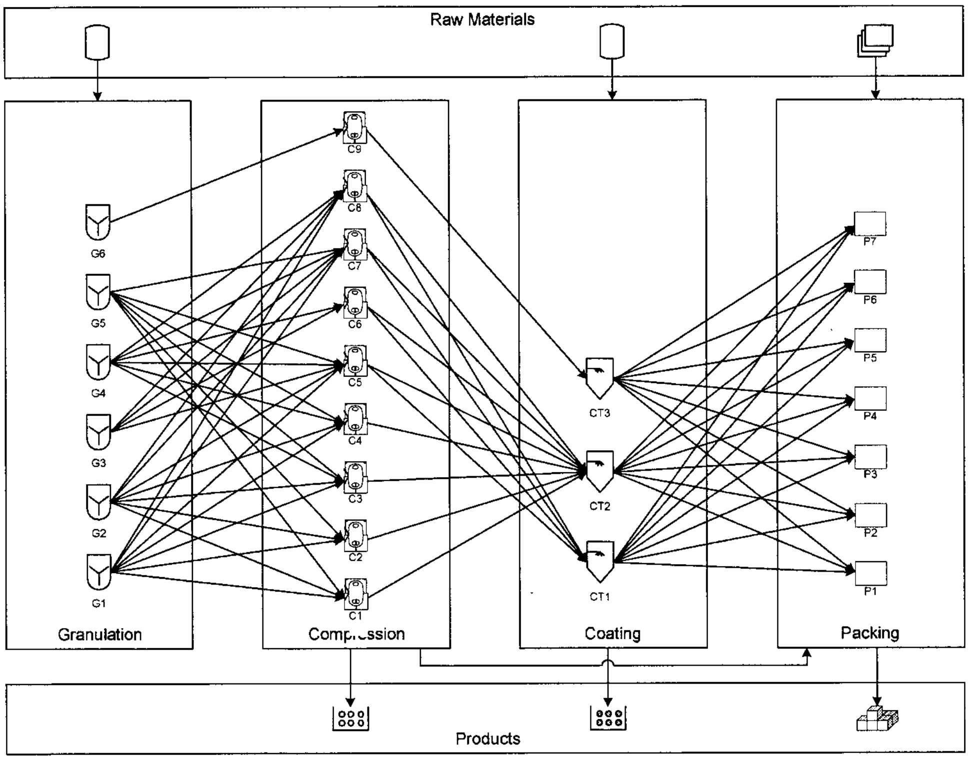

Figure 13. Flowsheet illustrating the production process under consideration.

relaxing the constraint we were able to obtain a valid lower bound for the delays of the complete model.

A simple constraint was added to the complete model to implement the lower bound for the minimum delays

$$\varepsilon^{\min} \leq \sum_{i} \varepsilon_i \tag{40}$$

where $\varepsilon^{\min}$ is the total delays obtained from the relaxed model.

The algorithm can be described with the following simple steps

Step 1

Solve the model with constraint (9) relaxed

Step 2

Read results and use as input for next step

Step 3

Solve the complete model with the additional constraint for the minimum delays

Results

Test cases

The optimization models and solution algorithms have been implemented and tested with real world data from the problem under consideration. The production process is illustrated with a flowsheet shown in Figure 13.

Several fullscale test cases were created from data collected in the pharmaceutical production plant which reflect the decision problems at each level in the hierarchical framework (see main dimensions of test cases in Table 2).

Results

The results obtained indicate that the models and solution procedures for the hierarchical framework are capable of obtaining solutions of good quality within an acceptable computational time for very difficult *real world* problems. The model for the detailed scheduling at the lowest level was

Table 2. Table Shows the Main Dimensions of the Test Cases.

Test Case no.	Horizon [months]	No. of New Orders	No. of Confirmed Orders	Total No. of Orders	No. of Machines	No. of Products / P. Families
1	1	0	25	25	16	24/4
2	1	3	47	50	19	45/9
3	1	3	72	75	23	66/12
4	3	6	94	100	24	86/14
5	3	17	133	150	26	130/19
6	3	32	168	200	27	160/25
7	6	32	193	225	27	177/26

The delivery dates of the orders are within the specified horizon. New orders are orders that are new in the system and have not already been scheduled and their delivery dates have not been confirmed yet. Confirmed orders have been scheduled before and their delivery dates confirmed; they are rescheduled if needed.

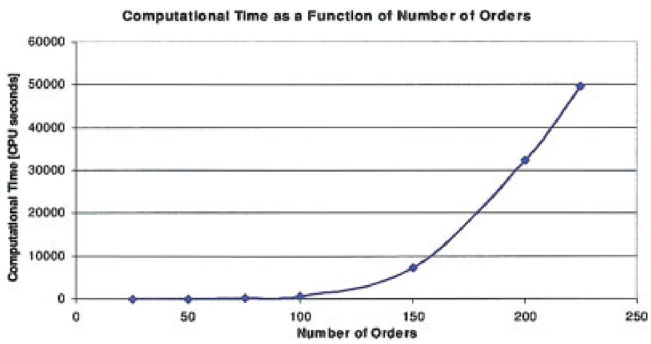

Figure 14. Computational time as a function of number of orders.

[Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

solved for all the test cases it was designed for within 6 min of computational time. The model will sometimes be used more than once per day and should, therefore, be very fast. The model for the simultaneous campaign planning and order scheduling at the middle level was solved for all its target test cases within 9 h of computational time which is an acceptable solution time as it is usually only used once per week. The model for the campaign planning at the top level in the hierarchical framework was solved for its target test case within 14 h of computational time which is acceptable as the long-term campaign planning is usually only done every 3 months. However, it must be mentioned that the target test case for the top level is not sufficiently large to fully reflect the challenge in the real world problem. The computational time as a function of number of orders is shown in Figure 14 and more detailed solution statistics are given in Table3.

In order for the optimization models and solution procedures to be practical they must offer quality results in acceptable solution times. The definition of acceptable solution time depends on the intended use of the models. For long term campaign planning at level 1 of the hierarchical framework it is fully acceptable to wait over one night, that is, for approximately 12 h. For the campaign planning and order scheduling model at level 2 it is also acceptable to wait over night as the customer orders are never confirmed on the same day as they are submitted to the factory. However, for the purpose of demand management and to answer CTP questions required by customer service (capable to promise), it is preferable to obtain some solution in one hour or less although that is very difficult. The customer service function can, however, also use visual Gantt charts representing the planned campaigns and unallocated amounts of each material available in each campaign, to get approximated answers to these kind of questions. Approximate answers can also be provided with acceptable accuracy by the use of simple heuristics or data analysis methods. For the detailed scheduling at level 3 of the hierarchical framework it is acceptable to wait for 1 h or less (for example, during lunch) as the schedule must often be reconsidered once or even more often per day when new information becomes available.

The hierarchical framework is well suitable to be embedded in decision support systems for planning and scheduling. The efficiency of the models and solution procedures used in the framework is really important for real world applications

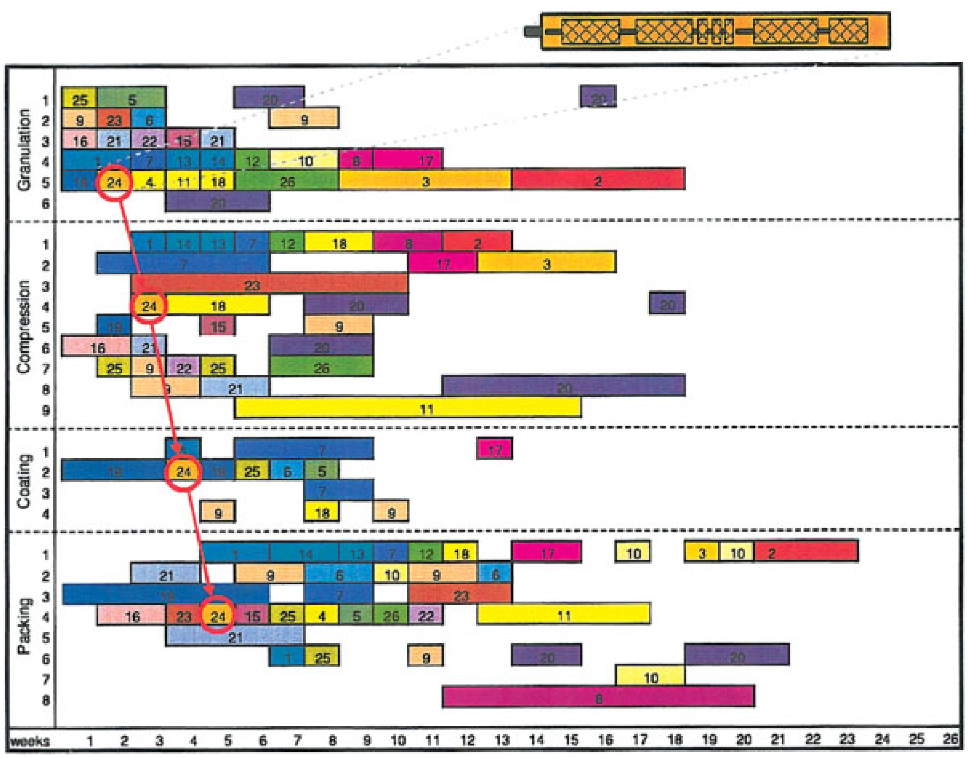

Figure 15. An example of a solution illustrated with a Gantt chart.

Each bar represents one campaign and the shadings of the bars represent a certain product group. The circles show an example of a path of one order through the production stages. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

Table 3. Statistics Representing Results from Solving Test Cases with the Proposed Models

Test Case no.	Level	Number of Orders	LP Solution	Objective Value	Solution Time [CPU seconds]	Gap [%]
1	3	25	17.11	17.11	10.39	0
2	3	50	17.11	17.11	64.04	0
3	3	75	17.11	17.11	311.71	0
4	2	100	13.15	14.05	643.10	6.4
5	2	150	20.17	20.39	7297	1.1
6	2	200	28.91	*29.51*	32364	2.0
7	1	225	29.87	32.84	49503	9.0

The models were solved with CPLEX 9.0 running on Unix based machine with 2GB of RAM and 2.0 GHz Pentium IV CPU.

where new plans and schedules are often needed quickly as new information become available. The recent research found in the literature on planning and scheduling in the process industry focuses on solution efficiency and techniques to render ever larger problems tractable. There remains work to be done on both model enhancements and improvements in solution algorithms if industrially relevant problems are to be tackled routinely, and software based on these are to be used on a regular basis by practitioners in the field. The quality of solutions is also vital factor for the acceptance of practitioners in the field. The results obtained with the optimization models have been presented for the production management division in the factory that provided the test cases. Their impression was that the results are more or less in accordance with the operational environment and the results respect the operational constraints required. The production management division also confirmed that the results seemed sensible in terms of optimality of the plans. It can, thus, be concluded that the results are practical in terms of quality.

Figure 15 gives an example of a solution obtained with the optimization models and solution algorithms that we have described. The figure only presents the campaign plan for the solution, and it does not show allocation or the exact schedule of the orders within each of the campaigns which is a very complicated part of the problem and adds to the complexity of the problem.

Conclusions and Future Work

A large amount of research has been accomplished in planning and scheduling for the process industry. However, a fixed time horizon is usually considered and it is assumed that all data is given. We believe that multiscale dynamic online procedures are more suitable for many specific types of planning and scheduling problems found in the process industry and should be further explored.

This research examines production planning and scheduling for a real world problem originating from a pharmaceutical manufacturer in a very competitive business dealing with a continuous and dynamic decision process where decisions have to be made before all data are available. We have proposed a hierarchically structured framework for tackling the decision process and efficient optimization models and solution approaches to solve the relevant decision problems at each level. The optimization models and solution approaches have been validated and tested with several actual test cases from the real world problem under consideration. The results obtained so far indicate that the models and solution procedures are capable of obtaining solutions of good quality within an acceptable computational time for very difficult real world problems.

A significant amount of work remains on improving the integration and flow of information between the levels in the hierarchical framework and different strategies need to be evaluated with the objective to capture the imperatives and respecting the dynamics of the decisions problem. It is also of interest to evaluate and perhaps introduce statistical uncertainty in the models as there are many different sources of uncertainties that affect the problem. We also aim to collect data over a longer period, for example, a period of one year and both collect the input data as well as output data, such as the production plans and the final result after random events have affected the plans. This type of dataset is fundamental in order to evaluate the potential benefits of hierarchical framework in its genuine operational environment characterized by the continuous rolling horizon and the dynamic decision process.

Acknowledgments

The authors would like to acknowledge support received from The Research Council of Iceland (RANNIS).

Notation

Indices

i = orders
i' = orders
j = machines
s = production stages
c = campaigns
f = product families/groups
w = weeks (the number of each time slot)

Sets

I = set of orders, $i \in I$ and $i' \in I$
J = set of machines, $j \in J$
J_s = set of machines on production stage s, $J_s \subseteq J$
J_i = set of machines that can process order i, $J_i \subseteq J$
$J_{i,s}$ = set of machines on production stage s that can process order i, $J_{i,s} \subseteq J_s$
$J_{i,i',s}$ = set of machines on production stage s that can process both order i and i', $J_{i,i',s} = J_{i,s} \cap J_{i',s}$
J_w = set of machines that can be used in week w, $J_w \subseteq J$
S = set of stages, $s \in S$
S_i = set of stages required for production of order i, $S_i \subseteq S$
$S_{i,i'}$ = set of stages that are common for order i and i', $S_{i,i'} = S_i \cap S_{i,s}$
C = set of campaigns, $c \in C$
C_i = set of campaigns suitable for production of order i, $C_i \subseteq C$
$C_{i,s}$ = set of campaigns suitable for production of order i on stage s, $C_{i,s} \subseteq C$
W = set of weeks, $w \in W$
$W_{i,j}$ = set of weeks possible for production of order i on machine j, $W_{i,j} \subseteq W$

F = set of product families, $f \in F$
I_f = set of orders i that belong to family f, $I_f \subseteq I$
I_f = set of orders i that can be produced with machine j, $I_j \subseteq I$
I_{old} = set of orders i that have already been placed on plan, $I_{old} \subseteq I$
I_{new} = set of orders i that have not been placed on plan earlier, $I_{new} \subseteq I$
$I_{f,w,j}$ = set of orders i in family f that can be produced in week w on machine j, $I_{f,w,j} \subseteq I$
$F_{w,j}$ = set of families that can be produced on machine j in week w, $F_{w,j} \subseteq F$
$IS_{i,i'}$ = set of orders that are not allowed to be produced at the same time at the same production stage in the plant
$IE_{i,i'}$ = set of orders that are not allowed to produced at the same time in the entire plant
$ISC_{i,i'}$ = set of orders that are supposed to be produced in the same campaign

Parameters

s_i^{first} = first production stage for order i
s_i^{last} = last production stage for order i
d_i = requested delivery date for order i, ($i \in I_{new}$)
dT_i = confirmed delivery date for orders that have already been scheduled on the planning horizon under consideration ($i \in I_{old}$)
$p_{i,j}$ = processing time of order i when machine j is used
$q_{i,s}$ = the time for order i that needs to pass on stage s before it is possible to start to work with it on next stage $s + I$ or to deliver the order to the customer
u_j = average setup cost for machine j
$u_{j,f}$ = average setup time for machine j and product family f
$u_{i,i',j}$ = sequence-dependent setup time between orders i and i'
sc_c = start of production slot for campaign c
ec_c = end of production slot for campaign c
pr_i = priority weight of order i ranging from minimum priority 1 to maximum priority 10
$ne_{i,s}$ = average number of employees needed for processing order i on production stage s
$ne_{i,j}$ = number of employees needed for processing order i on machine j
$neu_{f,j}$ = number of employees needed for setup of machine j for product family f
tne = average total number of available employees
tne_w = total number of available employees each time period
wh_w = total number of available working hours from employees in each time period
r_i = earliest start time for production of order I
lc_i = last possible completion time for production of order i
e_i = Maximum time duration from the finish of production of order i to the delivery date of the order
ft_i = maximum flow time of order i (time from start to finish of production of order)
$pt_{j,w}^{\max}$ = maximum capacity for machine j in each week w
$pt_w^{\max}$ = maximum capacity for factory in each week w
$z_s^{\min}$ = minimum number of setups on stage s
$z_{s,f}^{\min}$ = minimum number of setups on stage s for product family f
$z_s^{\max}$ = maximum number of setups on stage s
$z_{s,f}^{\max}$ = maximum number of setups on stage s for product family f
M = very large number

Literature Cited

1. Floudas CA, Lin XX. Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. *Comp & Chem Eng*. 2004;28(11):2109–2129.
2. Kallrath J. Planning and scheduling in the process industry. *Or Spectrum*. 2002;24(3):219–250.
3. Pinto JM, Grossmann IE. Assignment and sequencing models for the scheduling of process systems. *Annals of Operations Research*, 1998. 81: p. 433–466.
4. Shah N. Single- and multisite planning and scheduling: Current status and future challenges. *AIChE Symposium Series*. 1999;320(94):75–90.
5. Shah N. Process industry supply chains: Advances and challenges. *Comp & Chem Eng*. 2005;29(6):1225–1236.
6. Grunow M, Gunther HO, Lehmann M. Campaign planning for multistage batch processes in the chemical industry. *Or Spectrum*. 2002; 24(3):281–314.
7. Papageorgiou LG, Pantelides CC. A hierarchical approach for campaign planning of multipurpose batch plants. *Comp & Chem Eng*. 1993; 17(Supplement 1):27–32.
8. Papageorgiou LG, Pantelides CC. Optimal campaign planning scheduling of multipurpose batch semicontinuous plants. 1. Mathematical formulation. *Ind & Eng Chem Res*. 1996;35(2):488–509.
9. Shah N, Pantelides CC. Optimal Long-Term Campaign Planning and Design of Batch-Operations. *Ind & Eng Chem Res*. 1991;30(10): 2308–2321.
10. Tsirukis AG, Papageorgaki S, Reklaitis GV. Scheduling of Multipurpose Batch Chemical-Plants with Resource Constraints. *Ind & Eng Chem Res*. 1993;32(12):3037–3050.
11. Christofides N, Alvarezvaldes R, Tamarit JM, Project Scheduling with Resource Constraints - a Branch and Bound Approach. *European J of Operational Res*. 1987;29(3):262–273.
12. Dematta R, Guignard M. Dynamic Production Scheduling for a Process Industry. *Operations Research*. 1994;42(3):492–503.
13. Harjunkoski I, Grossmann IE. A decomposition approach for the scheduling of a steel plant production. *Comp & Chem Eng*. 2001; 25(11–12):1647–1660.
14. Hui C-W, Gupta A. A novel MILP formulation for short-term scheduling of multistage multi-product batch plants. *Comp & Chem Eng*. 2000; 24(2–7):1611–1617.
15. Jain V, Grossman IE. Decomposition techniques for multistage scheduling using mixed-integer and constraint programming. *INFORMS J on Computing*. 2001;13(4).
16. Kondili E, Pantelides CC, Sargent RWH. A General Algorithm for Short-Term Scheduling of Batch-Operations. 1. Milp Formulation. *Comp & Chem Eng*. 1993;17(2):211–227.
17. Mendez CA, Cerda J. Optimal scheduling of a resource-constrained multiproduct batch plant supplying intermediates to nearby end-product facilities. *Comp & Chem Eng*. 2000;24(2–7):369–376.
18. Neumann K, Schwindt C. Project scheduling with inventory constraints. *Math Methods of Operations Res*. 2003;56(3):513–533.
19. Pinto JM, Grossmann IE. Optimal Cyclic Scheduling of Multistage Continuous Multiproduct Plants. *Comp & Chem Eng*. 1994;18(9): 797–816.
20. Schwindt C, Trautmann N. Batch scheduling in process industries: an application of resource-constrained project scheduling. *Or Spektrum*. 2000;22(4):501–524.
21. Sgall J. On-line scheduling, in Online Algorithms, A. Fiat and G. Woeginger, Ed. Springer Verlag Berlin. Berlin; 1998;196–231.
22. Sand G, Engell S. Modeling and solving real-time scheduling problems by stochastic integer programming. *Comp & Chem Eng*. 2004; 28(6–7):1087–1103.
23. Sand G, Engell S, Markett A, Schultz R. Approximation of an ideal online scheduler for a multiproduct batch plant. *Comp & Chem Eng*. 2000;24(2–7):361–367.
24. Engell S, Markett A, J and G, Schultz R. Online scheduling of multiproduct batch plants under uncertainty, in Online Optimization of Large Scale Systems, M. Grötschel, S.O. Krumke, and J. Rambau, ed. Springer Verlag Berlin. Berlin; 2001;649–676.
25. Mendez CA, Cerda J. Dynamic scheduling in multiproduct batch plants. *Comp & Chem Eng*. 2003;27(8–9):1247–1259.
26. Mendez CA, Cerda J. An MILP framework for batch reactive scheduling with limited discrete resources. *Comp & Chem Eng*. 2004;28(6–7): 1059–1068.
27. Mendez CA, Henning GP, Cerda J. An MILP continuous-time approach to short-term scheduling of resource-constrained multistage flowshop batch facilities. *Comp & Chem Eng*. 2001;25(4–6):701–711.
28. Shah N. Pharmaceutical supply chains: key issues and strategies for optimisation. *Comp & Chem Eng*. 2004;28(6–7):929–941.
29. Luebbe R, Finch B. Theory of constraints and linear programming: a comparison. *Int J of Production Rese*. 1992;30(6):1471–1478.

Manuscript received Aug. 16, 2005, and revision received Jun. 30, 2006.