UNIFIED FRAMEWORKS FOR OPTIMAL PRODUCTION PLANNING AND SCHEDULING: CONTINUOUS-TIME DECOMPOSITION-BASED APPROACHES

by

DAN WU

A Dissertation submitted to the

Graduate School-New Brunswick

Rutgers, the State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Chemical and Biochemical Engineering

written under the direction of

Dr. Marianthi G. Ierapetritou

and approved by

____

____

____

____

New Brunswick, New Jersey

October, 2005

ABSTRACT OF THE DISSERTATION

Unified Frameworks For Optimal Production Planning And Scheduling: Continuous-Time Decomposition-Based Approaches

by DAN WU

Dissertation Director:

Dr. Marianthi G. Ierapetritou

Efficient plant operations involve making optimal decisions in different short-term scheduling and production planning. Although performed separately, it is evident that huge savings can be achieved from the integration of the planning and scheduling. In this dissertation a number of approaches are developed to address this problem. In chapter 2, the short-term scheduling problem is solved considering uncertainty using a two-stage stochastic programming approach. The production schedule for the first stage is determined considering the long-term objective of expected cost, where uncertainty is considered following a scenario-based model. To address the issue of computation complexity in optimizing scheduling problem a number of decomposition techniques are presented in chapter 3 for the efficient solution of large-scale scheduling problems. Heuristic based approaches are developed leading to an order of magnitude reduction of required computational time. The relaxation of different sets of constraints and variables is investigated in order to derive the tightest upper bound for Lagrangean relaxation and

Lagrangean decomposition. Based on these approaches, an iterative algorithmic procedure is proposed resulting in the determination of schedules for realistic size scheduling problems. Two approaches are developed for the integration of production planning and scheduling. In chapter 4, a new formulation is presented to this problem based on the idea of periodic scheduling. The proposed continuous-time formulation corresponds to a mixed integer nonlinear programming problem that determines the optimal cycle length and schedule. In chapter 5, a hierarchical approach is presented for the more general case, where there is no periodicity in production demands. A solution framework is developed, where the planning problem considers uncertainty utilizing a scenario-based multi-stage model; while the solution of the scheduling problem results in the optimal production schedule for the current time period. An iterative procedure is employed to guarantee that the results from planning and scheduling problems are consistent. The whole approach is implemented based on a rolling horizon strategy. Chapter 6 presents a general approach of improving the Lagrangean decomposition based on a modified Nelder-mead algorithm to update the Lagrangean multipliers which guarantees to give a bound at least as tight as that of the previous iteration.

Acknowledgement

I would like to express my gratitude to my advisor, Marianthi Ierapetritou, for her support, patience, and encouragement throughout my graduate studies. She always finds the time for listening and discussion. Her technical and editorial advice was essential to the completion of this dissertation and has taught me innumerable lessons and insights on the workings of academic research in general.

My thanks also go to the members of my dissertation committee, David Coit, Henrik Pederson and Yee Chiew for reading previous drafts of this dissertation and providing many valuable comments that improved the presentation and contents of this dissertation.

The friendship of Suhrid Balakrishnan is much appreciated and has led to many interesting and good-spirited discussions relating to this research. I am also grateful to my colleagues Jeetmanyu Vin, Zhenya Jia and Vishal Goyal for helping considerably with discussions on optimizations and projects.

My parents, Hanwen and Yiwen, receive my deepest gratitude and love for their dedication and the many years of support during my studies.

Table of contents

List of tables

List of illustrations

CHAPTER 1

INTRODUCTION

Process systems engineering utilizes optimization as a tool to determine the most efficient solution for plant production planning and short-term scheduling problems. Planning and scheduling can be distinguished based on various characteristics. First in terms of the considered time horizon, short-term scheduling spans a time horizon of few days; while the time horizon of planning extends to few months or a year. Second in terms of the decisions involved, short-term scheduling provides feasible production schedule considering the detailed operating conditions; while planning involves consideration of financial and business decisions over extended periods of time. Lastly considering uncertainty, short-term scheduling needs to consider the disturbing events such as rush orders, machine breakdown and attempts to absorb the impact; while the planning needs to foresee the possible changes in the future and the effects of the current decisions thus achieving an optimal solution for the benefits of the entire planning time horizon. However planning and scheduling decisions are always closely coupled in practical industrial plant, which requires the integration of the decision process for planning and scheduling. In this thesis, we analyze the difficulties in the current planning and scheduling work and develop mathematical programming models as well as solution approaches to overcome these issues. In later chapters, two efficient solution approaches are proposed to integrate the production planning and short-term scheduling based on different demand requirements and uncertainty considerations.

Scheduling involves the determination of the order in which different tasks are carried out on different equipments and the detailed timing of the execution of all tasks to optimize plant operation. There are a number of papers published in the last decade focusing on a variety of approaches of formulating the short-term scheduling problem in order to reduce the computational complexity of the resulting mathematical model. Extensive reviews can be found in Reklaitis (1992); Pantelides (1994); Applequist et al. (1997) and Pinto and Shah, (1998). Most of the proposed work can be classified on the basis of time representation. Kondili et al. (1993a) and Shah et al. (1993) developed short-term scheduling models and solution techniques using discrete-time formulation. The discretization of time results in approximation of time horizon and large number of variables. In order to overcome the limitation of time-discretization methods, continuous-time formulation has been given great attention. Zhang and Sargent (1994) presented a mixed integer nonlinear programming (MINLP) formulation based on the resource-state-task (RST) network representation and applied linearization. Mockus and Reklaitis (1996) proposed a MINLP formulation using state-task network (STN) representation employing a Bayesian heuristic approach. Ierapetritou and Floudas (1998) presented a novel continuous-time representation which is described briefly in section 1.1. This representation was proven to reduce the computational complexity of the scheduling problem by taking advantage of the event point concept thus avoiding the use of time slots.

However, there are still a number of issues involved in the efficient solution of short-term scheduling problems. The most important one is the computational complexity associated with realistic case studies due to the increasing dimensionality and the presence of uncertainty. The work on stochastic programming was initiated by Dantzig (1955). Although considerable improvements had been achieved in the following years in terms of algorithmic development and theoretical properties (Wets, 1990), the manageable size of stochastic problems is not at all comparable to practical large-scale problems. In this work, a two-stage stochastic programming approach is used to generate a solution with the optimal expected value. In chapter 2, a realistic industrial scheduling problem is considered where price of energy is the major uncertain parameter. The results illustrate the effectiveness of the proposed stochastic two-stage formulation since the schedule obtained using the proposed approach is the same as the optimal solution if future price of energy is assumed known.

The solution of realistic size scheduling problem is far from being resolved which usually deal with the production of dozens of different products as for example in pharmaceutical and chemical plants utilizing batch and semi-continuous process operations. In Ierapetritou and Floudas' model (1998), the number of binary variables increases proportionally to the number of event points, which is a general characteristic of any scheduling model. This means that for large time horizons (i.e. large number of event points) the computational requirement for the solution of scheduling problem will become intractable. Therefore decomposition appears to be a promising direction to build sub-problems that can be solved to optimality and thus lead to the solution of the original problem. In Chapter 3, first a number of heuristic decomposition techniques are developed to generate fast feasible solutions including time-based decomposition with smoothing technique, required production method and resource-based decomposition. Then Lagrangean relaxation (LR) and Lagrangean decomposition (LD) are utilized to decompose the scheduling problem based on the relaxation of important constraints and variables, respectively. Finally an overall framework is proposed based on generation of lower bound using heuristics and upper bound using LR/LD. The efficiency of the proposed approach is demonstrated with a number of examples where better solutions are obtained utilizing up to an order of magnitude less CPU time.

Production planning problem corresponds to a higher level of process operation decision making since it considers longer time horizon and multiple orders that involve different operating conditions as well as unit changes, price and cost variability. Studies were conducted in this work to integrate the scheduling level consideration within the planning problem. Corresponding to different demand scenarios and uncertainties, two solution approaches are proposed. In the context of a campaign-mode production where demands are relatively stable and uncertainty is minimum over the planning period, a periodic scheduling model is presented in chapter 4 to address the simultaneous consideration of scheduling and planning problem. A continuous-time formulation is exploited based on a scheduling formulation of Ierapetritou and Floudas (1998). New constraints are developed to determine the scheduling decisions between cycles and incorporated into the continuous-time planning model. This model results in an efficient solution of large scale planning problems where scheduling decisions are simultaneously determined. For the case where demand is distributed within the time horizon and changes frequently, a hierarchical solution framework is proposed in chapter 5. The planning and scheduling models are considered within a recursive algorithm that converges to the final optimal schedule. In this framework, uncertainty is considered in the planning problem using scenario-based multi-stage optimization modeling. Although future time periods are considered in the planning model, only the decisions for the current time period are made and the required production is transferred to the scheduling problems. Short-term scheduling model is utilized to generate an optimal schedule that satisfies the production from planning results. In the case where discrepancy appears such as over-optimistic or under-estimated planning results, an iterative procedure is employed to resolve the difference with necessary adjustments until the results become consistent.

Lagrangean relaxation and Lagrangean decomposition have been used to decompose the planning and scheduling problems as other authors reported in their work. However the performance of Lagrangean approaches for practical problems is not always satisfying due to their poor convergence. As a result, an improved Nelder-Mead based algorithm is developed to update the Lagrangean multipliers which guarantees the bound generated is at least as good as that of the previous iteration. This approach provides a good alternative to the current prevalent subgradient method, and can be exploited when subgradient method fails to improve the Lagrangean objective function as illustrated in the case studies. Since Lagrangean approaches can substantially improve the solution of practical size problems, the proposed framework is presented in the last chapter of the thesis.

1.1 MATHEMATICAL FORMULATION

In this section the mathematical formulation proposed by Ierapetritou and Floudas (1998) for the deterministic schedule is briefly presented since it constitutes the main building block of the approaches developed in this work. To motivate the need of the proposed research a small example is then presented, which will be repeatedly used in the following chapters.

Allocation Constraints

$$\sum_{i \in I_j} wv(i,n) = yv(j,n), \qquad \forall j \in J, n \in N \tag{1-1}$$

Capacity Constraints

$$V_{ij}^{\min} wv(i,n) \leq B(i,j,n) \leq V_{ij}^{\max} wv(i,n), \qquad \forall i \in I, j \in J_i, n \in N \tag{1-2}$$

Storage Constraints

$$ST(s,n) \leq ST_s^{\max}, \qquad \forall s \in S, n \in N \tag{1-3}$$

Material Balances

$$ST(s,n) = ST(s,n\text{-}1) - d(s,n) + \sum_{i \in I_s} \rho_{si}^{p} \sum_{j \in J_i} B(i,j,n\text{-}1) - \sum_{i \in I_s} \rho_{si}^{c} \sum_{j \in J_i} B(i,j,n), \quad \forall s \in S, n \in N \tag{1-4}$$

$$ST(s,n) = STIN) - d(s,n) - \sum_{i \in I_s} \rho_{si}^{c} \sum_{j \in J_i} B(i,j,n), \quad \forall s \in S, n \in N, n = 1 \tag{1-5}$$

Demand Constraints

$$\sum_{n \in N} d(s,n) \geq r_s, \qquad \forall s \in S \tag{1-6}$$

Duration Constraints

$$T^{f}(i,j,n) = T^{s}(i,j,n) + \alpha_{ij} wv(i,n) + \beta_{ij} B(i,j,n), \quad \forall i \in I, j \in J_i, n \in N \tag{1-7}$$

Sequence Constraints: Same task in the same unit

$$T^s(i,j,n+1) \geq T^f(i,j,n) - H(2 - wv(i,n) - yv(j,n)), \qquad \forall i \in I, j \in J_i, n \in N, n \neq N \tag{1-8}$$

$$T^s(i,j,n+1) \geq T^s(i,j,n), \qquad \forall i \in I, j \in J_i, n \in N, n \neq N \tag{1-9}$$

$$T^f(i,j,n+1) \geq T^f(i,j,n), \qquad \forall i \in I, j \in J_i, n \in N, n \neq N \tag{1-10}$$

Sequence Constraints: Different tasks in the same unit

$$T^s(i',j,n+1) \geq T^f(i,j,n) - H(2 - wv(i',n) - yv(j,n)),$$
$$\forall j \in J, i \in I_j, i' \in I_j, i \neq i', n \in N, n \neq N \tag{1-11}$$

Sequence Constraints: Different tasks in different units

$$T^s(i',j',n+1) \geq T^f(i,j,n) - H(2 - wv(i',n) - yv(j',n)),$$
$$\forall j, j' \in J, i_j, i' \in I_j, i \neq i', n \in N, n \neq N \tag{1-12}$$

Sequence Constraints: Completion of previous tasks

$$T^s(i,j,n+1) \geq \sum_{n' \in N, n' \leq n} \sum_{i' \in I_j} (T^f(i',j,n') - T^s(i',j,n')),$$
$$\forall i \in I, j \in J_i, i \neq i', n \in N, n \neq N \tag{1-13}$$

Time Horizon Constraints

$$T^f(i,j,n) \leq H, \qquad \forall i \in I, j \in J_i, n \in N \tag{1-14}$$

$$T^s(i,j,n) \leq H, \qquad \forall i \in I, j \in J_i, n \in N \tag{1-15}$$

Objective: Maximization of profit

$$\max \sum_{s} \sum_{n} price_s \times d(s,n) \tag{1-16}$$

These constraints can be classified as mass-related constraints (1-1)-(1-6) and time-related constraints (1-7)-(1-15). Constraints (1-1) state that only one task can be performed in the same unit at each event point *n*. Constraints (1-2) enforce the requirement for minimum amount, V_{ij}^{min} of material in order for a unit *j* to start processing task *i*, and the maximum capacity of a unit V_{ij}^{max}, to correspond to lower and upper bounds on the capacities of *B(i,j,n)* when task *i* is performed (i.e. *wv(i,n)* equals one.). All *B(i,j,n)* variables are forced to zero when *wv(i,n)* equals zero. Maximum storage capacity for each state *s* is represented as upper bound to storage of state *s* at each event point *n* in constraints (1-3). Material balances (1-4) and (1-5) state that the amount of material of state *s* at event point *n* is equal to that at event point *n*-1 adjusted by any amounts produced or consumed between the event points *n*-1 and *n* and the amount delivered to the market at event point *n*. Demand constraints (1-6) express that the production needs to satisfy the market orders.

The time-related constraints (1-7)-(1-15) are very important since they enforce the optimal sequencing and timing of all the tasks that satisfy the mass-related requirements. In this work, the duration is assumed a variation of 1/3 around the mean value of the processing time τ_{ij}^{mean}. α_{ij} takes the value of $2/3\tau_{ij}^{mean}$ and corresponds to the minimum processing time τ_{ij}^{min}, while

$$\beta_{ij} = \frac{\tau_{ij}^{\max} - \tau_{ij}^{\min}}{V_{ij}^{\max} - V_{ij}^{\min}}$$

$$where \quad \tau_{ij}^{\max} = \frac{4}{3}\tau_{ij}^{mean}, \tau_{ij}^{\min} = \frac{2}{3}\tau_{ij}^{mean}$$

and expresses the time required by the unit j to process one unit of material when performing task i. Constraints (1-7) express the dependence of the time duration of task i in unit j at event point n from the amount of material being processed. Note that when $wv(i,n)$ equals zero, the last two terms become zero due to the capacity constraints (1-2) and hence $T^f(i,j,n) = T^s(i,j,n)$. The sequence constraints enforce the recipe requirements between starting and final times of different tasks. Sequence constraints (1-8)-(1-10) state that task i starting at event point n+1 should start after the end of the same task performed at the same unit j which has already started at event point n. Constraints (1-11) establish the relationship between the starting time of a task i at point n+1 and the end time of task i' at event point n when these tasks take place at the same unit. Similarly, constraints (1-12) represent the order of different tasks i,i' that are performed in different units j,j' but take place consecutively according to the production recipe. The sequence constraints (1-13) represent the requirement of a task i to start after the completion of all the tasks performed at past event points in the same unit j. Time horizon constraints (1-14) and (1-15) require that all the tasks have to start and end within the time horizon.

1.2 EXAMPLE 1

In this example (Example 2 in Ierapetritou and Floudas, 1998), two different products are produced through five processing stages: heating, reactions 1, 2, and 3 and separation of

Product 2 from Impure E as shown in the State Task Network (STN) representation of the process in Figure 1-1. The data for this example are presented in Table 1-1.

Unit	**Capacity**	**Suitability**	**Mean Processing Time (τ_{ij}^{mean})**
Heater	100	Heating	1.0
Reactor 1	50	Reaction 1,2,3	2.0,2.0,1.0
Reactor 2	80	Reaction 1,2,3	2.0,2.0,1.0
Still	200	Separation	2.0
State	**Storage Capacity**	**Initial Amount**	**Price**
Feed A	Unlimited	Unlimited	0.0
Feed B	Unlimited	Unlimited	0.0
Feed C	Unlimited	Unlimited	0.0
Hot A	100	0.0	0.0
Int AB	200	0.0	0.0
Int BC	150	0.0	0.0
Impure E	200	0.0	0.0
Product 1	Unlimited	0.0	10.0
Product 2	Unlimited	0.0	10.0

Table 1-1: Data for Example 1

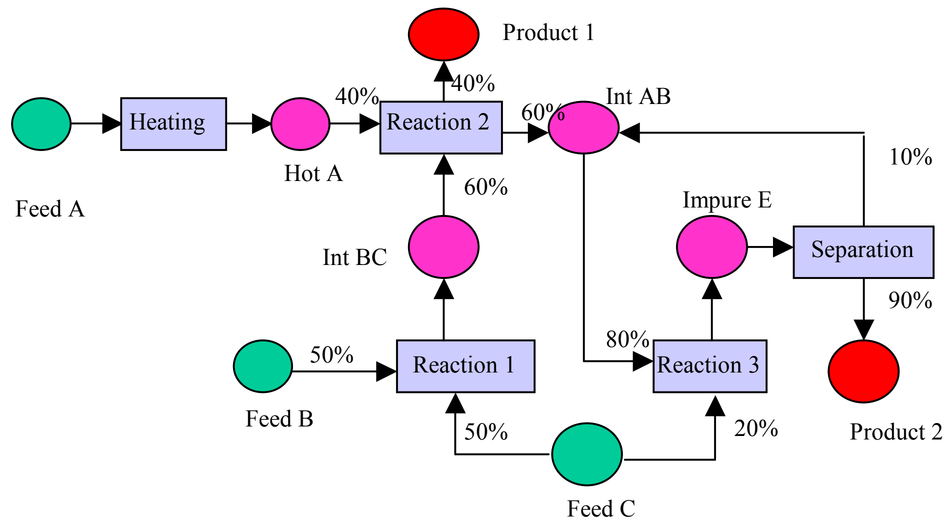

Figure 1-1: State Task Network for Example 1

The objective is to achieve maximum profit with given time horizon and sufficient raw materials. Formulation in section 1.1 is applied to this example and the problem is solved on Sun Ultra 60 workstation using CPLEX 6.6. The results corresponding to different time horizons are presented in Table 1-2.

Time Horizon	Number of Event Points	Objective Function	CPU time (sec)
8 hrs	5	1498.19	0.47
16 hrs	9	3737.10	177.93
24 hrs	13	6034.92	92367.94

Table 1-2: Results for Example 1 without Decomposition

Note that for time horizons of 8 and 16 hours the objective function corresponds to the optimal solution since further increase of the number of event points does not affect the value of the objective function (Ierapetritou and Floudas, 1998). For 24 hours however the solution can be sub-optimal since further increase of the number of event points makes the problem computationally infeasible to solve. Thus, the need to develop decomposition-based approaches becomes imperative.

It should also be pointed out that the optimal solution for time horizons of 8 and 16 hours corresponds to the solution of the deterministic model which means that all the parameters in this example are known. When uncertainty is taken into account, the same schedule could be sub-optimal since a change in price of products or demand will result in different optimal schedules. For example, the optimal production for time horizon of 16 hours is 147.533 for P1 and 224.764 for P2 and the corresponding schedule is shown in Figure 1-2. If a price change is considered such that the price of P1 rises to 11 and the price of P2 remains the same, the corresponding optimal production is 150.318 for P1 and 216.000 for P2. As shown in Figure 1-3, the optimal schedule is different from the previous one. Therefore it is important to incorporate uncertainty in the decision-making process.

Planning usually considers a time horizon of months. Obviously short-term scheduling formulation cannot lead to detailed scheduling decisions for this long period due to the computational difficulty revealed. In chapter 4 and 5, we are going to revisit this example in planning time scale using proposed planning and scheduling approaches.

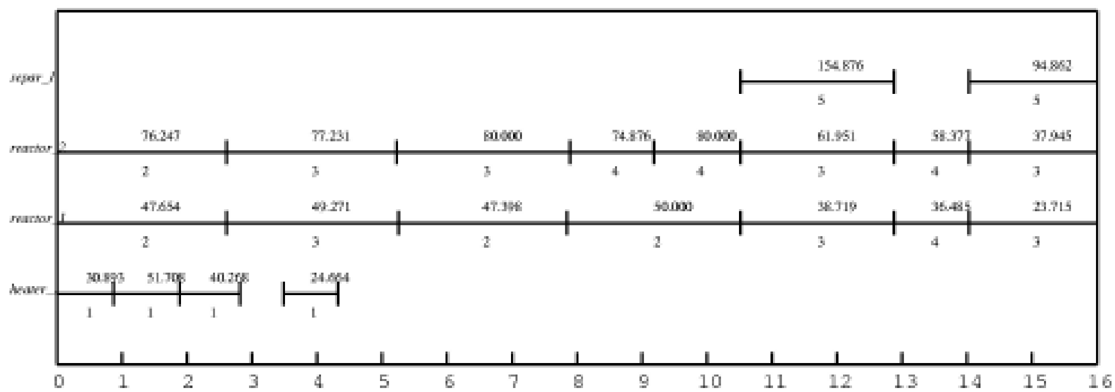

Figure 1-2: Optimal Schedule of Example 1

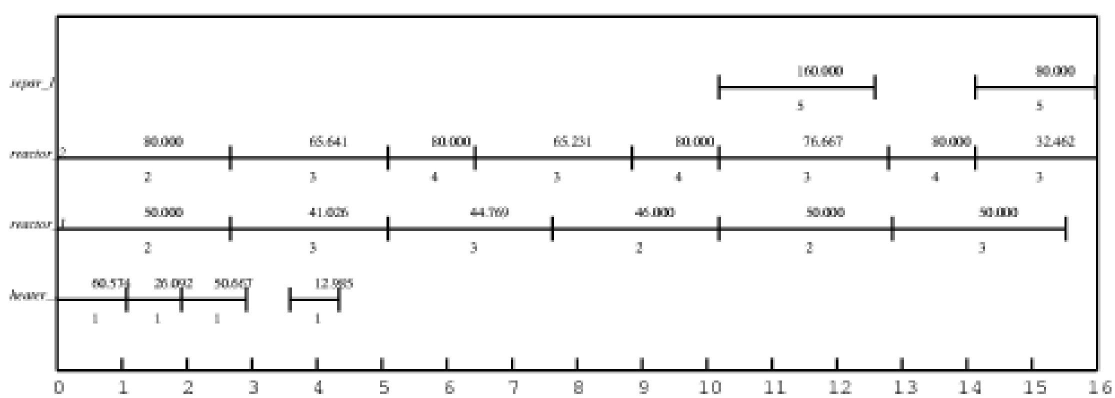

Figure 1-3: Optimal Schedule of Example 1 Considering Price Variability

CHAPTER 2

SHORT-TERM SCHEDULING UNDER UNCERTAINTY

The short-term scheduling problem with random prices and demands is considered in this chapter. A two-stage stochastic programming approach is proposed that utilizes a forecasting technique to generate scenarios of the uncertain parameters. A mathematical model is developed for an industrial problem where the applicability of the proposed approach is illustrated.

2.1. INTRODUCTION

As stated in chapter 1, a significant amount of work in the area of short-term scheduling has focused on the development of deterministic models, where the problem data are assumed well known in advance. In reality, uncertainty in the description of a number of different parameters such as processing times, costs and demands may have considerable impact on the objectives imposed by deterministic models. Consequently, the most suitable approach to handle these uncertainties is through the use of probabilistic models that describe the model parameters in terms of probability distribution.

The probabilistic models take into account the detailed statistical properties of the parameter variations and utilize two main solution approaches, the chance-constrained programming and two-stage stochastic programming. Chance-constrained programming attempts to reconcile optimization over uncertain constraints. These constraints, which contain uncertain parameters, are guaranteed to be satisfied with a certain probability, i.e. reliability level. The chance-constrained programming problem is generally in the following form (Birge and Louveaux, 1997).

$$\begin{aligned} \min \quad & f(x) \\ s.t. \quad & g1(x) \le 0 \\ & \Pr\{g2(p,x) \le 0\} \ge a \\ & x \in \Re^n, p \in \Re^n \end{aligned} \tag{2-1}$$

where x is the decision variable vector and p is the uncertain parameter vector which is contained in the probabilistic constraint. If the probability density function of p is known, then the probabilistic constraint can in principle, be substituted by a deterministic constraint of the form,

$$g3(x) \le 0, \tag{2-2}$$

thus the entire optimization problem can be solved with a nonlinear solver. In the case where only a single probabilistic constraint exists in the model, the solution can be derived simply by a coordinate transformation. In the case where uncertain parameters follow quasi-concave probability distribution, the relaxed problem is proved convex which can be easily solved (Kall and Wallace, 1994). The probabilistic constraint with normal distribution uncertain parameters can be computed with an efficient simulation approach (Prekopa, 1995). This approach has been utilized in linear predictive control model (Schwarm and Nikolaou, 1999; Li et al., 2000) and optimal polymer design (Maranas, 1997). Depending on the form of *g2*, however, the explicit form of *g3* may be difficult to obtain. For example, the chance-constrained programming is not well developed for nonlinear processes due to the difficulty in determining the relations between the uncertain parameters and the output constraints.

The two-stage stochastic programming approach is most commonly used in chemical engineering literature for process planning problem (Subrahmanyanm, 1994; Pistikopoulos and Ierapetritou, 1995; Clay and Grossmann, 1997; Acevedo and

Pistikopoulos, 1998 and Gupta and Maranas, 2000) since the problem can be converted to a large deterministic problem for discrete distributed uncertain parameters. The general form of the two-stage stochastic programming formulation is the following (Dantzig 1989):

$$\begin{aligned}
&\min_{x_1, x_2} \quad \underset{\theta}{E}[f_1(x_1) + f_2(x_2, \theta)] \\
&s.t. \quad g_1(x_1) + g_2(x_2, \theta) \le 0 \\
&\qquad\ \ h_1(x_1) + h_2(x_2, \theta) = 0
\end{aligned} \tag{2-3}$$

where x_1 is the vector of the first stage decision variables corresponding to the time period where the value of the involving parameters exists, whereas x_2 is the vector of the second stage decision variables for the time periods where the scheduling decisions depend on the specific realizations of the uncertain parameters, θ. The advantage of such an approach is that the first stage decisions take into account the uncertainty in future parameters. This model allows superior decisions to be made since it considers the risk of variability in the model parameters in the future. In order to incorporate the future variability of uncertain parameters, a scenario-based approach for the second stage decision model is presented. To illustrate the scenario-based two-stage approach, a parametric example is given as follows. Considering the problem:

$$\begin{aligned}
&Min \quad c^T x + E[Q(x, w)] \\
&s.t. \quad Ax = b \\
&\qquad\ \ x \ge 0 \\
&where \quad Q(x, w) = Min \quad d(w)^T y \\
&\quad s.t. \quad T(w)x + W(w)y(w) = h(w),
\end{aligned} \tag{2-4}$$

the first objective is the minimization of the first stage direct costs $c^T x$ plus the expected recourse cost $E[Q(x,w)]$ over all of the possible scenarios while meeting the first stage constraints $Ax = b$. The recourse cost Q depends both on x, the first stage decision, and on the random event w and describes the optimal selection of the second-stage decisions $y(w)$ that depend on the realization of the uncertain parameters w. It minimizes the cost $d^T(w)y$ subject to recourse function, $T(w)x + W(w)y(w) = h(w)$. Nonanticipativity property is assumed here which means that the decision in the first stage x is independent of which second stage scenario actually occurs, because the decision at the current time cannot take advantage of knowledge in the future. Problem (2-4) can be expressed in a deterministic equivalent formulation (2-5) by introducing a second stage variable y_i for each scenario i.

$$
\begin{aligned}
Min \quad & c^T x + \sum_{i=1}^{N} p_i d_i^T y_i \\
s.t. \quad & Ax = b \\
& T_i x + W_i y_i = h_i, \quad i = 1,\dots,N \\
& x \geq 0 \\
& y_i \geq 0
\end{aligned}
\tag{2-5}
$$

where N is the number of scenarios and p_i is the probability of the scenario to occur. The first stage decision cannot discriminate one scenario from another and must be feasible for each scenario, *i.e.* $Ax = b$ and $T_i\, x + W_i\, y_i = h_i$ for all $i = 1,...,N$. Because all the decisions x and y_i are solved simultaneously, x is determined to be optimal over all the scenarios. Thus the nonanticipativity property is maintained.

In the rest of this chapter, a realistic problem is used to illustrate the applicability the two-stage stochastic approach in the production scheduling problems. The problem is described in section 2.2 followed by two-stage stochastic mathematical model in section 2.3. In section 2.4, forecasting techniques are discussed and incorporated in the solution framework. Results by using proposed approach are compared with that of deterministic model.

2.2 PROBLEM STATEMENT

Cryogenic air separation technology is currently the most efficient and cost-effective technology for the production of large quantities of oxygen, nitrogen and argon as gaseous and liquid products. A detailed review of cryogenic air separation processes and comparison with other air separation alternatives can be found in a recent paper by Smith and Klosek 2001. Most of the work that appears in the literature to date addressing the optimization of air separations deals with the energy integration and process synthesis alternatives as ways to improve energy efficiency. This work presents a completely different application of process optimization focused on operational optimization when there is power price variability. One difficulty that the plant faces during operation is the power price at which the utility company supplies electricity to the plant. The power prices are often subject to high fluctuations, which can significantly increase the total production cost in the plant. To deal with power cost variability the plant can operate in three different modes (regular, assisted, shutdown) that vary with respect to operation efficiency and energy requirements. Regular mode of operation is the most efficient and most expensive one. When the plant is running on the regular mode the plant power consumption consists of air separation unit power which is approximately 20% of total power and liquefier power which corresponds to the rest 80% of the total power consumption. This means that the plant consumes the maximum amount of power on the regular mode and if the power price is high the regular mode is very expensive.

Obviously, the shutdown mode is the cheapest one because the plant consumes the only miscellaneous power. When the plant is running on the assisted mode the liquefier is shut down, so the plant consumes only around 20% of the total power however, it requires the consumption of one of the plant products from the storage (which was made early) for the refrigeration. It must be noted that there is some extra cost for both shutdown and assisted mode, which is related to the recovery of the plant back to the regular mode. Consequently, when the energy cost increases the plant can shift from regular to the alternative assisted mode which is less expensive or in some cases, to the shutdown mode in which there is no production but the plant continues to satisfy the demand by utilizing the stored inventory. A method to generate a detailed schedule of process operation mode and production rates that minimizes the total cost of power for plant operation is needed. The problem is made more challenging because the power price is typically known for only a portion of the desired scheduling horizon. For the rest of the horizon, the power price is uncertain and can be only forecasted. There is also uncertainty in the product demands from the plant. This uncertainty is not addressed explicitly; however, special attention is given to the storage tank level calculations to minimize the effect of this uncertainty.

In order to optimize plant production taking into account the uncertainty in future information, a two stage stochastic programming approach was developed. This approach was chosen since it would allow the power price uncertainty to be addressed in a manner where the risk of incurring high cost operation periodically could be traded off against the benefit of low cost operation most of the time. This required development of mathematical models of the plant operation, formulation the objective function for the problem, development of a robust procedure for solving the optimization problem in a reasonable amount of time and development of a forecasting model for power prices. The forecasting model for power prices should be based on historical power price data and must be capable of generating alternative price scenarios for use in the stochastic programming approach.

2.3 MATHEMATICAL MODEL

In the scheduling of air separation process operations, uncertainty appears as a result of internal and external influences. Product flow rate and distribution variability are influences internal to the plant while power price uncertainty is an external influence. The internal influences on uncertainty are generally constant with respect to time and small with respect to the total storage volume. Consequently, they will not need to be changed each time the schedule is prepared. Therefore, these uncertainties are accounted for in the storage tank level constraints. On the other hand, the external influence of power price on uncertainty is not generally constant with time and will change each time the schedule is prepared. The two-stage stochastic programming approach is implemented to account the effects of power price uncertainty on production scheduling.

Storage tank capacity constraints include the parameters, Δ, to take care of the uncertainty in the production and distribution of liquid N_2 and O_2 over a time step i. The schedule assumes production changes instantly every hour. This does not happen in reality. Distribution entails emptying the tank into discrete trailers. It is very unlikely (in a real situation) that this is going to occur over exactly one time step and that the trailer filling rate will be constant over the time horizon. To account these problems, an additional safety parameter Δ is added in both sides of the liquid level constraints.

The power price *c(i)* is an hourly varying parameter. The power prices for three days in the future are assumed deterministically known in this work. The power prices beyond three days are assumed to be stochastic. If the production schedule is set using only the deterministic information, future changes in price may cause the current best schedule to be significantly less than optimal over a longer time horizon. For example, knowledge that prices will be higher in the future should cause the schedule to favor more periods of operation in regular mode during the scheduling horizon even if the power prices within the horizon are currently high relative to normal expectations. The power price forecast could be assumed to be correct and treated as deterministic information to solve the scheduling problem over a longer time horizon. However, this approach would not allow the risk due to power price uncertainty to be managed.

The two-stage stochastic modeling approach was developed to accommodate the need to consider uncertainty in the future prices as early as possible in the decision making process to better manage risk. The basic assumption made in developing the mathematical model for the power optimization considering power price uncertainty in future time periods was to separate the decision making into two stages. In the first stage, the power prices are assumed known with specific deterministic values. The second stage decisions represent the operating schedule and production in the future time period where power prices are uncertain. In particular, the two-stage model developed here involves a first stage consisting of three days where we assume that the power price is known with certainty and a second stage of three to five days where forecasted power prices are used.

Note that since the feasibility of the operations schedule does not depend on power price, no consideration of future feasibility is necessary.

2.3.1 FIRST STAGE

The indices, variables and parameter definitions used in the proposed mathematical formulation are in the notation section. On the basis of this notation, the mathematical model of the air separation plant is presented below:

Balance Equations and Productivity Constraints

$$
\begin{aligned}
&A^{rm}x^{rm}(i) = 0 \\
&where \quad x^{rm}(i) = [x^{rm}_{LIN}(i), x^{rm}_{LOX}(i), x^{rm}_{ARG}(i), x^{rm}_{GOX}(i), x^{rm}_{AIR}(i), x^{rm}_{VENT}(i), x^{rm}_{LIQ}(i)]^T, i = 1,2,...I \\
&A^{am}x^{am}(i) = 0 \\
&where \quad x^{am}(i) = [x^{am}_{LIN}(i), x^{am}_{LOX}(i), x^{am}_{ARG}(i), x^{am}_{GOX}(i), x^{am}_{AIR}(i), x^{am}_{VENT}(i)]^T, i = 1,2,...I
\end{aligned}
\tag{2-6}
$$

The variables $x^j_p(i)$ correspond to the production rate of product p during time period i when the plant operates in the mode j *(rm, am)*. A^j and b^j are the coefficient matrix and right hand side vector respectively (Coefficient Matrix and Other Parameters section). I is number of time steps in the optimization horizon.

Capacity Constraints for Mode j=*rm* and *am* (regular mode and assisted mode)

$$
\alpha^j p^j(i) \le B^j x^j(i) \le \beta^j p^j(i), \qquad j = \{rm, am\} \tag{2-7}
$$

α^j_p, β^j_p are the minimum and maximum production rates of product p when the plant operates in the mode j (Values of α^j, β^j and matrix B^j see Coefficient Matrix and Other Parameters section). $p^j(i)$ is a binary variable indicating which mode the plant is operating in during time period *i*. These constraints are derived from the capacity limits of the equipment used in the two modes.

Storage Constraints

$$L(i+1) = L(i) + C^{rm} x^{rm}(i) + C^{am} x^{am}(i) - Q(i) - V$$
$$where \quad L(i) = \begin{bmatrix} L_{LIN}(i) \\ L_{LOX}(i) \end{bmatrix}, L(1) = L_{initial}, \quad Q(i) = \begin{bmatrix} Q_{LIN}(i) \\ Q_{LOX}(i) \end{bmatrix} \tag{2-8}$$

These constraints connect step i+1 with the previous step i. According to these constraints the level of liquid N_2 and O_2 in the storage tank at time step i+1 ($L_{LIN}(i+1)$, $L_{LOX}(i+1)$), equals the level of liquid O_2 and N_2 in the tank at the previous time step ($L_{LIN}(i)$, $L_{LOX}(i)$), adjusted by the amounts produced and distributed from the storage to customers. $Q(i)=(Q_{LIN}(i), Q_{LOX}(i))^T$ – the amount of distributed product during the time step i. A small correction term that reflects material loss during storage and distribution is also added: $V= (V_{LIN}, V_{LOX})^{\mathrm{T}}$. C^r and C^a are matrices depicted in Coefficient Matrix and Other Parameters section.

Storage Tank Level Bounds

$$L_{\min} - Slack(i) \le L(i) \le L_{\max,}$$
$$where \quad L_{\min} = \begin{bmatrix} S_{LIN}^{\min} - \Delta_{LIN}^{low} \\ S_{LOX}^{\min} - \Delta_{LOX}^{low} \end{bmatrix}, L_{\max} = \begin{bmatrix} S_{LIN}^{\max} - \Delta_{LIN}^{high} \\ S_{LOX}^{\max} - \Delta_{LOX}^{high} \end{bmatrix}, Slack(i) = \begin{bmatrix} Sin(i) \\ Slox(i) \end{bmatrix} \tag{2-9}$$

These constraints express the requirement for a minimum level of liquid N_2 and O_2 in the storage tanks ($S_{LIN}^{\min}$ and $S_{LOX}^{\min}$), to enable the efficient pumping of the liquid from the tank. The maximum level constraints ($S_{LIN}^{\max}$ and $S_{LOX}^{\max}$) arise from the storage tank capacity limits. Slack variables *Slin(i)* and *Slox(i)* are introduced to relax the lower bound on the liquid level in the storage tanks. This relaxation needs to be done to accommodate instances where the total demand for liquid oxygen/nitrogen over the time horizon under consideration exceeds the amount that can be produced by the plant in that time. The slack variables prevent the problem from becoming infeasible and failing to deliver a result. It is important to note that the objective function should minimize not only the cost of operation, but also the slack variables (through an appropriately scaled penalty parameter), so that the constraint is satisfied as closely as possible. Since the slack variables are used to relax the lower bound of inventory levels, the existence of nonzero values does not imply solution infeasibility and thus is acceptable.

Logical Constraints

In order to reflect the ability of the plant to switch between three different operating modes, we introduce the following binary variables and constraints.

$$E \cdot p(i) = 1,$$
$$where \quad p(i) = \begin{bmatrix} p^{rm}(i) \\ p^{am}(i) \\ p^{sh}(i) \end{bmatrix}, \quad E = [1 \quad 1 \quad 1] \tag{2-10}$$

This constraint ensures that only one mode of operation is selected at any time step. For example, if the plant operates in regular operation mode, then $p^{rm}(i) = 1$ and $p^{am}(i) = 0$ and $p^{sh}(i) = 0$ and capacity constraints for regular mode correspond to the upper and lower bounds on the production rates for the various products. In assisted mode the upper and lower bounds will be those appropriate for the corresponding production mode.

Switch variable constraints

The variables $Sw^{am}(i)$ and $Sw^{sh}(i)$ are introduced into the system to account for the minimum switching time between the operating modes. $Sw^{am}(i)$ takes a value of 1, whenever the mode of operation shifts from regular at time step i-1 to assisted at time step i (which means that there is a 'switch' to assisted mode) and is 0 otherwise. Similarly Sw^{sh}(i) takes a value of 1 whenever the mode of operation shifts from regular or assisted at time step i-1 to shutdown at time step i and is 0 otherwise.

$$
\begin{aligned}
&Sw(i) \geq H \cdot p(i-1) + F \cdot p(i) - 1 \\
&Sw(i) \leq F \cdot p(i) \\
&Sw(i) \leq J \cdot p(i-1) \\
&where \;\; Sw(i) = \begin{bmatrix} Sw^{am}(i) \\ Sw^{sh}(i) \end{bmatrix}
\end{aligned}
\tag{2-11}
$$

Matrices H, F and J are depicted in Coefficient Matrix and Other Parameters section.

In this example, two operating criteria for mode switching have been introduced into the model. These are:

- If the plant switches into the assisted mode of operation, then it must remain in that mode for at least four time steps.
- If the plant switches into the shutdown mode of operation, then it must remain in that mode for at least eight time steps.

The first condition is satisfied using the following condition:

$$
\begin{aligned}
&\sum_{t=i}^{i+3} Q^{am} \cdot p(t) \leq 8\left(1 - G^{am} \cdot Sw(t)\right) \\
&where \;\; Q^{am} = [1 \quad 0 \quad 1], \quad G^{am} = [1 \quad 0]
\end{aligned}
\tag{2-12}
$$

If the mode of operation switches to assisted mode at time step i, *i.e.* $Sw^{am}(i) = 1$, then the LHS of equation becomes zero, forcing all p^{rm} and p^{sh} variables for the next four time steps to be zero. This ensures that the plant remains in assisted mode for the next four hours. If $Sw^{am}(i) = 0$, then the condition is relaxed. The second requirement is satisfied by a similar condition:

$$\sum_{t=i}^{i+7} Q^{sh} p(t) \leq 16\left(1 - G^{sh} Sw(t)\right)$$
$$where \quad Q^{sh} = [1 \quad 1 \quad 0], \quad G^{sh} = [0 \quad 1] \tag{2-13}$$

If $Sw^{sh}(i) = 1$, this forces all p variables on the LHS to be zero, thus ensuring that the plant remains in Shutdown mode for eight hours. If $Sw^{sh}(i) = 0$, the condition is relaxed.

<u>*Power Consumption Model:*</u>

$$pw(i) = pw_{ASU}(i) + pw_{NLU}(i) + pw_{misc} \tag{2-14}$$

$$pw_{ASU}(i) = K^0_{ASU}(1 - p^{sh}(i)) + K^1_{ASU}[D^{rm}_{ASU} x^{rm}(i) + D^{am}_{ASU} x^{am}(i)] + K^2_{ASU}[D^{rm}_{ASU} x^{rm}(i) + D^{am}_{ASU} x^{am}(i)]^2, \tag{2-15}$$
$$where$$
$$D^{rm}_{ASU} = [0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0]; \quad D^{am}_{ASU} = [0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0];$$

$$pw_{NLU}(i) = K^0_{NLU} p^{rm}(i) + K^1_{NLU}[D_{NLU} x^{rm}(i)] + K^2_{NLU}[D_{NLU} x^{rm}(i)]^2, \tag{2-16}$$
$$where \quad D_{NLU} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1];$$

pw(i) denotes total power consumption of the plant in the time period *i*. $pw_{ASU}(i)$ the power consumed by the air separation unit, $pw_{NLU}(i)$ the power consumed by the liquefier and pw_{misc} the miscellaneous power, which is used for plant support needs, such as lights, instruments etc. This amount of power is assumed to be constant for all operating modes and does not vary with time.

2.3.2 SECOND STAGE

In order to determine the expected value of the objective function, the second stage mathematical model must be developed to include a set of power price scenarios. Denoting with I^1 the number of steps of the first stage of the horizon, with *q* - the subscript of the set of scenarios considered for each step of the second stage and the continuous and binary variables for the second stage $x2^j_q(i), L2_q(i), pw2_q(i)$ and $p2^j_q(i)$, $Sw2^j_q(i)$, we will get the following set of constraints that are similar to (2-6)- (2-16):

Balance Equations and Productivity Constraints

$$\begin{aligned}
&A^{rm} x2^{rm}_q(i) = 0 \\
&where \quad i = \{I^1+1,, I^1+2,..., I), \forall q \in \{1,2,...,N^q\} \\
&x2^{rm}_q(i) = [x2^{rm}_{LIN,q}(i), x2^{rm}_{LOX,q}(i), x2^{rm}_{ARG,q}(i), x2^{rm}_{GOX,q}(i), x2^{rm}_{AIR,q}(i), x2^{rm}_{VENT,q}(i), x2^{rm}_{LIQ,q}(i)]^T \quad \text{(2-17)} \\
&A^{am} x2^{am}_q(i) = 0 \\
&where \quad i = \{I^1+1,, I^1+2,..., I), \forall q \in \{1,2,...,N^q\} \\
&x2^{am}_q(i) = [x2^{am}_{LIN,q}(i), x2^{am}_{LOX,q}(i), x2^{am}_{ARG,q}(i), x2^{am}_{GOX,q}(i), x2^{am}_{AIR,q}(i), x2^{am}_{VENT,q}(i), x2^{am}_{LIQ,q}(i)]^T
\end{aligned}$$

N^q is the total number of scenarios.

Capacity Constraints for Mode *j=rm, am*

$$\alpha^j p2_q^j(i) \le B^j x2_q^j(i) \le \beta^j p2_q^j(i), \qquad j = \{rm, am), \forall q \in \{1,2,\dots,N^q\} \tag{2-18}$$

Storage Constraints

$$\begin{aligned} &L2_q(i+1) = L2_q(i) + C^{rm} x2_q^{rm}(i) + C^{am} x2_q^{am}(i) - Q(i) - V \\ &where \quad L2_q(i) = \begin{bmatrix} L2_{LIN,q}(i) \\ L2_{LOX,q}(i) \end{bmatrix}, L2_q(1) = L(I^1); \forall q \in \{1,2,\dots,N^q\} \\ &\qquad\qquad Q2_q(i) = \begin{bmatrix} Q2_{LIN,q}(i) \\ Q2_{LOX,q}(i) \end{bmatrix} \end{aligned} \tag{2-19}$$

Note that the integration of the first and second stage models is performed through the levels of product in storage tanks: the final inventory of the first stage ($L_{LIN}(I^1)$, $L_{LOX}(I^1)$) is considered to be the initial inventory of the second stage.

Storage Tank Level Bounds

$$\begin{aligned} &L_{\min} - Slack2_q(i) \le L2_q(i) \le L_{\max,} \\ &where \quad Slack2_q(i) = \begin{bmatrix} Slin2_q(i) \\ Slox2_q(i) \end{bmatrix}, \forall q \in \{1,2,\dots,N^q\} \end{aligned} \tag{2-20}$$

L_{max} and L_{min} are defined in Equation (2-5).

Logical Constraints

$$E \cdot p2_q(i) = 1,$$
$$where \quad p2_q(i) = \begin{bmatrix} p2_q^{rm}(i) \\ p2_q^{am}(i) \\ p2_q^{sh}(i) \end{bmatrix}, \forall q \in \{1,2,...,N^q\} \qquad (2\text{-}21)$$

Switch Variable Constraints

$$Sw2_q(i) \geq H \cdot p2_q(i-1) + F \cdot p2_q(i) - 1, \forall q \in \{1,2,...,N^q\} \qquad (2\text{-}22)$$
$$Sw2_q(i) \leq F \cdot p2_q(i), \forall q \in \{1,2,...,N^q\}$$
$$Sw2_q(i) \leq J \cdot p2_q(i-1), \forall q \in \{1,2,...,N^q\}$$
$$where \;\; Sw2_q(i) = \begin{bmatrix} Sw_q^{am}(i) \\ Sw_q^{sh}(i) \end{bmatrix}$$

$$\sum_{t=i}^{i+3} Q^{am} \cdot p2_q(t) \leq 8\left(1 - G^{am} \cdot Sw2_q(t)\right), \forall q \in \{1,2,...,N^q\} \qquad (2\text{-}23)$$

$$\sum_{t=i}^{i+7} Q^{sh} p2_q(t) \leq 16\left(1 - G^{sh} Sw2_q(t)\right), \forall q \in \{1,2,...,N^q\} \qquad (2\text{-}24)$$

Power Consumption Model

$$pw2_q(i) = pw2_{ASU,q}(i) + pw2_{NLU,q}(i) + pw_{misc}, \forall q \in \{1,2,...,N^q\} \qquad (2\text{-}25)$$

$$\begin{aligned} pw2_{ASU,q}(i) = {} & K_{ASU}^0 (1 - p2_q^{sh}(i)) + K_{ASU}^1 [D_{ASU}^r x2_q^{rm}(i) + D_{ASU}^{am} x2_q^{am}(i)] \\ & + K_{ASU}^2 [D_{ASU}^{rm} x2_q^{rm}(i) + D_{ASU}^{am} x2_q^{am}(i)]^2, \forall q \in \{1,2,...,N^q\} \end{aligned} \qquad (2\text{-}26)$$

$$\begin{aligned} pw2_{NLU,q}(i) = {} & K_{NLU}^0 p2_q^{rm}(i) + K_{NLU}^1 [D_{NLU} x2_q^{rm}(i)] \\ & + K_{NLU}^2 [D_{NLU} x2_q^{rm}(i)]^2, \forall q \in \{1,2,...,N^q\} \end{aligned} \qquad (2\text{-}27)$$

2.3.3 OBJECTIVE FUNCTION FORMULATION

For the first stage, the power prices *c(i)* for all time periods are known. The cost of the plant operation in the first stage can be calculated as:

$$PC = \sum_{i=1}^{I^1} c(i) \times pw(i) \tag{2-28}$$

In the second stage, there are different scenarios, q, for the power prices *c2(i,q)* for the remaining time periods in the horizon. The cost of the plant operation in the second stage can be calculated as:

$$PC2 = \sum_{q=1}^{N^q} w_q \sum_{i=I^1+1}^{I} c2(i) \cdot pw2_q(i), \forall q \in \{1,2,...,N^q\} \tag{2-29}$$

w_q correspond to the weight for each scenario. Taking into consideration the general form of the two stage stochastic programming formulation, the objective function should minimize the total cost of the power consumption *PC** across both stages simultaneously subject to the constraints in each stage. The constraints are defined by, (2-6) for all $i=1,2,...I^1$ and (2-17) for all $i = I^1+1, I^1+2, ...I$.

$$\begin{aligned} PC^* &= \min\left\{ PC + PC2 + \sum_{i=1}^{I^1} \mathrm{P} \times Slack(i) + \sum_{q=1}^{N^q} w_q \sum_{i=I^1+1}^{I} \mathrm{P} \cdot Slack2_q(i) \right\} \\ where \quad \mathrm{P} &= [Penalty_{LIN} \quad Penalty_{LOX}], \qquad \forall q \in \{1,2,...,N^q\} \end{aligned} \tag{2-30}$$

2.4 POWER PRICE FORCASTING

The two stage stochastic formulation of the optimization problem requires the utilization of a forecasting technique to predict the future prices and their confidence intervals. The quality of the results depends strongly on the quality of the forecasting model. Extensive testing and analysis of the power forecasting model was done to quantify its accuracy and potential impact on the optimal operating schedule.

2.4.1 ARIMA MODEL

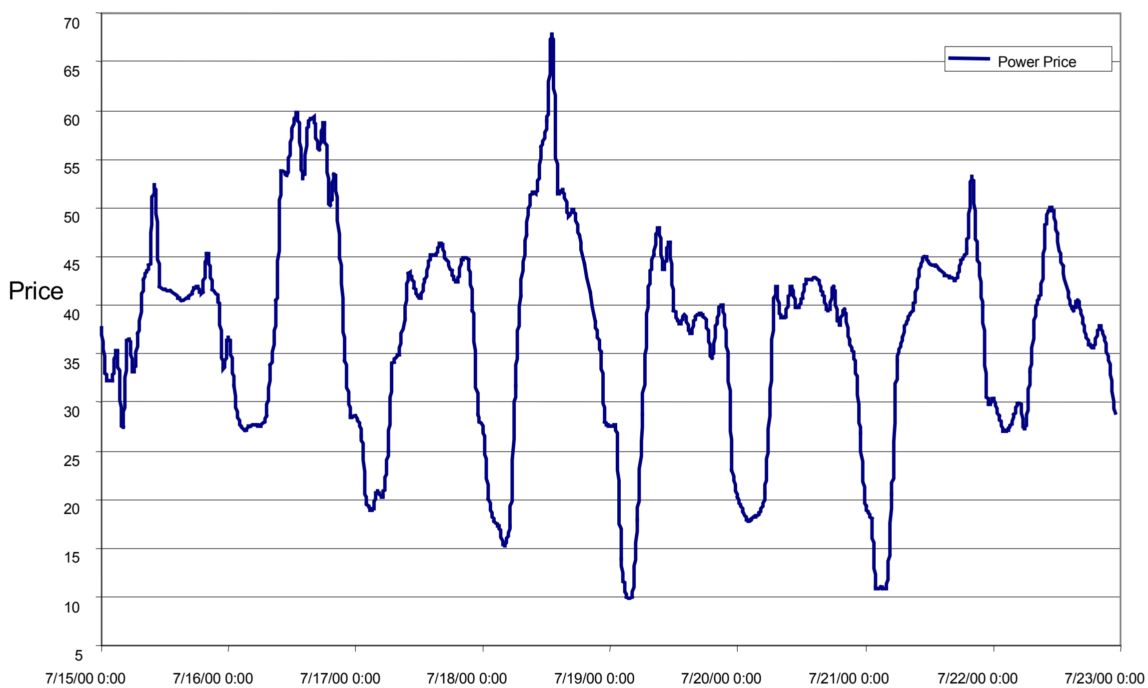

Figure 2-1: Power Price Profile

It can be observed from Figure 2-1, where the price fluctuations are shown for a period of 8 days, that although the power price varies tremendously with time there is a pattern within each day through out the year. As a result of this, the following procedure was developed to predict the future power prices based on historic data. A forecasting method was developed to predict the average daily power price for future days. A separate model was developed which correlates the hour of the day with the ratio of the hourly power price to average power price for the day. The combination of these models allows the power price to be predicted on an hourly basis for the future days as required by the problem formulation.

There is an extended literature in the development and application of forecasting techniques. A survey and detailed comparison of various methodologies can be found in the book by Makridakis et al. 1984. Among the models that we tested for the power price prediction, the moving average with trend and the exponential smoothing method with trend cannot reflect the change of trend of price due to their simplicity. Winters's method for seasonal variation and Harrison's harmonic smoothing method with seasonal estimates through Fourier analysis don't work well since the series of energy price didn't show obvious seasonal pattern. Neither does Brown's quadratic exponential smoothing method since the variation fails to be approximated by quadratic terms. After testing these forecasting methods, we utilized an ARIMA model to forecast the average power prices for a period of 2 to 5 days. An ARIMA (Autoregressive Integrated Moving Average) model has three adjustable parameters:

p - the number of autoregressive terms in the model

d - the number of non-seasonal differences

q - the number of lagged forecast errors in the prediction equation

Depending on the values of those parameters, the generalized ARIMA mathematical model corresponds to different forecasting formulas. For example an ARIMA(0,1,1) model corresponds to a simple exponential smoothing method as shown below:

$$\hat{Y}(t) = Y(t-1) - \theta\, e(t-1) \tag{2-31}$$

where $\hat{Y}(t)$ is the predicted value of variable Y at time t, $Y(t-1)$ is the actual value of Y at time *t-1*, $e(t-1)$ is the lagged forecast error and θ is a constant multiplier.

Using power price data from the ISO New England website for the period from Jan,1st to July,17th, 2000, it was found that an ARIMA(2,1,1) was best at predicting the trend of average daily power price. ARIMA(2,1,1) is a mixed mode model characterized by the following forecasting equation

$$\hat{Y}(t) = \mu + Y(t-1) + \phi_1 (Y(t-1) - Y(t-2)) + \phi_2 (Y(t-2) - Y(t-3)) - \theta\, e(t-1) \tag{2-32}$$

where μ is where the constant term, ϕ_1 is the first order autoregressive term coefficient, ϕ_2 is the second order autoregressive term coefficient. The purpose of introducing these terms is to minimize or eliminate the autocorrelation of the errors. Table 2-1 shows the statistical results comparing the ARIMA(2,1,1) to other ARIMA forecasting models. In Table 2-1, AIC is Akaike's Information criterion and SBC is Schwarz's Bayesian Criterion. ARIMA models with smaller values of these parameters indicate to fit the data series better than the competing alternatives. The ARIMA models were developed using the SAS statistical toolbox (SAS online documentation 1999).

Model	Constant Estimate	Variance Estimate	Std Error Estimate	AIC	SBC	Number of Residuals
ARIMA(0,1,0)	0.135394	25.66067	5.065637	1120.239	1123.454	184
ARIMA(0,2,0)	0.255499	46.30518	6.804791	1215.507	1218.711	182
ARIMA(1,1,0)	0.162413	25.37821	5.037679	1119.194	1125.624	184
ARIMA(1,1,1)	0.069262	23.79662	4.878178	1108.341	1117.985	184
ARIMA(2,1,0)	0.245482	23.94656	4.893522	1109.496	1119.141	184
ARIMA(2,1,1)	0.1282	23.16539	4.813044	1104.375	1117.234	184

Table 2-1: Comparison of ARIMA Models

The power price data from the ISO New England website for the year 2000 were also used to develop a model for the expected hourly pattern within a day. The model correlating the hour of the day with the ratio of the hourly power price to average power price for the day is plotted in Figure 2-2.

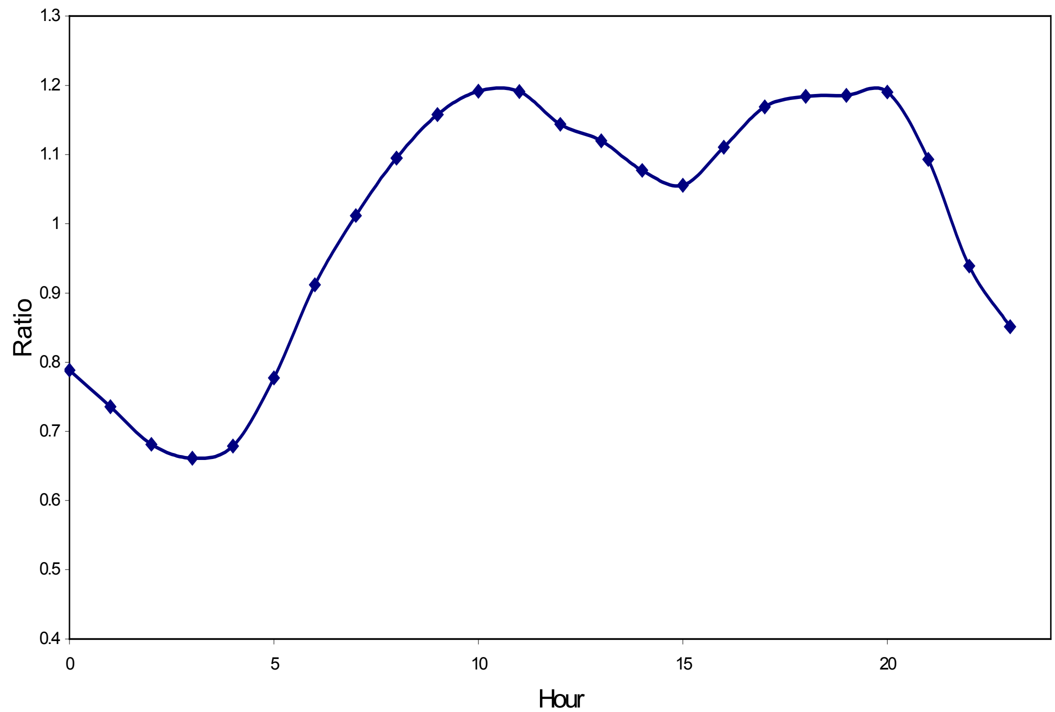

Figure 2-2. Hourly to Average Daily Price Ratio

2.4.2 FORECASTING RESULTS

Let us consider the specific example of predicting the power prices on July 18th and July 19th of 2000. Following the above procedure, first we predict the average daily prices based on previous data using the ARIMA(2,1,1) model. This results in average values of 37.43 and 39.64, respectively, compared to actual averages of 39.90 and 33.33. Then using the model for hourly to average daily price ratio, we predict the hourly power price fluctuations for these two days as shown in Figure 2-3.

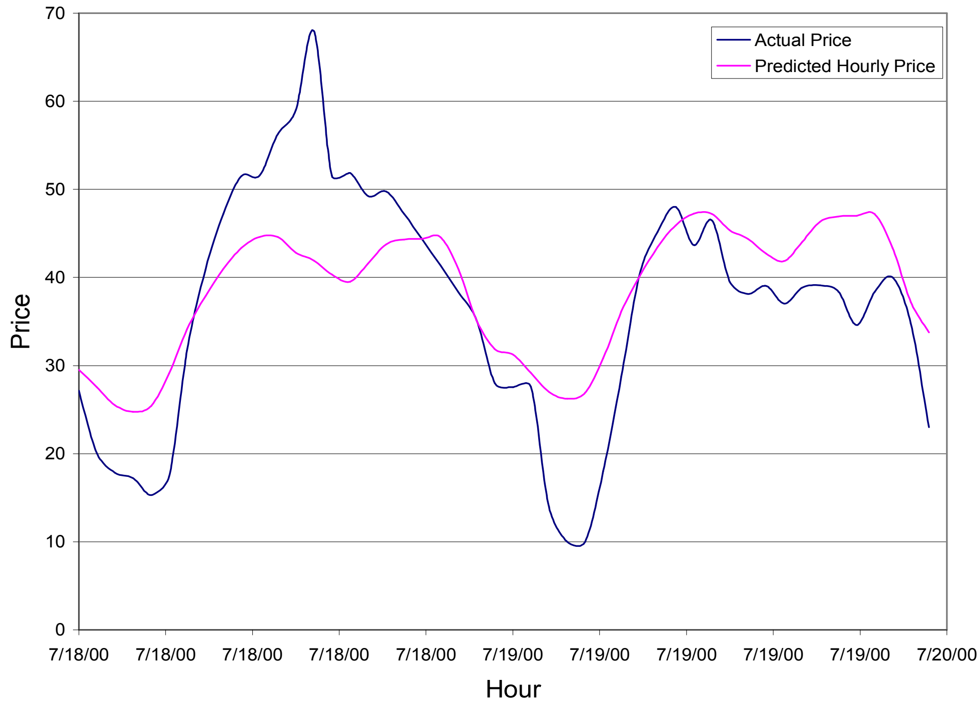

Figure 2-3: Actual and Predicted Prices for July 18th and July 19th

In addition to the predicted power prices, confidence intervals can be also estimated. Figure 2-4 and 2-5 show the 70% and 95% confidence limits. As expected by increasing the probability of capturing the price variability the range of uncertainty increases. The error between the predicted and the actual prices is plotted in Figure 2-6 and reaches a maximum of approximately 70%. One can argue from the error in power price prediction that the power prices cannot be predicted to any reasonable accuracy. However, we will illustrate in the next section that the power price prediction can be very effectively used in deciding the optimal operating schedule taking into account future price variability.

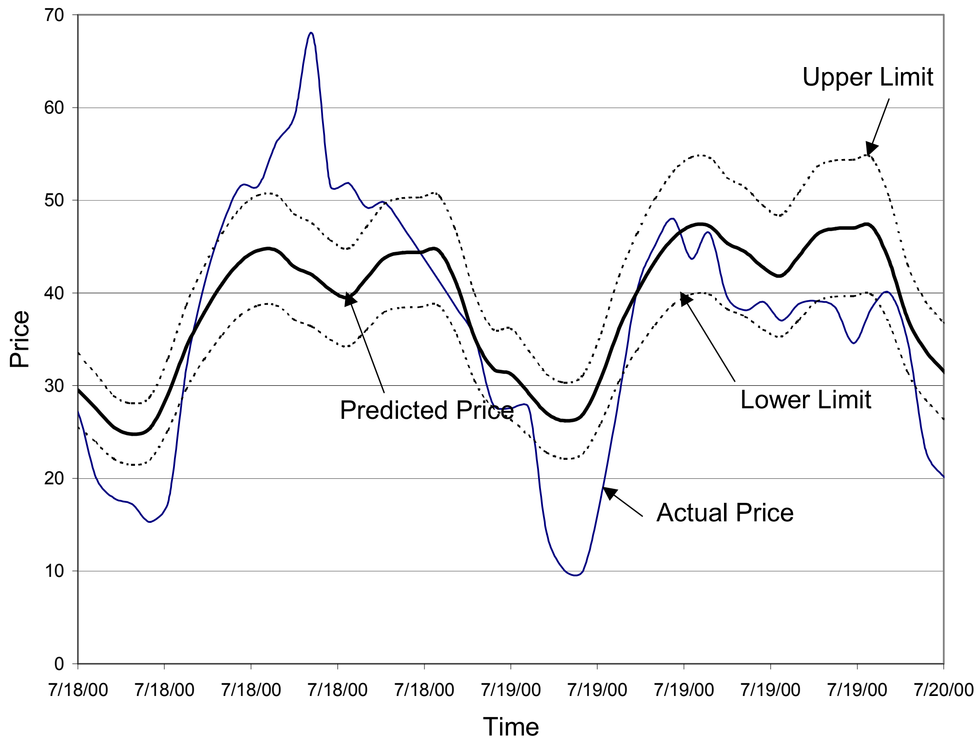

Figure 2-4: 70% Confidence Interval

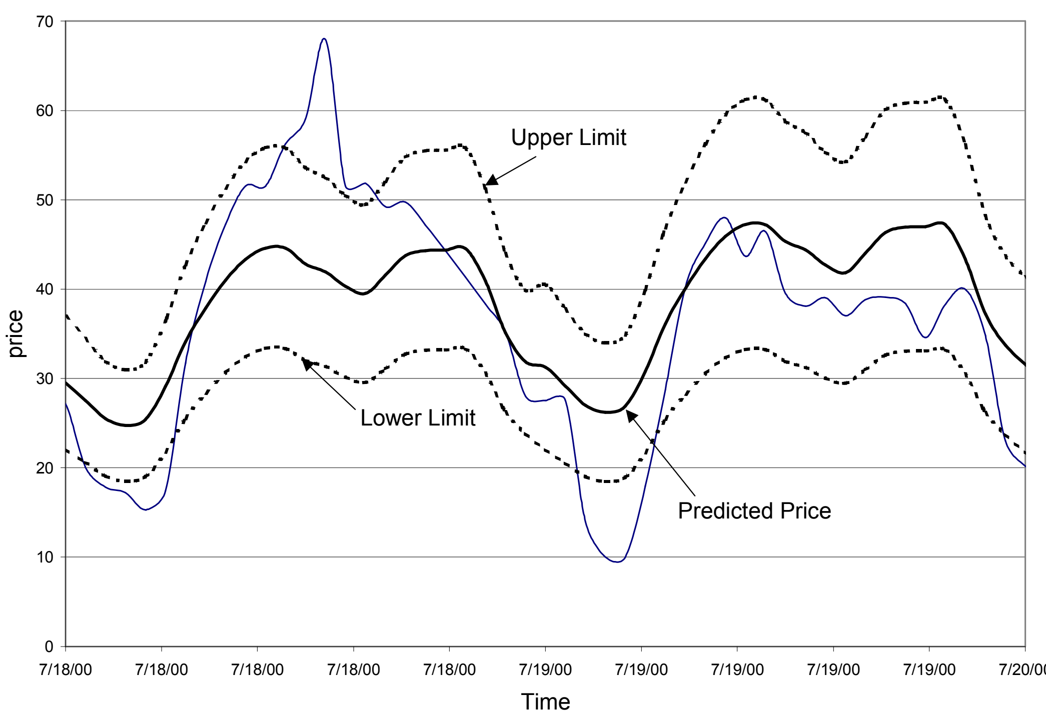

Figure 2-5: 95% Confidence Interval

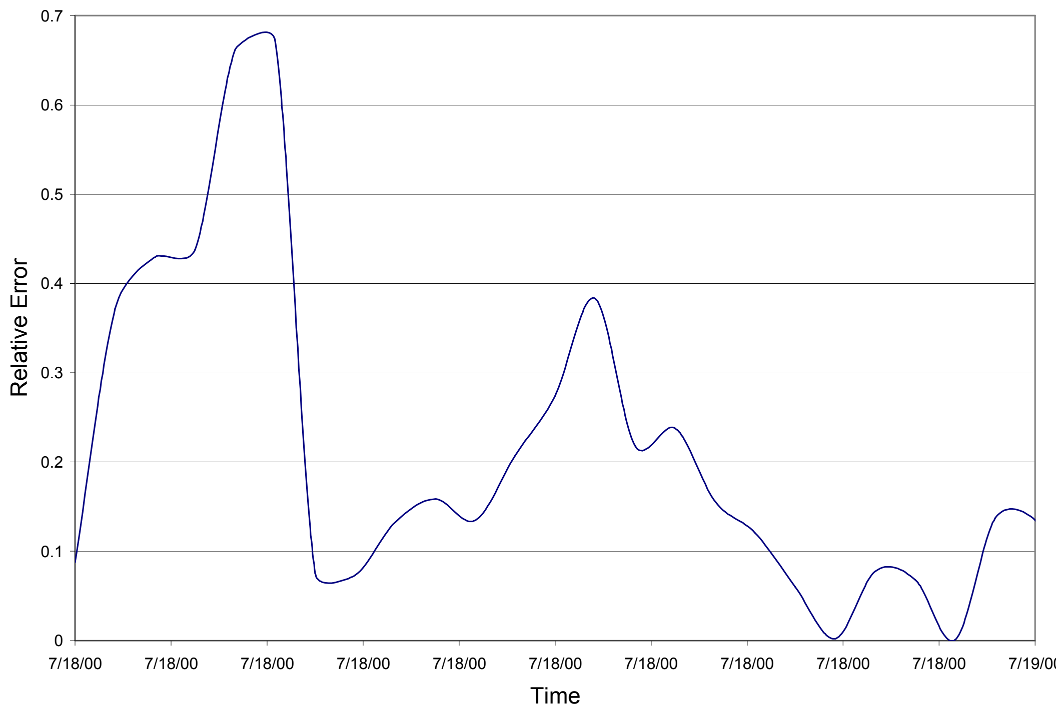

Figure 2-6: Relative Error in Power Price Prediction

2.4.3 FORECASTING MODEL IMPACT ANALYSIS

The following experiment was carried out to test the impact of the forecasted power prices. A period of 5 days was randomly selected from the historic data. The chosen days were from July 15th until July 20th of 2000. The problem was solved using the first stage only with actual power prices for July 15, 16, 17, 18 and 19. The results for this case are shown in Figure 2-7. The problem was again solved using the first stage only with actual power prices for July 15, 16, 17 and power prices forecasted using the methodology presented above for July 18 and 19. The results for this case are shown in Figure 2-8. By examining the results, especially in terms of operating mode schedule, it is found that the schedules obtained using the predicted and the actual prices have only one major difference. The schedule with the predicted prices for the last two days called for an additional period of operation in shut down mode during the first day. This situation occurred because extraordinary peaks cannot be precisely predicted. Specifically, a significant price spike occurred on July 18th (Figure 2-1) that was not predicted by the forecast. This caused the schedule decisions made using the predicted prices to be more optimistic. However, it is important to remember that it is equally possible for the power price to be over predicted and result in pessimistic schedule decisions.

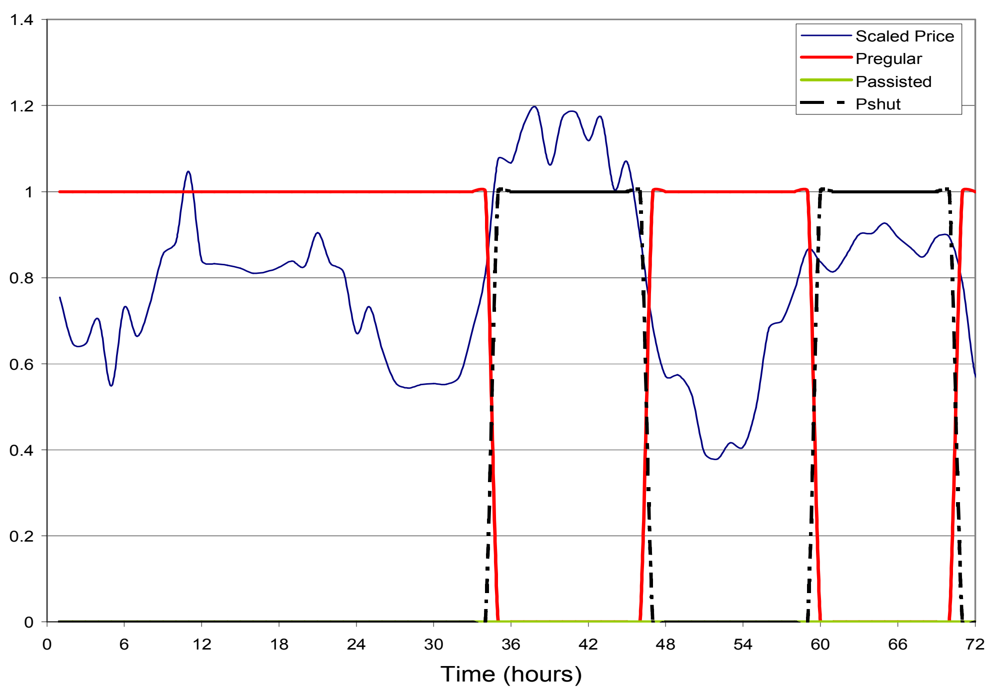

Figure 2-7: Optimal Operating Schedule for the First 3 Days When the Actual Power Price is Used for All 5 Days

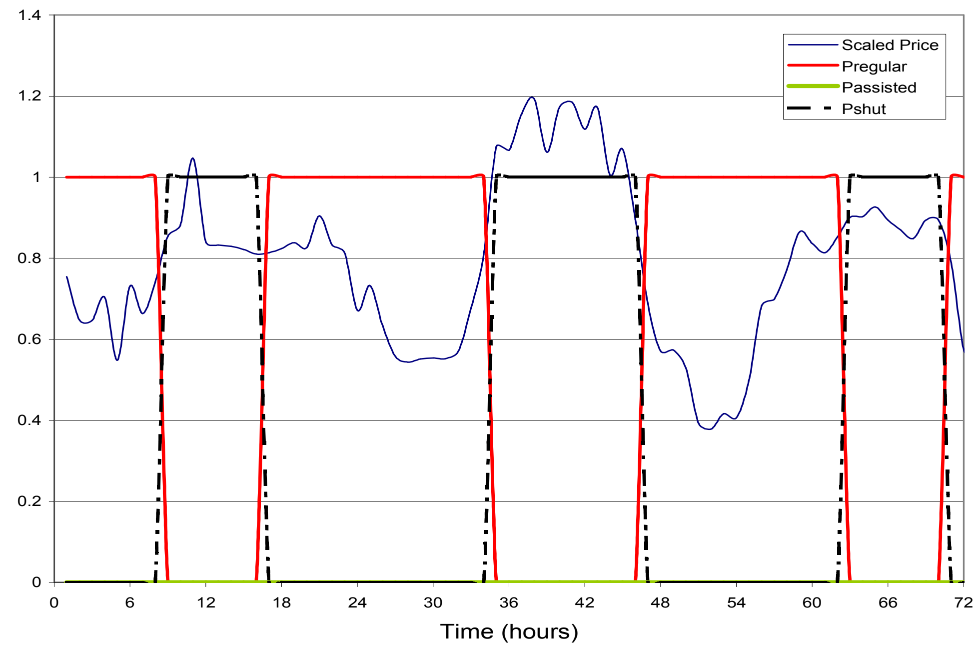

Figure 2-8: Optimal Operating Schedule for the First 3 Days When the Actual Power Price is Used for 3 Days and Forecasted Prices for the Last 2 Days

To evaluate the impact of the power price forecasts for more than two days in the future, we followed the same approach but used an eight-day horizon. The problem was solved using the first stage formulation and the actual power prices for eight days. These results are shown in Figure 2-9. The problem was then solved again using actual power prices for the first three days and predicted power prices for the last 5 days. These results are shown in Figure 2-10. In this case when more information about future prices is included into the model, the schedules obtained for the first three days are very close. This occurs even though actual prices were used in one case and forecasted prices were used in the other. The only minor difference in the first three days is one more hour of operation in shutdown mode when forecasted prices are used. The schedules for the last five days have similar amounts of time in shutdown mode but differ significantly in when the shutdowns occur. In this case, the schedule is more aggressive when the forecasted prices are used but it needs to be emphasized that this result cannot be generalized. It is equally likely that other cases will result in less optimistic schedules when forecasted prices are used.

The above analysis shows that using the power forecasting model developed here in the scheduling optimization problem generates reasonable schedules compared with using actual power prices. The forecasted prices also exhibit the expected behavior when the scheduling horizon is increased. The five-day scheduling horizon cases using forecasted information have more optimistic schedules compared to the eight-day horizon cases using forecasted information. As discussed earlier, this should be expected since there is no significant penalty for consuming products in storage until the lower storage limits are approached. Consequently, it is believed that the forecasting model is sufficiently accurate for the purpose of using it in a two stage stochastic optimization to better manage the risk due to uncertain future power prices.

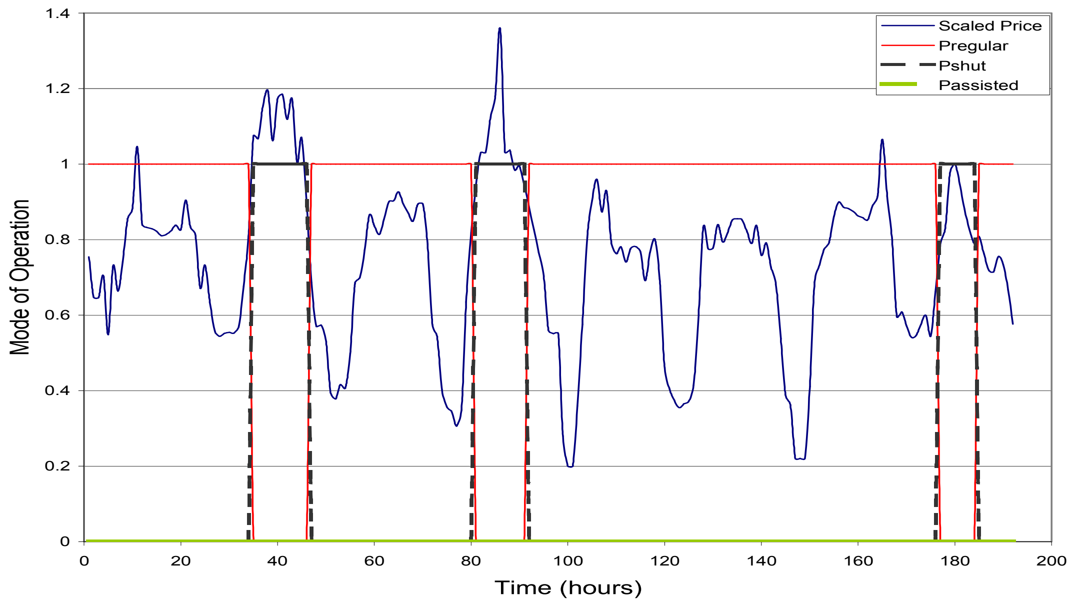

Figure 2-9: Optimal Operating Schedule When the Actual Power Price is Used for All 8 Days

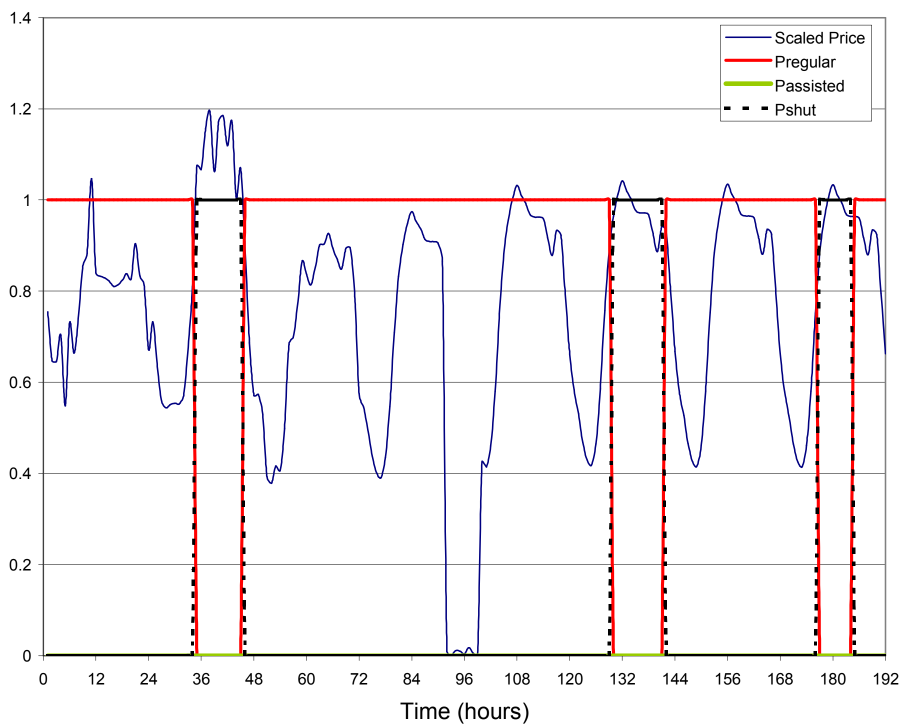

Figure 2-10: Optimal Operating Schedule When the Actual Power Price is Used for 3 Days and Forecasted Prices for the Last 5 Days

2.4.4 STOCHASTIC PROBLEM RESULTS

The results of presented the previous sections show that the use of predicted prices for future time period results in operation schedules close to the ones that would have chosen if the real prices were known. However, forecasting accuracy will typically decrease for larger prediction horizons.

To incorporate the uncertainty in future power price predictions, the two-stage stochastic programming approach will be utilized, which will allow to manage risk by optimizing the expected value of various power price forecast scenarios using different weights for each scenario rather than optimizing using a single uncertain power price forecast.

The most important part of the two stage stochastic optimization is defining the scenarios that define possible future power prices and the weights they should be given in the objective function. These scenarios should be selected to cover the expected range of variability of power price. In this work, the scenarios were chosen to be the values of the power price forecasts at the upper and lower confidence limits for a given probability. The weight for each scenario was chosen to be the value of the standard normal probability density function at the given probability normalized by the values for all scenarios. Additional work can also be incorporated following different probability distribution functions.

The specific cases considered here use deterministic power prices for the first three days followed by predicted power prices for the last five days. The data used are the same as those used to generate the results of Figure 2-10. Three scenarios are considered in the second stage, the predicted hourly power price values that correspond to the maximum, median and minimum values of the 85% confidence intervals. The resulting operation schedule is shown in Figure 2-11. The schedule is the same as the one obtained when only the mean predicted values are considered. However, the decision-making becomes more risk averse if higher confidence limits are considered for the future price prediction. The schedule results derived for the 95% confidence intervals of the power price forecast are shown in Figure 2-12. In this schedule, the switch into shutdown mode is for a shorter duration, eight hours versus ten. This indicates that the power price variability results in a potential negative impact of the maximum scenario greater than the potential positive impact of the minimum scenario. This result demonstrates that one can trade-off the risk they are willing to take versus the operating cost of production. The maximum allowable cost to avoid the risk is also quantifiable by examining the differences in the objective functions between cases.

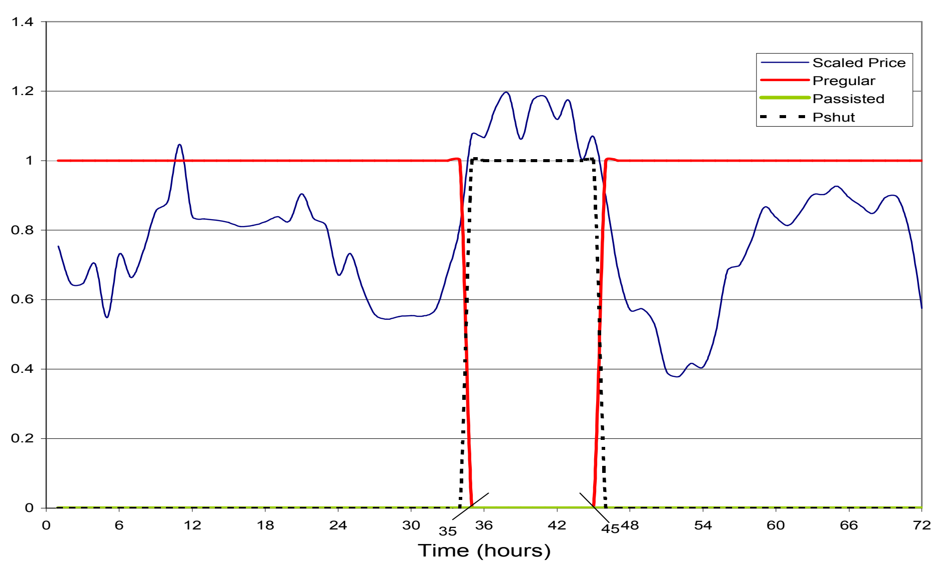

Figure 2-11: **Optimal Operating Schedule Using Two Stage Stochastic Programming Approach for 8 Days Covering 85% Confidence Intervals**

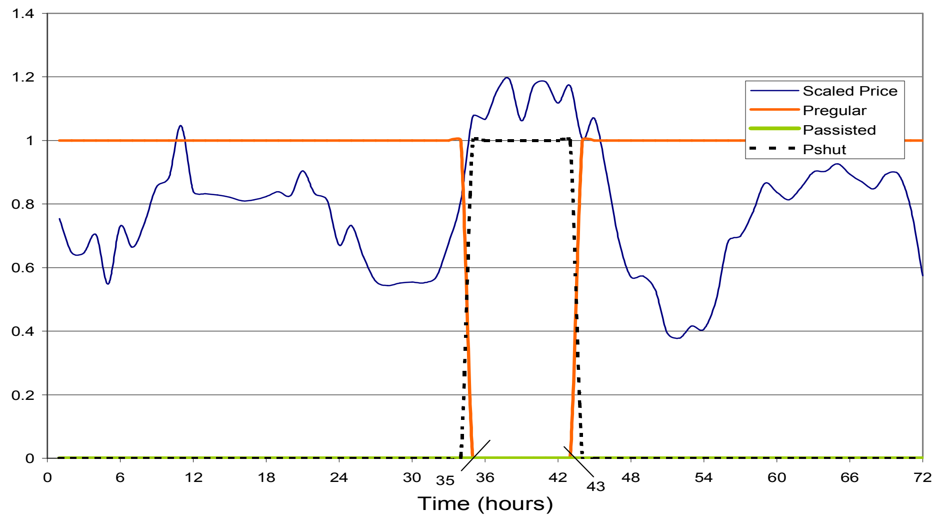

Figure 2-12: Optimal Operating Schedule Using Two Stage Stochastic Programming Approach for 8 Days Covering 95% Confidence Intervals

2.5 SUMMARY

Realistic short-term scheduling problems involve uncertainty since the operating specifications such as product demands and prices usually vary during the process operation. The two-stage stachastic programming approach proposed in this work establishes a stochastic optimization framework which gives emphasis on the influence of uncertain parameters on the current production plan. ARIMA model is utilized to provide scenarios of future value of uncertain parameters. These scenarios are considered in the second stage model that corresponds to the period when perfect information is absent. The production schedule for the current period thus is determined with reasonable consideration towards long term goal by incorporating the second stage into the objective function. The efficiency of this approach is illustrated with a realistic industrial problem.

A production optimization model is developed for an air separation plant which is subject to high fluctuation of energy price and demand. The production decisions achieved in this problem with the proposed approach are the same as the actual production decisions with perfect information, while those generated without considering variability of future energy price give an inferior schedule.

NOTATION

Variables

$x^{rm}_{LIN}(i)$	:	liquid N_2 production rate in regular mode at step i
$x^{am}_{LIN}(i)$	:	Net consumption rate of liquid N_2 in assisted mode at step i
$x^{rm}_{AIR}(i)$	:	Air production rate in regular mode at step i
$x^{am}_{AIR}(i)$	:	Air production rate in assisted mode at step i
$x^{rm}_{ARG}(i)$	:	Argon production rate in regular mode at step i
$x^{am}_{ARG}(i)$	:	Argon production rate in assisted mode at step i
$x^{rm}_{LOX}(i)$	:	Liquid O_2 production rate in regular mode at step i
$x^{am}_{LOX}(i)$	:	Liquid O_2 production rate in assisted mode at step i
$x^{rm}_{GOX}(i)$	:	Gaseous O_2 production rate in regular mode at step i
$x^{am}_{GOX}(i)$	:	Gaseous O_2 production rate in assisted mode at step i
$x^{rm}_{vent}(i)$	:	Rate of vent gas production in regular mode at step i
$x^{am}_{vent}(i)$	:	Rate of vent gas production in assisted mode at step i
$x_{liq}(i)$	:	equivalent liquid rate in regular mode at step i
$p^{rm}(i)$	:	binary variables corresponding to regular mode of operation at step i
$p^{am}(i)$	:	binary variables corresponding to assisted mode of operation at step i
$p^{sh}(i)$	:	binary variables corresponding to shutdown mode of operation at step i
$Sw^{am}(i)$	:	Switch variables that define a switch to assisted mode
$Sw^{sh}(i)$	:	Switch variables that define a switch to shutdown mode
$L_{LIN}(i)$	:	Level of liquid N_2 in storage tank at end of step i
$L_{LOX}(i)$	:	Level of liquid O_2 in storage tank at end of step i
$pw(i)$	:	Total power consumption at step i
$pw_{ASU}(i)$	:	Power consumed by Air Separation unit at step i
$pw_{NLU}(i)$	:	Power consumed by Nitrogen Liquefying unit at step i

pw_{misc}	:	Miscellaneous power consumed at step i (constant)
$x2^{rm}_{LIN,q}(i)$	:	liquid N_2 production rate in regular mode at step i in scenario q
$x2^{am}_{LIN,q}(i)$	:	Net consumption rate of liquid N_2 in assisted mode at step i in scenario q
$x2^{rm}_{AIR,q}(i)$	:	Air production rate in regular mode at step i in scenario q
$x2^{am}_{AIR,q}(i)$	:	Air production rate in assisted mode at step i in scenario q
$x2^{rm}_{ARG,q}(i)$	:	Argon production rate in regular mode at step i in scenario q
$x2^{am}_{ARG,q}(i)$	:	Argon production rate in assisted mode at step i in scenario q
$x2^{rm}_{LOX,q}(i)$	:	Liquid O_2 production rate in regular mode at step i in scenario q
$x2^{am}_{LOX,q}(i)$	:	Liquid O_2 production rate in assisted mode at step i in scenario q
$x2^{rm}_{GOX,q}(i)$	:	Gaseous O_2 production rate in regular mode at step i in scenario q
$x2^{am}_{GOX,q}(i)$	:	Gaseous O_2 production rate in assisted mode at step i in scenario q
$x2^{rm}_{vent,q}(i)$	:	Rate of vent gas production in regular mode at step i in scenario q
$x2^{am}_{vent,q}(i)$	:	Rate of vent gas production in assisted mode at step i in scenario q
$x2_{liq,q}(i)$	:	equivalent liquid rate in regular mode at step i in scenario q
$p2^{rm}{}_q(i)$	:	binary variables corresponding to regular mode of operation at step i in scenario q
$p2^{am}{}_q(i)$	:	binary variables corresponding to assisted mode of operation at step i in scenario q
$p2^{sh}{}_q(i)$	:	binary variables corresponding to shutdown mode of operation at step i in scenario q
$Sw2^{am}{}_q(i)$	:	Switch variables that define a switch to assisted mode in scenario q
$Sw2^{sh}{}_q(i)$	:	Switch variables that define a switch to shutdown mode in scenario q
$L2_{LIN,q}(i)$	:	Level of liquid N_2 in storage tank at end of step i in scenario q
$L2_{LOX,q}(i)$	:	Level of liquid O_2 in storage tank at end of step i in scenario q
$pw2_q(i)$	:	Total power consumption at step i in scenario q
$pw2_{ASU,q}(i)$	:	Power consumed by Air Separation unit at step i in scenario q

$pw2_{NLU,q}(i)$: Power consumed by Nitrogen liquefying unit at step i in scenario q

Parameters

$c(i)$: Price rates at step i

$c2(i)$: Price rates at step i at the second stage

$Q_{LIN}(i)$: Amount of liquid N_2 distributed at every time step i

$Q2_{LIN,q}(i)$: Amount of liquid N_2 distributed at every time step i in scenario q

V_{LIN} : Losses of liquid N_2 at every time step (constant)

$Q_{LOX}(i)$: Amount of liquid O_2 distributed at every time step i

$Q2_{LOX,q}(i)$: Amount of liquid O_2 distributed at every time step i in scenario q

V_{LOX} : Losses of liquid O_2 at every time step (constant)

$\alpha_{LIN}, \alpha_{LOX}, \alpha_{LIQ}, \alpha_{AIR}, \alpha_{GOX}$: minimal production rates for Liquid N_2, Liquid O_2, equivalent liquid, air and gaseous O_2, respectively

$\beta_{LIN}, \beta_{LOX}, \beta_{LIQ}, \beta_{AIR}, \beta_{GOX}$: maximum production rates for Liq N_2, Liq O_2, equivalent liquid, air and gaseous O_2, respectively

$S_{LIN}^{\min}$: Minimum level of Liquid N_2 allowed in storage tank

$S_{LIN}^{\max}$: Maximum level of Liquid N_2 allowed in storage tank

$S_{LOX}^{\min}$: Minimum level of Liquid O_2 allowed in storage tank

$S_{LOX}^{\max}$: Maximum level of Liquid O_2 allowed in storage tank

$\Delta_{LIN}^{low}, \Delta_{LIN}^{high}$: Factors accounting for random variations in level of liquid N_2 in storage tank because of random distribution of liquid N_2 i.e. V_{LIN}, over time period i; for lower level and upper level.

$\Delta_{LOX}^{low}, \Delta_{LOX}^{high}$: Factors accounting for random variations in level of liquid O_2 in storage tank because of random distribution of liquid O_2 i.e. V_{LOX}, over time period i; for lower level and upper level.

Coefficient Matrix and Other Parameters

$$A^r = \begin{bmatrix} 1 & 0 & 0 & 0 & -0.6 & 1 & 0 \\ 0 & 1 & 0 & 1 & -0.209 & 0 & 0 \\ 0 & 0 & 1 & 0 & -0.009 & 0 & 0 \\ 1 & 0.95 & 0.009 & 0 & 0 & 0 & -1 \end{bmatrix} \qquad A^{ass} = \begin{bmatrix} 0 & 0 & 0 & 0 & -0.54 & 1 \\ 0 & 1 & 0 & 1 & -0.209 & 0 \\ 0 & 0 & 1 & 0 & -0.009 & 0 \\ 1 & -1.3 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B^r = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad B^{ass} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$C^r = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad C^{ass} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \qquad F = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad V = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

$K^0_{ASU}=1800$; $K^1_{ASU}=-1.3$; $K^2_{ASU}=0.003$; $K^0_{NLU}=4000$; $K^1_{NLU}=1.0$; $K^2_{NLU}=0.02$;

α_{LIN} ,	:	250 KSCFH	β_{LIN}	:	480 KSCFH
α_{LOX}	:	10 KSCFH	β_{LOX}	:	160 KSCFH
α_{LIQ}	:	275 KSCFH	β_{LIQ}	:	770 KSCFH
α_{AIR}	:	550 KSCFH	β_{AIR}	:	660 KSCFH
α_{GOX}	:	70 KSCFH	β_{GOX}	:	85 KSCFH
$S^{\min}_{LIN}$	:	18000 KSCF	Δ^{low}_{LIN}	:	3000 KSCF
$S^{\max}_{LIN}$	:	61000 KSCF	Δ^{high}_{LIN}	:	4000 KSCF
$S^{\min}_{LOX}$	:	5000 KSCF	Δ^{low}_{LOX}	:	1200 KSCF
$S^{\max}_{LOX}$	:	20500 KSCF	Δ^{high}_{LOX}	:	160 KSCF

Q_{LOX} = 25 KSCFH , V_{LOX} = 150 KSCFH

P_{misc} = 325 kW;

CHAPTER 3

DECOMPOSITION APPROACHES FOR SHORT-TERM SCHEDULING PROBLEMS

Decomposition techniques for short-term scheduling problems are discussed in this chapter. These techniques include heuristics that can solve original problem to near-optimality in computational time of up to one order of magnitude reduction, and Lagrangean relaxation and Lagrangean decomposition that provide a stronger upper bound than LP relaxation problems. A solution framework for short-term scheduling based on these techniques is proposed with two examples.

3.1 INTRODUCTION

In the last decade, a number of papers have been devoted to solving short-term scheduling problems such as Pekny and Reklaitis (1998), Shah (1998), Pinto and Grossmann (1998) and others discussed in chapter 1. Among these work, mixed integer linear programming is a competitive alternative due to its general framework to incorporate realistic conditions. However, a major difficulty lies in the computational expense since complexity brought by number of integer variables and corresponding constrains requires exponential computation time. Therefore, solving MILP scheduling problems without simplification makes this approach prohibitive for industrial applications where it often fails to find a feasible solution (Pekny and Reklaitis, 1998). This trend can be observed in many approaches such as the uniform time discretization (Kondili et al.1993a), the time-slot based approach (Pinto and Grossmann, 1995), and continuous-time approaches (Zhang and Sargent, 1996; Mockus and Reklaitis, 1996; Ierapetritou and Floudas, 1998) when the problem size is increased to realistic scale.

In order to overcome this difficulty, many decomposition methods have been developed, which generally could be classified into algorithmic and heuristic. Following the early decomposition algorithm (Benders, 1962 and Dantzig, 1963), Olaf et al. (1993) summarized their generalizations, developing the variable decomposition and constraint decomposition approaches. New approaches in the former category include branch-and-cut algorithms (Johnson, Nemhauser and Savelsbergh, 2000) in which extra cuts are added to the MILP to strengthen its relaxation gap and combined constraint logic programming methods with MILP (Jain and Grossmann, 2001). While a large number of heuristics are presented to provide a fast solution for scheduling problems. Basset et al. (1996) presented a number of time-based decomposition approaches based on a discrete-time formulation. Khmelnitsky et al. (2000) proposed a time-decomposition approach to solve large-scale scheduling problems and obtain near-optimal solution based on the decomposition of the original problem into three sub-problems, namely sequencing, loading and timing problems. Boskma (1982) conducted a serial of experiments regarding time decomposition approaches, and the results showed that the division of the time in sub-periods applied in midterm planning model could impact substantially the outcomes of the model. Gupta and Maranas (1999) proposed a hierarchical Lagrangean relaxation procedure, along with an upper bound generating heuristic with an application on the solution of midterm planning problem. The algorithmic approaches usually suffer the weak relaxation and heuristics cannot guarantee the solution quality.

Lagrangean relaxation was presented by Held and Karp (1971) in their work on the traveling salesman problem. Based on the observation that mathematical models often consist relative easy constraints and hard constraints, Lagrangean relaxation replaces the hard constraints with a penalty term in the objective function, thus leaves the remaining problem relatively easy to solve. In addition, the solution from Lagrangean relaxation provides an upper bound to the original MILP problem assuming the objective is maximization. This upper bound is proven to be stronger than that provided by LP relaxation. The wide applications of Lagrangean relaxation are given, for example, in capacitated network design problem by Holmberg and Yuan (2000), pooling problem by Adhya and Sahinidis (1999) and planning problem by Gupta and Maranas (1999). Lagrangean decomposition (Guignard and Kim, 1987) is a special case of Lagrangean relaxation where identical copies of variables are created and dualized. The original problem is thus decomposed into separate sub-problems and has even stronger bound than Lagrangean relaxation. Lagrangean decomposition has been successfully applied to plant location problem (Marin and Pelegrin, 1998), transportation and scheduling problem (Equi et al., 1997) and routing problem (Rana and Vickson, 1991). A detailed review as well as an improved algorithmic development will be given in Chapter 6.

In this work, a hybrid method is developed where novel heuristics give efficient feasible solution to scheduling problems while Lagrangean relaxation and Lagrangean decomposition provide upper bound to indicate how good these solutions are compared to optimum. Various heuristic-based decomposition approaches are presented in section 3.2, whereas the application of Lagrangean relaxation and Lagrangean decomposition approach on the solution of short-term scheduling is presented in section 3.3. Finally in section 3.4 a systematic iterative procedure is proposed based on a lower bound obtained from a heuristic approach and an upper bound from Lagrangean relaxation and Lagrangean decomposition.

3.2 HEURISTIC-BASED DECOMPOSITION APPROACH

3.2.1 TIME DECOMPOSITION

Time decomposition methods focus on finding sequence and mass correlation between different time periods. The basic time decomposition method consists of dividing a large time horizon into several smaller sub-periods where the scheduling problem can be solved efficiently. The resulted sub-problems are then solved sequentially beginning with the earliest one. After the schedule of a period is determined, the production in that period is calculated and transferred to the initial inventory of the next period. This decomposition procedure is very simple and results in much smaller overall computational requirements than the original problem as shown in Table 3-1 with Example 1 in section 1.2.

Time Horizon	Number of Sub-periods	Number of Event Points ($1^{st} + 2^{nd} + 3^{rd}$)	Objective Function	CPU time (sec)
16 hrs	2 (8hrs)	11 (5 + 6)	3262.87	14.03 (0.44+13.59)
24 hrs	2 (12hrs)	16 (8 + 8)	5549.41	116.37 (110.30+6.07)
24 hrs	3 (8hrs)	17 (5 + 6 + 6)	5005.56	24.56 (0.44+13.59+10.53)

Table 3-1: Results for Example 1 with Time Decomposition

Note that the time decomposition gives rise to a sub-optimal solution. The reasons are i) it maximizes the production of all final products in each sub-period since it uses the same objective function as the original problem and thus all the processes that result in different products have to be executed in each sub-period which is not an optimum operating policy for the whole time horizon, ii) all tasks have to end-up exactly at the end of each period, which is an extra constraint in the scheduling problem. To avoid this interruption of operation at the end of each sub-period the following "smoothing" procedure is applied.

I. All the tasks that finish at the end of the previous period are determined and the earliest starting time among them is denoted as EST. This is then used as new starting time of the next period with time horizon of $H_{sub} + (H_{sub} - EST)$, where H_{sub} is the pre-defined time horizon of sub-problem.

II. All the tasks that finish after EST are classified into three categories (Figure 3-1 for an illustrating case):

a. the tasks that start before EST and finish before the end of the current period H_{sub} (task 1 and task 4),

b. the tasks that start after EST and finish before the end of the current period H_{sub} (task 5), and

c. the tasks that start at or after EST and finish at the end of the current period H_{sub} (task 2, task 3 and task 6).

III. For the tasks in the first category their starting times and processing times are fixed allowing them to continue into the next period, whereas for the tasks in the second category only their starting times are fixed. For tasks in the third category nothing is fixed and thus the new schedule of the next period determine their starting times and batch sizes.

The sub-problem that corresponds to the next period is solved with starting time equal to EST.

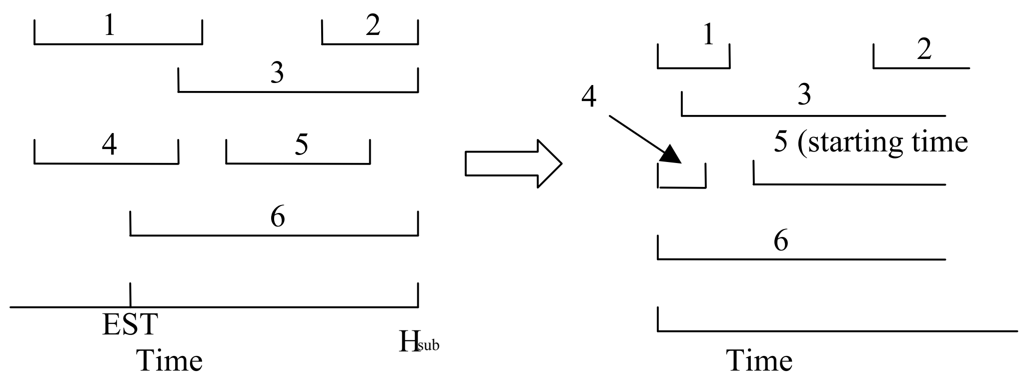

Figure 3-1: Illustration of Smoothing Technique

This method adjusts the processes at the end of each sub-period so that tasks can continue across the periods resulting in smoother plant operation and thus improves quality of solution in terms of the objective function value as shown in Table 3-2.

Time Horizon	Number of Sub-periods	Number of Event Points ($1^{st} + 2^{nd} + 3^{rd}$)	Objective Function	CPU time (sec)
16 hrs	2 (8hrs)	11 (5 + 6)	3550.42	2.7 (0.44+2.26)
24 hrs	2 (12hrs)	16 (8 + 8)	5918.42	492.55 (110.30+382.25)
24 hrs	3 (8hrs)	17 (5 + 6 + 6)	5484.96	4.34 (0.44+2.26+1.64)

Table 3-2: Results for Example 1 with Smoothing Technique in Time Decomposition

However, the solution is still sub-optimal since by using the objective of production maximization at each sub-period, the schedule increases the production in each smaller time horizon, which may result in a sub-optimal solution due to inability to perform larger batch sizes. To overcome this shortcoming the following method is proposed.

3.2.2 REQUIRED PRODUCTION METHOD

The basic idea of the required production method is to change the problem target at each sub-period aiming the production of the necessary intermediates in order to produce the final products. Therefore, large batches of intermediates can be produced without been completely consumed to produce final products during the same sub-period. As a key step in this method, the material requirement table is utilized to determine the necessary amount of intermediates. The material requirement table is created based on the mass balance of the whole process and determines the necessary units of intermediates in order to produce one unit of any specific product. Based on the STN representation the following cases can be found (Figure 3-2) in order to calculate the material requirement table.

1) Normal task line, where in order to determine the material requirements, one unit of final product is assumed and the necessary intermediates are calculated based on recipe.
2) Bifurcation task line, where it is necessary to compare the required amounts for task 2 and task 3, and take the larger value to determine the required amount of materials in task 1. This is because in this case the State Task Network (STN) expresses that two kinds of material are produced from Task 1 since Task 2 and

Task 3 are using two state-nodes from Task 1. Thus, the required production for Task 1 could be calculated from material in either Task 2 or Task 3. In order to satisfy both requirements, the larger one is picked.

3) Loop task line, where the amount produced by the loop task should be considered. For the case where the last batch of the loop task does not contribute to the next task, the necessary adjustment should be made.

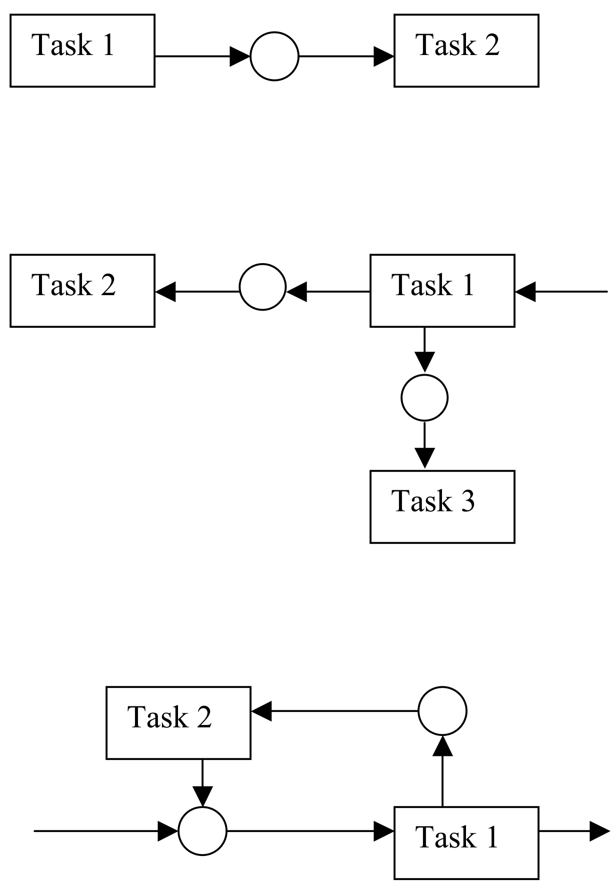

Figure 3-2: Illustration of STN Basic Structures

The material requirement table for Example 1 is shown in Table 3-3.

State s	Ratio Calculated from P1	Ratio Calculated from P2
P1	1	0.5926
P2	1.6875	1
HA	1	0.5926
Int BC	1.5	0.8889
Int AB	1.6875	1
Impure E	1.875	1.1111

Table 3-3. Material Requirement Table for Example 1

Given specific production of all final products, the amounts of intermediates, which are denoted as 'required production', are calculated by multiplying the ratio in the material requirement table and the amount of final product. Then, instead of maximizing production of the final products at each sub-period, the objective is to produce the required amounts for all final products and intermediates. In mathematical terms two additional constraints are added.

Production Constraints

$$P(s,n) = \sum_{i \in I_s} \left(\rho_{si}^{p} \sum_{j \in J_i} B(i, j, n) \right) \qquad \forall s \in S, n \in N \tag{3-1}$$

where *P(s,n)* denotes the production of state s at the event point n. Constraints (3-1) are used to calculate the amount of state *s* that is produced at each event point.

Production Requirement Constraints

$$\sum_n P(s,n) \geq P_s^{res} - slack(s) \qquad \forall s \in S, n \in N \tag{3-2}$$

where $P^{res}(s)$ denotes the required production of state s in order to fulfill the production of final products. Constraints (3-2) represent that the amount of state *s*, which has been produced during the whole period, has to be greater or equal to P_s^{res}.

Objective function

$$\min \quad z = \sum_s slack(s) \tag{3-3}$$

The objective function corresponds to the minimization of the slack variables introduced in the production requirement constraints. Therefore in each period the solution maximizes the production of all the required intermediates as well as final products. It should be pointed out that constraints (3-1)-(3-3) don't change any mass balance or sequence constraints defined in section 1.1. The required production method is usually applied together with time decomposition.

3.2.3 COMBINATION OF SMOOTHING TECHNIQUE AND REQURIRED PRODUCTION METHOD

In this section, the required production method is considered simultaneously with the time decomposition with smoothing technique. For the scheduling problem with the objective to minimize production, where $P^{res}(s)$ is not given, the following approach is proposed. i) Consider specific volumes of final products, using the solution of the relaxed

LP relaxation problem or Lagrangean relaxation (LR) or Lagrangean decomposition (LD) as it will be discussed in the next section. ii) Calculate the required production for intermediates. iii) Except the last sub-period, set (3-3) as the objective function for other sub-periods and apply the decomposition technique described in 3.2.1. iv) Set (1-15) as the objective function for the last sub-period. v) Fix the binary variables and solve the original problem. It should be noted that for step (iii), a positive objective value means that P_s^{res} is sufficiently large, whereas a zero objective function requires an additional iteration with an increased P_s^{res}.

The results for Example 1 are listed in Table 3-4. Compared with the results of time decomposition without the required production constraints (Table 3-1 and Table 3-2), it should be noticed that generally better quality objective values are achieved, especially when applied to problem with large time horizon that needs to be decomposed into multi-periods. This is because better connection between the sub-problems is achieved compared to time decomposition method and thus the solution of each sub-period involves some large batch sizes independent of whether or not production of all different final products are achieved in that sub-period.

Time Horizon	Number of Sub-periods	Number of Event Points ($1^{st} + 2^{nd} + 3^{rd}$)	Objective Function	CPU time (sec)
16 hrs	2 (8hrs)	11 (5+6)	3373.25	18.97 (16.14+2.83)
24 hrs	2 (12hrs)	14 (6+8)	5875.96	30.77 (22.93+7.84)
24 hrs	3 (8hrs)	16 (5+5+6)	5662.17	27.12 (14.81+4.99+7.32)

Table 3-4: Results for Example 1 Using Required Production Method with Smoothing Technique

The advantage of this approach is illustrated better when larger scale problems are considered. The same example is solved here but considering a larger time horizon of 48 hours. Following this approach, first the linear relaxation of the original problem (LP) was solved using 26 event points. The necessary amounts of products were calculated based on the LP result, and the required production method and smoothing technique were used with decomposing the whole time horizon into 3 periods of 16 hours each. The final step involves refinement of the schedule by fixing the binary variables with positive value and resolving the original scheduling problem of 48 hours. The result obtained by this approach is shown in Table 3-5 and compared with the solution of the original problem and the solution of the overall problem by applying cyclic scheduling mode which repeatedly executes the same operation schedule for a smaller time horizon. As illustrated in Table 3-5, the proposed approach generates an improved objective function value reducing the computational time in more than one order of magnitude.

Solution Approach		Objective Value	Number of Event points	CPU time (sec)
Best solution by solving original problem	48 hours	12637.60	24	10000.47* (3717.96**)
Cyclic mode	8×6 hours	8989.14 (1498.19×6)	30 (5×6)	0.47
	12×4 hours	10631.60 (2657.90×4)	32 (8×4)	107.11
	16×3 hours	11211.3 (3737.10×3)	27 (9×3)	177.93
	24×2 hours	12039.22 (6019.16×2)	26 (13×2)	10000.12* (9394.66**)
Proposed approach		12710.98	25	804.95

* Default time to stop the calculation

** Computational time to find this solution.

Table 3-5: Computation Results for 48 Hours Schedule of Example1

The same approach was applied to Example 2. In this example, four products are produced through eight tasks from three feeds. There are nine intermediates in the system. In all, six different units are required for the whole process. The STN representation for this process is shown in Figure 3-3, and the required data is presented in Table 3-6. The processing times are allowed to vary 1/3 around the mean values (τ_{ij}^{mean}). The objective is to maximize the profit with the given time horizon. The computational results of the proposed approach, original scheduling problem and cyclic mode approach are listed in Table 3-7, pointing to the fact that the proposed approach performs very well since a very good solution in terms of objective value is obtained in reasonably short time.

Unit	Capacity	Suitability	Mean Processing Time (τ_{ij}^{mean})
Unit 1	1000	Task 1	1.0
Unit 2	2500	Tasks 3,7	1.0
Unit 3	3500	Task 4	1.0
Unit 4	1500	Task 2	1.0
Unit 5	1000	Task 6	1.0
Unit 6	4000	Tasks 5,8	1.0
State	**Storage Capacity**	**Initial Amount**	**Price**
Feeds 1,2,3	unlimited	0.0	0.0
Intermediate 4	1000	0.0	0.0
Intermediate 5	1000	0.0	0.0
Intermediate 6	1500	0.0	0.0
Intermediate 7	2000	0.0	0.0
Intermediate 8	0	0.0	0.0
Intermediate 9	3000	0.0	0.0
Products 1,2,3,4	unlimited	0.0	18,19,20,21
Feeds 1,2,3	unlimited	unlimited	0.0

Table 3-6: Data for Example 2

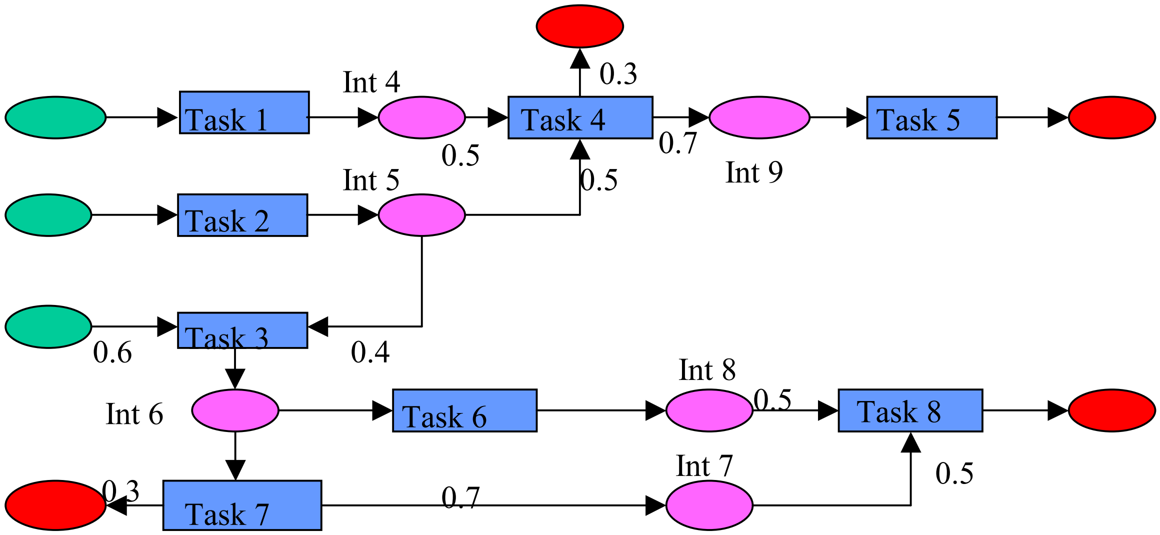

Figure 3-3: State Task Network for Example 2

Solution Approach		Objective Value	Number of Event points	CPU time (sec)
Best solution by solving original problem	48 hours	2193616.47	37	10001.24*
Cyclic mode	8×6 hours	1517849.88 (252974.98×6)	54 (9×6)	2552.85
	12×4 hours	1785114.66 (446278.67×4)	48 (12×4)	10000.14* (8592.72**)
	16×3 hours	1930541.304 (643513.768×3)	45 (15×3)	10000.22* (4087.29**)
Proposed approach		2169426.95	44	2148.05

* Default time to stop the calculation

** Computational time to find this solution.

Table 3-7: Computation Results for 48 Hours Schedule of Example 2

3.2.4 RESOURCE-BASED DECOMPOSITION

In this section two different decomposition methods are proposed based on the partition of various scheduling resources. The first one is based on the decomposition of the event points whereas the second one decomposes and treats separately the units involved in the production.

3.2.4.1 EVENT POINT DECOMPOSITION

Since the main reason of the excessive computational complexity of large-scale scheduling problem is due to the increased number of events occurring during the time horizon under consideration, the basic idea of this proposed method is to successively increase the number of events while fixing those that have already used. In particular, this approach includes the following steps. First, a small number of event points are used and the scheduling problem is solved. Then the task sequence is fixed based on the solution of the problem with the limited event points, and additional event points are considered for the problem to be resolved. Since fixing the task sequence decreases the number of binary variables, fast solution can be achieved. The iterations terminate when no further improvement of the objective function can be achieved. The results for example 1 are shown in Table 3-8.

Time Horizon	Iteration Number	Number of Event Points	Objective Function	CPU time (sec)
16 hrs	1	5	1760.00	0.12
	2	6	1920.00	0.08
	3	7	2253.33	0.17
	4	8	2253.33	0.16
	5	9	3354.75	0.51
	6	10	3563.27	0.37
	7	11	3563.27	0.60
		Total CPU		**2.01**
24 hrs	1	5	1760.00	0.13
	2	6	1920.00	0.08
	3	7	2253.33	0.17
	4	8	2703.33	0.14
	5	9	3873.33	0.16
	6	10	4787.08	0.22
	7	11	4800.42	0.24
	8	12	5250.42	0.28
	9	13	5335.74	0.99
	10	14	5649.32	0.65
	11	15	5649.32	1.14
		Total CPU		**4.2**

Table 3-8: Results for Example 1 Using Event Point Decomposition

This method performs very well in terms of the required computational time but shares the same disadvantage of the previous approaches regarding the optimality of the solution. Optimality is not guaranteed for the original problem since once the event points are fixed, the solution cannot accommodate schedules that require different allocation of the event points.

3.2.4.2 CRITICAL UNIT IDENTIFICATION

This approach can be used very efficiently when a bottleneck unit exists. For these cases, the critical unit is forced to work continuously during the whole time horizon. This requirement is translated to the following constraints.

Full Utility Constraint

$$\sum_{i \in I_j} \sum_{n} \left(T^f(i,j,n) - T^s(i,j,n) \right) = H, \quad \forall j \in \{Set\ of\ critical\ units\} \tag{3-4}$$

To identify a bottleneck unit one may check the processing capability of each unit together with the process recipe. The results for example 1 of 24 hours using each of the reactors as a critical unit are shown in Table 3-9. These results are possibly still a sub-optimal solution since no verification can be achieved using increased number of event points. However, a better solution than the one generated by solving the original MILP problem has been obtained with less CPU time.

Time Horizon	Critical Unit	Number of Event Points	Objective Function	CPU time (sec)
16 hrs	Reactor 1	9	3737.10	224.40
	Reactor 2	9	3737.10	182.10
24 hrs	Reactor 1	13	6022.57	40045.08
	Reactor 2	13	6035.49	21855.05

Table 3-9: Results for Example 1 Using Critical Unit Identification Method

The proposed decomposition approaches are case dependent and cannot guarantee optimality of the scheduling obtained. However the proposed approaches provide a number of alternatives in order to obtain solutions of large-scale scheduling problems when the available MILP solvers fail to generate even feasible solution to the original problem. In addition, the proposed heuristic-based approaches will be utilized in order to provide a lower bound to the solution framework proposed in section 3.4.

3.3 LAGRANGEAN APPROACHES

3.3.1 LAGRANGEAN RELAXATION (LR)

Lagrangean relaxation provides a systematic way of obtaining upper bounds for specific classes of complex large-scale problems. These problems exhibit the characteristic that the removal of a set of constraints results in a problem much easier to solve, either decomposable or with a special structure (Geoffrion, 1974 and Fisher, 1981). A detailed description of Lagrangean relaxation and Lagrangean decomposition is given in chapter 6. However, a brief introduction is also provided in this section for the sake of the continuity of the thesis. Consider the following optimization problem:

$$\begin{aligned} Z &= \max \; cx \\ s.t.\; Ax &\le b \\ Dx &\le e \\ x &\ge 0,\; x \;\; integral \end{aligned} \tag{3-5}$$

The Lagrangean relaxation of this problem with respect to the first set of constraints has the following form:

$$\begin{aligned} Z_D(u) &= \max \; cx + u(b - Ax) \\ s.t. \quad & Dx \le e \\ & x \ge 0, \quad x \;\; integral \end{aligned} \tag{3-6}$$

Assume x^* is the optimal solution of the original problem, then the following relation holds:

$$Z_D(u) = cx^* + u(b - Ax^*) \geq cx^* = Z \tag{3-7}$$

$u>0$ since it corresponds to the Lagrange multiplier vector of the constraints $Ax \leq b$. Consequently $Z_D(u)$ provides an upper bound to the original problem. In order to get the tightest bound, certain u should be found such that:

$$Z_D = \min_u Z_D(u) \tag{3-8}$$

Subgradient method is typically used to optimize $Z_D(u)$ over u. This involves the following updating procedure:

1) Calculate the step size

$$t_k = \frac{\lambda^k \left(Z_D(u^k) - Z^*\right)}{\left\| b - Ax^k \right\|^2} \tag{3-9}$$

where λ^k is a scalar usually $0<\lambda^k \leq 2$ and Z^* is a lower bound on Z_D. Most of the reported work provides evidence that the performance of Lagrangean relaxation is quite sensitive to the heuristic choice of the step size sequence t_k. A common empirical way is to half λ and half the number of iterations when there is no increase in $Z_D(u)$. Z^* is typically obtained by applying a heuristic to the original problem. These heuristics may involve one of the approaches discussed in section 3.2.

2) Update u

$$u^{k+1} = u^k + t_k(b - Ax^k) \tag{3-10}$$

$$u = \max\left(0, u^{k+1}\right)$$

Since the optimal solution for $Z_D = \min_u Z_D(u)$ is not guaranteed by using subgradient method, the computation for LR problem usually terminates after a certain number of iterations. In the following illustrating examples, it is shown that LR provides better bound than linear relaxation, however it is not guaranteed to provide a bound at all cases.

The scheduling problem formulation (1-1)-(1-15), involves equality constraints representing mass balances and duration constraints, and inequality constraints expressing capacity limitation, allocation constraints, sequence constraints, and horizon constraints. To investigate the quality of LR each of these sets of constraints was selected to be relaxed in the objective function.

The results for Example 1 using a time horizon of 8, 16 and 24 hours are tabulated in Table 3-10. The iteration criterion used is to half λ when the objective value fails to drop within 5 iterations. The stopping criterion is t_k < 0.00001 or the total iteration number reaches 100. When the horizon constraints are relaxed, LR problem stops because the number of iterations reaches 100, while for other constraints, the iterations terminate with t_k reaching the specific tolerance.

Note that all the LR problems for time horizon of 8 and 16 hours except the ones where the mass balances are relaxed converge and result in good upper bounds to the overall problem. The upper bound provided by the relaxation of allocation constraint is the tightest one, but too expensive in terms of computation time for time horizon of more than 16 hours compared to the original problem solution time. The results also indicate that the maximum capacity constraints are the critical ones in the model. Once we relax it, the optimization problem can be solved rather easily. The mass balances appear as good candidate for relaxation since they connect the inventory, batch size and define the mass flow. However, if no storage limitations are posed on products, *ST(s,n)* and *d(s,n)* are free variables after relaxing the mass balances, thus resulting in unbounded problem. For the case of 24 hours the computational requirements are still very high for all cases except when the maximum capacity constraints are relaxed which reinforce the above statement about the crucial nature of these constraints.

To investigate further the quality of the bounds obtained by relaxing various constraints, only one specific constraint was relaxed selected randomly from the set of binding constraints generated by the linear relaxation problem (LP). The results are shown in Table 3-11. Note that the relaxation of only one of the binding constraints results in the optimal solution requiring a very small computational time for time horizons of 8 and 16 hours. However, for 24 hours no solution can be obtained within the imposed limit of 3000 CPU sec.

Constraint Set	8 hours 5 event points		16 hours 9 event points		24 hours 13 event points	
	OBJ	CPU (sec)	OBJ	CPU (sec)	OBJ	CPU (sec)
Maximum Capacity	1781.47	18.8	4371.88	47.6	7227.45	14.6
Duration	1765.25	12.4	4361.56	31.7	---------	> 3000
Allocation	1498.32	33.7	3783.73	19.92	---------	> 3000
Horizon	1654.68	29.0	4071.56	17.3	---------	> 3000
Sequence (Different tasks in different units)	1567.74	26.2	4200.86	850	---------	> 3000
Mass Bal.	UNBD	------	UNBD	------	UNBD	------

Table 3-10: LR Results for Example 1 Relaxing the Whole Set of Constraints

Constraint Set	8 hours 5 event points		16 hours 9 event points		24 hours 13 event points	
	OBJ	CPU (sec)	OBJ	CPU (sec)	OBJ	CPU (sec)
Maximum Capacity	1498.19	0.45	3737.10	285.42	---------	> 3000
Horizon	1498.19	1.49	3737.10	330.24	---------	> 3000
Sequence (Different tasks in different units)	1498.19	2.49	3737.10	259.69	---------	> 3000

Table 3-11: LR Results for Example 1 Relaxing Only One Constraint

3.3.2 LAGRANGEAN DECOMPOSITION (LD)

Lagrangean decomposition has been proposed by Guignard and Kim (Guignard et al., 1987) as a extension of Lagrangean relaxation by relaxing a set of variables such that the resulting model becomes decomposable into two or more sub-problems that are easier to solve. Consider the following general model:

$$\begin{aligned} & \max f^T x + d^T(y^1+y^2) \\ & s.t.\ Ax + B^1 y^1 \le b^1 \\ & \quad Cx + B^2 y^2 \le b^2 \\ & \quad x \in X,\ y^1, y^2 \in \{0,1\} \end{aligned} \tag{3-11}$$

The model is decomposable in terms of the integer variables y^1, y^2 but linked through the continuous variable x. The application of Lagrangean decomposition can be achieved by creating identical copy of variable x in both two sets of constraints and relaxing the equality constraint as follows:

$$\begin{aligned} Z^D(u) = \ & \max f^T x + d^T(y^1+y^2) + u(z-x) \\ & s.t.\ \ Ax + B^1 y^1 \le b^1 \\ & \quad Cz + B^2 y^2 \le b^2 \\ & \quad x,z \in X,\ y^1, y^2 \in \{0,1\} \end{aligned} \tag{3-12}$$

The resulting problem can be decomposed to the following sub-problems that can be solved independently:

$$\begin{array}{lcl} \max f^T x + d^T y^1 - ux & + & \max\ d^T y^2 + uz \\ s.t.\ \ Ax + B^1 y^1 \le b^1 & & s.t.\ \ Cz + B^2 y^2 \le b^2 \\ x \in X,\ y^1 \in \{0,1\} & & z \in X,\ y^2 \in \{0,1\} \end{array} \tag{3-13}$$

Lagrangean decomposition is typically used to obtain good upper bound to the original problem. The tightest bound corresponds to the solution of the following non-differentiable optimization problem:

$$Z^{LD} = \min_{u} Z^{D}(u) \tag{3-14}$$

This minimization problem is difficult and time-consuming to solve. Thus similar to Lagrangean relaxation, Lagrangean decomposition employs an iterative technique such as subgradient method to update the Lagrangean multiplier *u* and stops after a certain pre-defined criteria are satisfied. For realistic size problems, subgradient method is not always performing well. An extended research on Lagrangean decomposition is conducted in chapter 6 where an alternative method of updating Lagrangean multipliers is developed. This new method improves the Lagrangean objective function at each iteration and thus has a promising application on scheduling problems.

To apply the Lagrangean decomposition approach to the solution of scheduling problem the crucial question is to identify constraints to be decomposed and variables to be duplicated such that the model becomes easily decomposable and thus easier to solve. From the basic description of the mathematical formulation of scheduling problem it can be noticed that there are two separable sets of constraints, the mass related constraints consisting of mass balances, capacity constraints, and the time related constraints consisting of the duration and sequence constraints. These two sets of constraints are only connected through the binary variables *wv(i,n)* and the batch sizes *B(i,j,n)*. Thus we propose the relaxation of these sets of variables that give rise to the following sub-problems:

Sub-problem 1:

$$\max \sum_{s}\sum_{n} price_s \times d(s,n) - \sum_{i}\sum_{j}\sum_{n}(u(i,j,n)\times B1(i,j,n) + v(i,n)\times wv1(i,n))$$

subject to:

Allocation Constraints

$$\sum_{i\in I_j} wv1(i,n) = yv(j,n), \qquad \forall j \in J, n \in N$$

Capacity Constraints

$$B1(i,j,n) \ \le V_{ij}^{\max} wv1(i,n), \qquad \forall i \in I, j \in J_i, n \in N$$

2.2 Material Balances

$$ST(s,n) \ = ST(s,n\text{-}1) - d(s,n) \ + \sum_{i\in I_s}\rho_{si}^{p}\sum_{j\in J_i} B1(i,j,n\text{-}1) - \sum_{i\in I_s}\rho_{si}^{c}\sum_{j\in J_i} B1(i,j,n), \quad \forall s \in S, n \in N$$

Storage Constraints

$$ST(s,n) \le \ ST_s^{\max}, \qquad \forall s \in S, n \in N$$

Demand Constraints

$$\sum_{n\in N} d(s,n) \ge \ r_s, \qquad \forall s \in S$$

Sub-problem 2:

$$\max \sum_{i}\sum_{j}\sum_{n}(u(i,j,n)\times B2(i,j,n) + v(i,n)\times wv2(i,n))$$

subject to:

Allocation Constraints

$$\sum_{i \in I_j} wv2(i,n) = yv(j,n), \qquad \forall j \in J, n \in N$$

Capacity Constraints

$$V_{ij}^{\min} wv2(i,n) \leq B2(i,j,n) \leq V_{ij}^{\max} wv2(i,n), \qquad \forall i \in I, j \in J_i, n \in N$$

Duration Constraints

$$T^f(i,j,n) = T^s(i,j,n) + \alpha_{ij} wv2(i,n) + \beta_{ij} B2(i,j,n), \qquad \forall i \in I, j \in J_i, n \in N$$

Sequence Constraints

Same Task in Same unit:

$$T^s(i,j,n+1) \geq T^f(i,j,n) - H(2 - wv2(i,n) - yv(j,n)),$$
$$\forall i \in I, j \in J_i, n \in N, n \neq N$$

$$T^s(i,j,n+1) \geq T^s(i,j,n), \qquad \forall i \in I, j \in J_i, n \in N, n \neq N$$

$$T^f(i,j,n+1) \geq T^f(i,j,n), \qquad \forall i \in I, j \in J_i, n \in N, n \neq N$$

Different Tasks in the Same unit

$$T^s(i',j,n+1) \geq T^f(i,j,n) - H(2 - wv2(i',n) - yv(j,n)),$$
$$\forall j \in J, i \in I_j, i' \in I_j, i \neq i', n \in N, n \neq N$$

Completion of previous tasks

$$T^s(i',j',n+1) \geq T^f(i,j,n) - H(2 - wv2(i',n) - yv(j',n)),$$
$$\forall j, j' \in J, i \in I_j, i' \in I_j, i \neq i', n \in N, n \neq N$$

Completion of previous tasks

$$T^s(i,j,n+1) \geq \sum_{n' \in N, n' \leq n} \sum_{i' \in I_j} (T^f(i',j,n') - T^s(i',j,n')), \quad \forall i \in I, j \in J_i, i \neq i', n \in N, n \neq N$$

Time Horizon Constraints

$$T^f(i,j,n) \leq H, \qquad \forall i \in I, j \in J_i, n \in N$$

$$T^s(i,j,n) \leq H, \qquad \forall i \in I, j \in J_i, n \in N$$

The results of implementing this decomposition for Example 1 are presented in the first column of Table 3-12. The second column shows results of an alternative decomposition where sub-problem 1 contains only one of the binding maximum capacity constraints from LP relaxation of original problem, and sub-problem 2 contains all the rest constraints.

		Proposed decomposition	Decomposition where sub-problem 1 contains only one of the binding maximum capacity constraints
8 hours	OBJ	1760.00	1498.19
5 event points	CPU (sec)	0.06	0.41
16 hours	OBJ	4343.75	3737.10
9 event points	CPU (sec)	1.80	286.71
24 hours	OBJ	6873.333	-----------
13 event points	CPU (sec)	615.06	> 4000 sec. resource limit

Table 3-12: LD Results for Example1

Although the second decomposition with single constraint gives rise to better objective function, it does not provide efficient speed up in order to be used for larger scale problems. No solution could be obtained for the case of 24 hours time horizon with resource limit of 4000 seconds where 13 event points are required.

Comparing the Lagrangean relaxation with the Lagrangean decomposition, it should be noticed that LD performs better than LR for the larger problem where the need for decomposition is imperative. However both of LD and LR are valid alternatives in order to generate an upper bound in the framework presented in the next section.

3.4 PROPOSED ALGORITHMIC APPROACH

Based on the heuristic-based approaches and Lagrangean relaxation/Lagrangean decomposition presented in the previous sections the following algorithmic procedure is proposed for the efficient solution of the short-term scheduling problem (Figure 3-4).

Step 1: Initialize the procedure by solving the linear relaxation of the problem to obtain the initial values of the dual variables and binding constraints.

Step 2: Formulate and solve the Lagrangean relaxation problem and Lagrangean decomposition problem.

Step 3: Based on the results of optimization problems in step 2 determine the current upper bound as: $Z^{UB}=min\ (Z^{LR}(u^k),\ Z^{LD}(u^k))$

Step 4: According to the nature of the scheduling problem utilize one of the heuristic based approaches proposed in section 4 to obtain a feasible solution Z^{LB} of the original problem. Update the lower bound in LR and LD with $Z^* = Z^{LB}$.

Step 5: Check for convergence using the following two criteria, first if the difference between the upper and lower bound is less than a required tolerance or second if a specific number of iterations has been reached. If one of these criteria is met, the calculation stops, otherwise continues to step 6.

Step 6: Update the values of dual variables u using the subgradient method:

$$u^{k+1} = u^k + t_k (x^k - z^k)$$

$$t_k = \frac{\lambda^k \left(Z^{UB} - Z^*\right)}{\left\|x^k - z^k\right\|^2}$$

where the x and z vectors correspond to the identical copies of the set of batch size variables and assignment variables.

Step 7: Increase the iteration count $k=k+1$ and return to Step 2.

The proposed approach was used to solve Example 1 for a time horizon of 18 hours using 10 event points. All computations performed on a Sun Ultra 60 workstation with CPLEX 6.6. The steps of the solution procedure are listed in Table 3-13.

Lower Bound	Upper Bound
	Solve the LP relaxation to optimality of 4894.48 in 0.45 seconds.
Use event-based heuristic decomposition and generate a lower bound of 3873.33 in 0.86 seconds.	
	Apply Lagrangean decomposition and generate an upper bound of 4969.17 in 9.6 sec.
	Apply Lagrangean relaxation and

	generate upper bound of 4813.55 in 25.4 seconds. Choose the lower upper bound.
Determine the Required production amount according to the solution of 4813.55 at LR and use the required production method to obtain a solution of 4193.94 in 351.40 seconds.	
Based on the Gantt Chart of the above solution, identify reactor 1 as critical unit. This step results in an objective value of 4334.38 in 787.04 seconds.	
	Apply Lagrangean relaxation based on the solution of 4334.38. A solution of 4449.98 is achieved in 526.77 seconds.

Table 3-13: Solving Example 1 with Proposed Algorithm

At the end of the steps shown in Table 3-13, the gap between upper and lower bounds is within the required tolerance of 3%, thus the iterations stop. The schedule obtained is shown in Figure 3-5 and compared with the optimal schedule of the original problem shown in Figure 3-6. Note that although these two schedules are slightly different in terms of the task assignments and timing, the two schedules are equivalent regarding the optimal production of the two products. It should be pointed out that the proposed algorithm did not result in a decreased computational time compared with the original problem since this takes only 760.12 seconds to solve with the same objective value.

However, the efficiency of the algorithm is shown with Example 2 since in this case the original scheduling problem cannot be solved to optimality. The steps of the solution procedure are shown in Table 16. The time horizon under consideration is 18 hours and 16 event points are used. All computations were performed on Sun Ultra 60 workstation with CPLEX6.6.

Lower Bound	Upper Bound
	Solve the LP relaxation to optimality of 852366.80 in 0.74 seconds.
Use upper bound to determine the required production amount, employ required production method to obtain a feasible solution of 680312.91 in 120.46 seconds.	
	Using feasible solution as lower bound in Lagrangean decomposition, reduce the upper bound to 826382.35 in 1.93 seconds.
Applying required production method with updated upper bound, improve the lower bound to 707164.10 in 1517.04 seconds.	
Identify Unit 3 as the critical machine in the Gantt Chart and use the critical unit method. This results in an objective value of 748568.08 in 4083.29 seconds.	
	Incorporate the newest feasible solution as lower bound in

	Lagrangean relaxation and generate the next upper bound of 770085.84 in 599.57 seconds.

Table 3-14: Solving Example 2 with Proposed Algorithm

At the end of the iterations shown in Table 3-14, the gap between upper and lower bounds is 2.87%, satisfying the tolerance of 3%. The algorithm stops with total computation time of 6323.03 seconds. The optimal schedule obtained corresponds to the objective function value of 7.48E5 and is shown in Figure 3-7. Figure 3-8 shows the schedule obtained by solving the original problem after 10,000 seconds computation time.

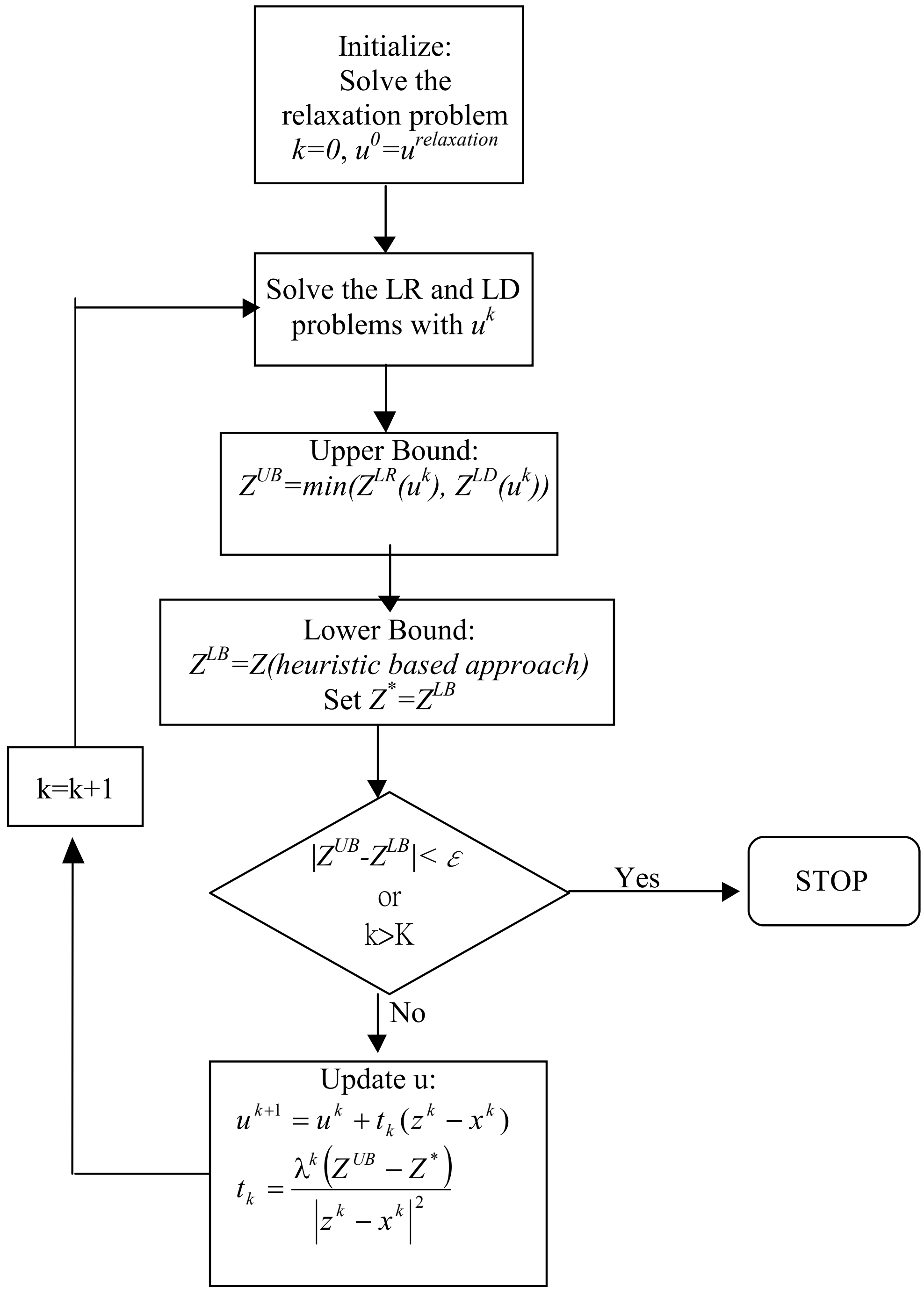

Figure 3-4: Proposed Algorithmic Procedure

3.5 SUMMARY

In this chapter a number of decomposition-based approaches have been proposed for the efficient solution of the short-term scheduling problem. The heuristic-based approaches are shown to be very efficient resulting in up to one order of magnitude reduction of required computational time, thus allowing the solution of large-scale scheduling problem. The main advantages and disadvantages of all proposed approaches are presented and illustrated through a few example problems. Lagrangean relaxation and Lagrangean decomposition approaches are utilized to obtain an upper bound of the scheduling problem. A number of different constraints and variables to be decomposed are investigated in order to provide the tightest upper bound. Finally, an iterative algorithmic procedure is proposed exploiting the advantages of the heuristic-based approaches and the Lagrangean relaxation/Lagrangean decomposition procedure, resulting in refined schedules close to the optimal solution for realistic size scheduling problems.

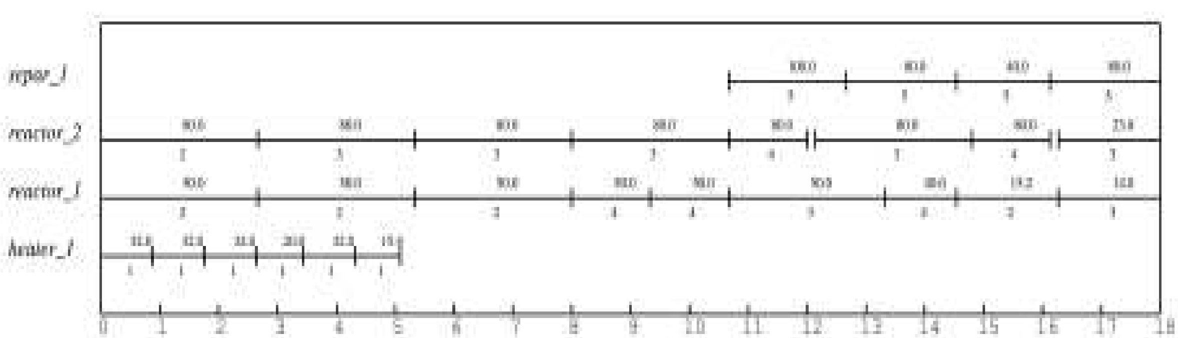

Figure 3-5: Schedule of Example 1 Generated by Implementing Proposed Algorithm

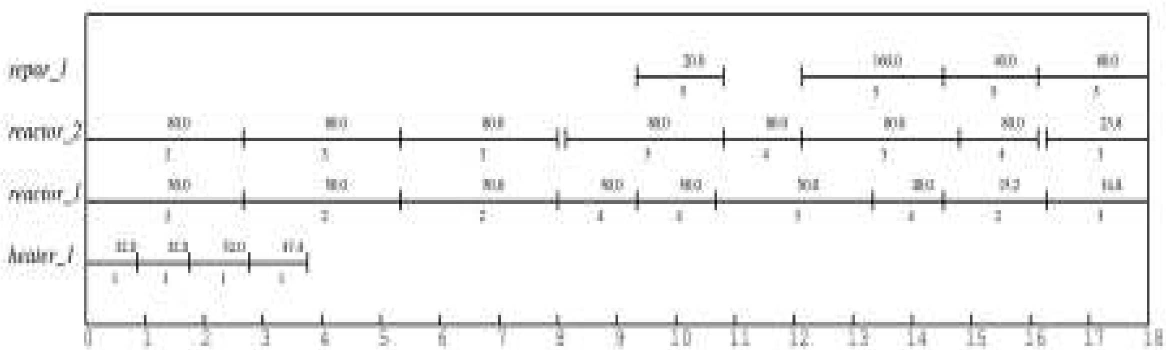

Figure 3-6: Optimal Schedule of Example 1 Generated by Solving the Original Problem

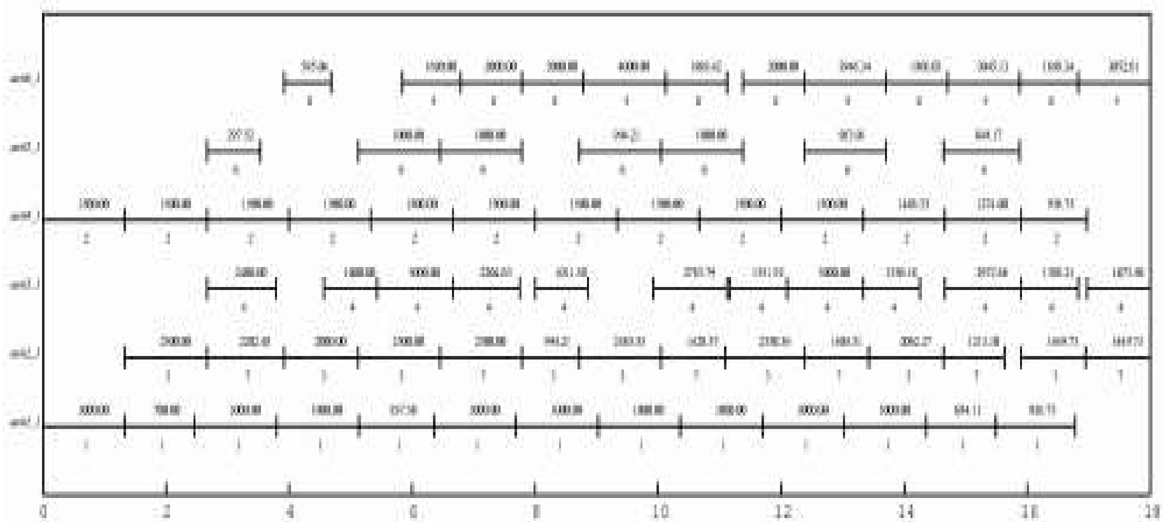

Figure 3-7: Schedule of Example 2 Generated by Implementing Proposed Algorithm

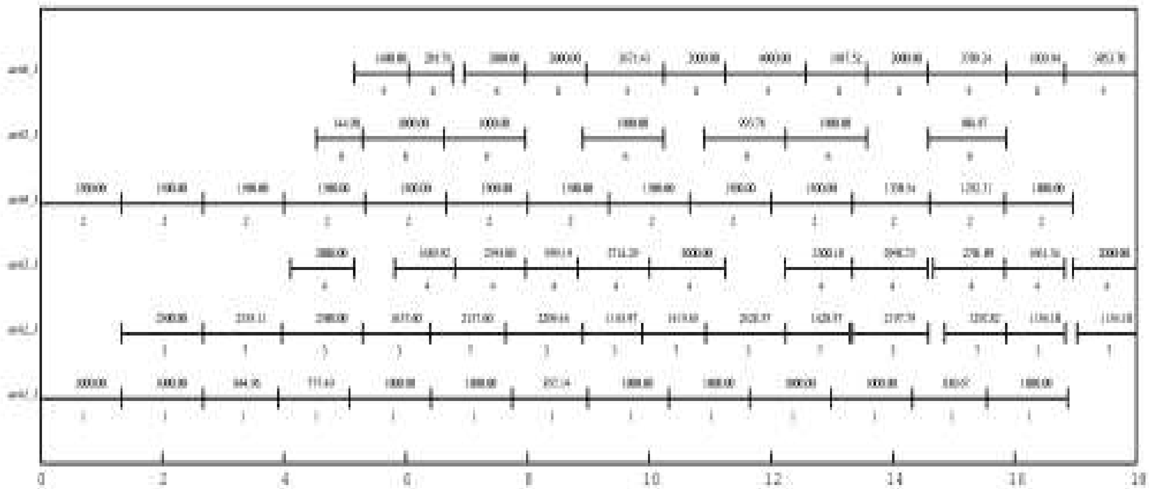

Figure 3-8: Optimal Schedule of Example 2 Generated by Solving the Original Problem

NOTATION

Indices

i	task
j	unit
n	event point representing the beginning of a task
s	state

Sets

I	tasks
I_j	tasks that can be performed in unit j
I_s	tasks that process state s and either produce or consume
J	units
J_i	units that are suitable for performing task i
N	event points within the time horizon
S	states

Parameters

V_{ij}^{min}	minimum amount of material processed y task i required to start operating unit j
V_{ij}^{max}	maximum amount of material processed by task i required to start operating unit j
ST_s^{max}	available maximum storage capacity for state s
r_s	market requirement for state s at event point n
$\rho^p{}_{si}$, $\rho^c{}_{si}$	proportion of state s produced, consumed from task i, respectively
α_{ij}	constant term of processing time of task i at unit j
β_{ij}	variable term of processing time of task i at unit j
H	time horizon
$price_s$	price of state s
P_s^{res}	required production of state s

Variables

$wv(i,n)$	binary variables that assign the beginning of task i at event point n
$yv(j,n)$	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
$d(s,n)$	amount of state s being delivered to the market at event point n

$ST(s,n)$	amount of state s at event point n
$STIN(s)$	amount of state s imputed initially
$T^s(i,j,n)$	time that task i starts in unit j at event point n
$T^f(i,j,n)$	time that task i finishes in unit j while it starts at event point n
$P(s,n)$	production of state s at event point n

CHAPTER 4

SIMULTANEOUS APPROACH FOR PRODUCTION PLANNING AND SCHEDULING

Planning problem with time horizon from few days to months usually results in expensive computational cost that prevents it from being solved to optimum, sometimes even a feasible solution. A periodical scheduling approach is proposed in this chapter. The mathematical model determines the optimal production schedule within a cycle as well as time length of the cycle such that it can in principle constitute a near optimal solution in reasonably small computational time for any large-scale planning problem. Thus the planning and scheduling problems are simultaneously tackled.

4.1 INTRODUCTION

In chemical industry, the production planning and short-term scheduling have played important roles. Production planning determines the optimal allocation of resources within the production facility over a time horizon of a few weeks up to a year. On the other hand, short-term scheduling provides the feasible production schedules to the plant. The integration of planning and scheduling usually leads to intractable model in terms of the required computational time. Thus one has to trade-off optimality with computational efficiency.

A large number of publications are devoted to modeling and solution of the planning and scheduling problems. Early work has focused on the development of mathematical models based on discretization of time horizon into a number of intervals of equal duration (Kondili et al., 1993b). Sahinidis and Grossmann (1989) presented a multiperiod MILP model for long range planning. In order to reduce the computational expense, several strategies were investigated in their work, including branch-and-bound, the use of integer cuts, strong cutting planes, Benders decomposition and heuristics. Liu and Sahinidis (1996) reformulated this MILP and developed a solution approach based on constraints generation scheme and projection in conjunction with the strong cutting plane algorithm. Heever and Grossmann (1999) proposed a disjunctive Outer Approximation algorithm for the solution of a multiperiod design and planning problem, which is an extension of the logic-based Outer Approximation algorithm for single period MINLPs (Turkay and Grossmann, 1996). The main limitations of these time discretization methods are that (a) they require all the tasks to start and finish at the boundaries of time intervals, thus resulting in sub-optimal solutions and (b) they require a large number of binary variables due to unnecessary time discretization that results in large mathematical models difficult to be addressed.

The solution methodologies of addressing planning problems can be distinguished into two main categories the simultaneous and the hierarchical approaches (Bose and Pekny, 2000). The hierarchical approaches involve consideration of planning and scheduling models which can be decoupled. A detailed review will be given in chapter 5. Following the simultaneous approach the whole planning and scheduling problem is considered and solved for the entire horizon. Papageorgiou and Pantelides (2000) proposed a single-level formulation for campaign planning problem. The algorithm determines the campaigns (i.e., duration and constituent products) as well as the production schedule simultaneously. Bassett et al. (2000) presented a model that spans a longer time periods giving more details towards the intermediate future than distant future. Orcun et al. (2001) developed a unified continuous-time model of MINLP for planning problem. After the MINLP is reformulated as MILP by using linearization techniques, the problem is still addressed as “extremely difficult almost impossible” to solve. Similar computational barrier is met when the continuous formulation in section 1.1 is applied to longer time horizon.

Periodic scheduling is developed in the context of campaign-mode operation (Shah. et al. 1993 and Kondili et al. 1993a). Schilling and Pantelides (1999) presented a mathematical programming formulation which is based on a continuous representation of time. A novel branch-and-bound algorithm that branches on both discrete and continuous variables was proposed. This work was extended to multipurpose plants periodic scheduling problem. The proposed model resulted in the determination of both the optimal duration of the operating cycle and the detailed schedule in each cycle. The objective function was to minimize the average cost which corresponds to a nonlinear function. A relaxed form of the optimization problem was generated after replacing the definition of the objective function by a set of linear constraints. Castro et al. (2003) modified their short-term scheduling formulation to fit periodic scheduling requirements for an industrial application.

In this chapter, a new model is proposed to address the planning and scheduling problem simultaneously based on the concept of periodic scheduling stated in section 4.2. This continuous-time representation is extended from that of the short-term scheduling for batch plants in section 1.1 and thus requires less number of variables and constraints compared to discrete time and other continuous-time formulations. The mathematical formulation is presented in detail in section 4.3 and applied to different examples in section 4.4. The results are compared with existing approaches.

4.2 PERIODIC SCHEDULING APPROACH

The planning problem that is considered in this work is defined as follows. Given are:

(i) the production recipe (i.e., the processing times for each task at the suitable units, and the amounts 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;

(iv) the time horizon under consideration;

(v) the market requirements of products.

The objective is to determine the optimal operational plan to meet a specified economic criterion such as maximal profit or minimal cost while satisfying all the production requirements. It should be noted however, that the product demands are considered at the end of time horizon and all of the above constraints are fixed within the time horizon.

The idea of periodic scheduling is frequently utilized for the solution of planning problem described above. The optimal solution of planning problem implies that the schedule does not exhibit any periodicity (Pantelides, 1994). However, one has to balance against the computational complexity of solving non-periodic schedules for a long time horizon. The presented periodic scheduling approach resides on the following assumption. For the case that the time horizon is long compared with the duration of individual tasks, a proper time period exists, which is much smaller than the whole time horizon, within which, some maximum capacities or crucial criteria have been reached so that the periodic execution of such schedule will achieve results very close to the optimal one by solving the original problem without any periodicity assumption. Thus the size of the problem is reduced to a much smaller one that can be efficiently solved. Besides its computation efficiency the proposed operation plan is more convenient and easier to implement since it assumes repetition of the same schedule. In this approach, the variables include the cycle time length as well as the detailed schedule of this period, which are defined as unit period and unit schedule, respectively. Unlike the short-term scheduling where all intermediates other than those provided initially have to be produced before the beginning of the tasks, unit schedule can start with certain amounts of intermediates as long as storage capacity constraints are not violated. The initial amounts of intermediates are equal to the amounts stored at the end of unit period, so as to preserve the material balance across the boundaries as shown in Figure 4-1.

It should be noticed that in periodic scheduling, each processing unit may have an individual cycle as long as the cycle time is equal to the duration of the unit period, so as illustrated in Figure 4-2a, all the units do not necessarily share the same starting and ending time points. This concept can be found in Shah et al. (1993) in their discrete time representations for periodic scheduling problem as “wrap-around”. Schilling and Pantelides (1999) incorporated the same concept into their continuous-time formulation based on the resource-task network (RTN) representation (Pantelides, 1994). In this work, the same concept is used together with the continuous-time representation using the idea of event points as will be explained in detail in section 4.3.

Figure 4-2a illustrates the unit schedule that corresponds to the periodic schedule of Figure 4-1. When a larger time period has to be scheduled using the unit schedule, overlapping is allowed in order to achieve better resource utilization. In this way the equivalent unit schedule is determined as shown in Figure 4-2b. Note that by using this idea better schedules are determined since tasks are allowed to cross the unit schedule boundaries. If time decomposition was applied even using the optimal cycle time length, the resulted schedule would be inferior since only small batches are allowed.

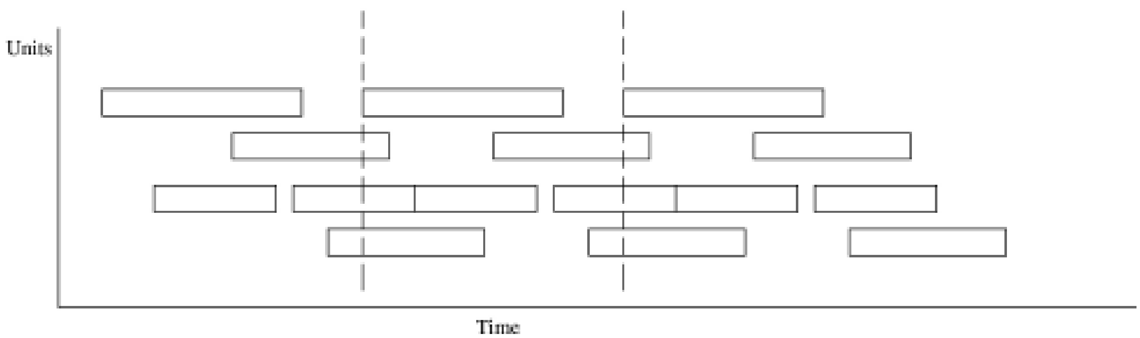

Figure 4-1: Periodic Schedule

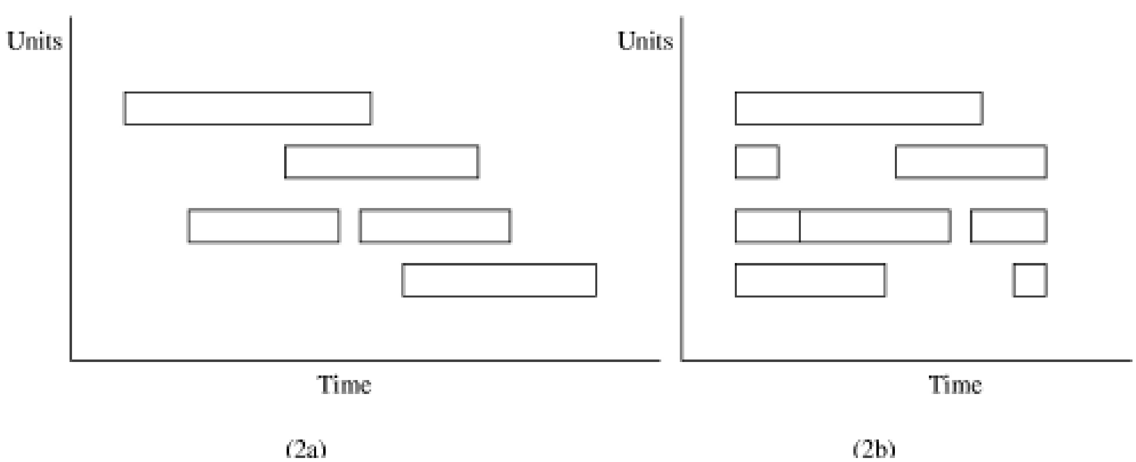

Figure 4-2: Unit Schedule

4.3 MATHEMATICAL MODEL

When large time horizon is considered (i.e. a few days), the size of the scheduling model becomes intractable. For example considering a time horizon of 24 hours, the formulation of Ierapetritou and Floudas (1998) involves 1517 constraints, 546 continuous variables and 156 binary variables using 13 event points. It takes 79551.29 CPU seconds on P*III* 500MHz using GAMS/CPLEX 7.1 to get a solution of 6036.491 objective function value which cannot be approved optimality since further increase of event points causes computational infeasibility. When the same formulation is used for a time horizon of 168 hours, a feasible schedule cannot be obtained for the whole time horizon. These results point to the importance of developing a new approach for the simultaneous solution of planning and scheduling problem.

4.3.1 FORMULATION

In order to represent the features of periodic scheduling for planning, the following constraints are introduced that enforce the continuity in plant operation between cycles.

Material Balances Between Cycles

$$STIN(s) = ST(s,n), \quad \forall s \in S, n = N \tag{4-1}$$

Constraints (4-1) represent the key feature of periodic scheduling. The intermediates stored at the last event point of the previous cycle should equal the amount of material needed to start the next cycle in order to maintain smooth operation without any accumulation or shortage in between. Raw material and product are calculated based on the consumed or produced amounts in the first cycle.

Demand Constraints

$$\sum_{n \in N} d(s,n) \geq r_s H, \quad \forall s \in S \tag{4-2}$$

where r_s represents the average requirement. Constraints (4-2) express the requirement of meeting demand specifically for all products. Note that the requirements for the time horizon of planning problem are assumed to be evenly distributed to each cycle.

Cycle Timing Constraints

Unlike the sequence constraints (1-7)-(1-12) in the previous part which describe the sequence of tasks within the same cycle, cycle timing constraints express the timing relationship of the last task in the previous cycle and the first task in the current cycle so as to maintain continuity of operation between cycles.

Cycle Timing Constraints: Task in the same unit

$$T^s(i^{'},j,n0) \geq T^f(i,j,n) - H, \qquad \forall j \in J, \forall i,i' \in I_j, n = N \tag{4-3}$$

where n_0 stands for the first event point in the current cycle. $T^f(i,j,n)-H$ corresponds to the time of last event point in the previous cycle. Constraints (4-3) represent that task i' performing at the beginning of the cycle has to start after the end of task i at the previous cycle. Since only one task can take place in the same unit at each event point n, constraints (4-3) also express the correct recipe sequence for the same unit.

Cycle Timing Constraints: Task in the different units

$$T^s(i^{'},j',n0) \geq T^f(i,j,n) - H\,, \qquad \forall j,j' \in J, \forall i,i' \in I_j, i = i', n = N \tag{4-4}$$

Constraints (4-4) represent the requirement of the first task in a new cycle to start after the completion of the tasks in different units in previous cycle based on the recipe requirements. Similar to constraints (4-4), these constraints are written for the tasks that should take place consecutively in different units and ensure the correct sequence of tasks between cycles.

Time Horizon Constraints

$$T^s(i,j,n) \leq 2H, \quad \forall i \in I, j \in J_i, n \in N \tag{4-5}$$

$$T^f(i,j,n) \leq 2H, \quad \forall i \in I, j \in J_i, n \in N \tag{4-6}$$

Since the starting points of a cycle are not necessarily synchronized for all units, some units may start performing tasks later than others. The maximum idle time, however won't be greater than a cycle period given constraints (4-3), (4-4). Therefore the time horizon constraints (4-5), (4-6) represent the requirement of each task i to start and finish before two cycle lengths $2H$.

Cycle Length Constraints

$$\sum_{n\in N}\sum_{i\in I_j}\left(T^f(i,j,n)-T^s(i,j,n)\right)\le H,\quad \forall i\in I_j, n\in N \tag{4-7}$$

The cycle length constraints (4-7) state that the duration of all tasks performed in the same unit must be less than the cycle length *H*, which ensures that cycle of each unit cannot be longer than the cycle length.

Objective: Maximization of average profit

$$\frac{\sum_s\sum_n price_s \times d(s,n)}{H} \tag{4-8}$$

The objective function for the planning problem is to maximize the production in terms of profit due to product sales. Assuming periodic scheduling this objective is transformed to maximizing the average profit as shown in (4-8). The average profit is considered to express the dependence of the profit over the whole time horizon on both the production during each cycle and the cycle time. Note that the objective function involves fractional terms $d(s,n)/H$, thus giving rise to a MINLP problem. Alternative objectives can be also incorporated to express different scheduling targets such as cost minimization.

4.3.2 PROPOSED DECOMPOSITION APPROACH

In order to consider the whole planning problem the time horizon is divided into three periods, the initial period when the necessary amounts of intermediates are produced to start the periodic schedule, the main period when periodic scheduling is applied and the final period to wrap up all the intermediates. The initial and final periods are bounded by a time range and solved independently. The sum of time lengths of all three periods equals that of the whole time horizon. Given the optimal cycle length resulted from solving the periodic scheduling problem described in section 4.2, the problem for initial period is solved first with the objective function of minimum make-span so as to ensure the existence of feasible solution in order to provide those intermediates for periodic scheduling. Then the same problem is solved with the objective of maximizing the profit with the time horizon as obtained from the first solution. The problem for the final period can be solved in parallel once the time horizon for the cycle length and the initial period are determined. The intermediates considered for the final period are obtained from the unit schedule and the time horizon is the time left for the planning problem. Both initial and final problems are using the same set of constraints presented in the previous section except that for the final period the time horizon is fixed. The overall approach is schematically shown in Figure 4-3.

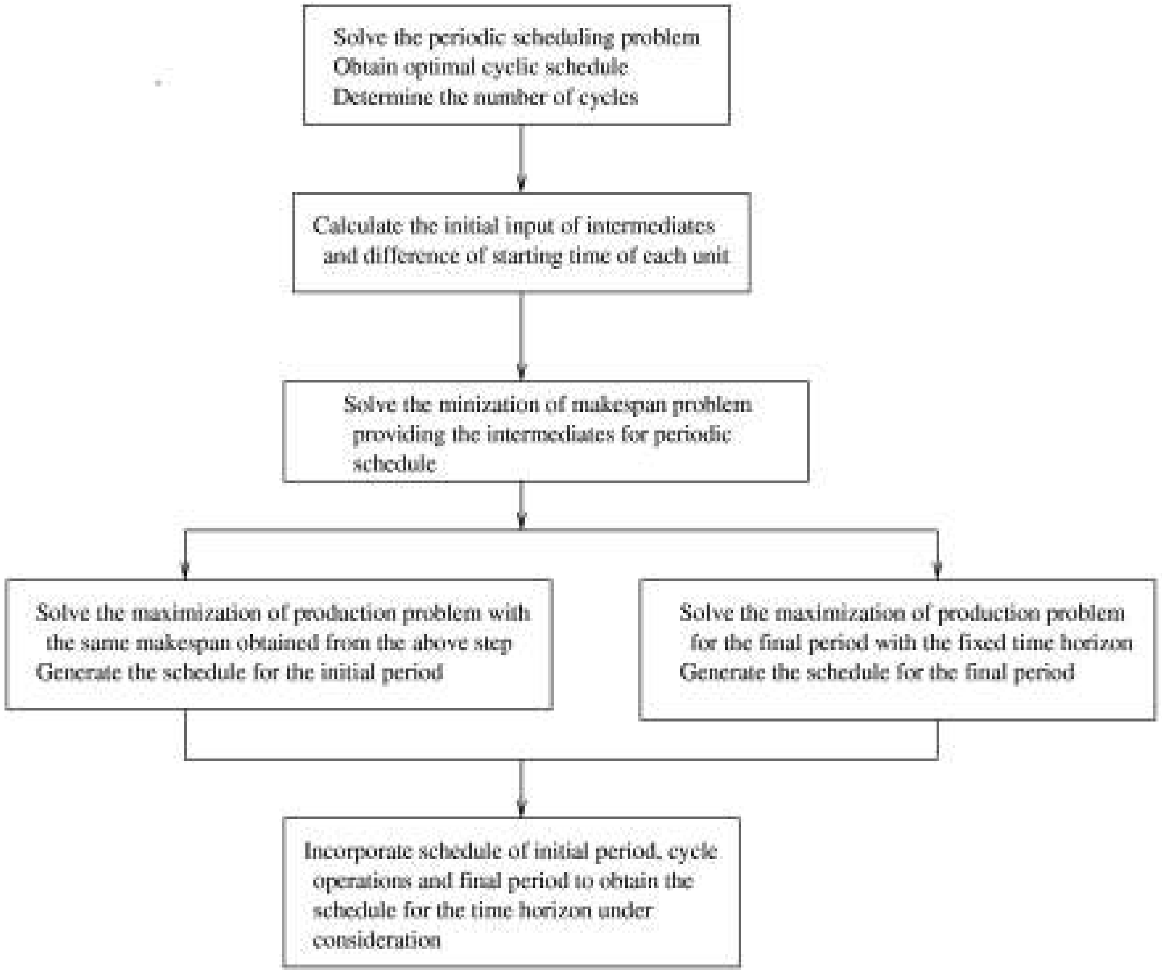

Figure 4-3: Flow Chart of Proposed Approach

4.4 CASE STUDIES

4.4.1 EXAMPLE 1

The model developed in section 4 corresponds to a mixed-integer non-linear programming (MINLP) problem. The nonlinearities appear only at the objective function as fractional terms of continuous variables. Thus using local optimization solvers such as

DICOPT or SBB global optimality cannot be guaranteed, however practically the solution often corresponds to global optimum as illustrated in the following examples. GAMS (GAMS Corporation Inc.)/DICOPT (DIscrete and Continuous OPTimizer) is used in this work that uses Outer Approximation/Equality Relaxation (Grossmann et al., 2002) as a MINLP solution procedure.

The example 1 in section 1.2 is solved on a Linux system with processor Pentium *III* 500MHz using DICOPT, CONOPT2 and CPLEX 7.1 as the MINLP, NLP and MILP optimization solver, respectively. In order to determine the optimal schedule and cycle length, the following strategy is considered. Instead of considering the whole cycle time range, for example 2-24 hours, several sub-ranges are considered 2-6 hours, 6-10 hours, up to 24 hours and the resulting problems are solved independently. The advantages are that

(i) each of the sub-period problem utilizes small number of event points, thus it speeds up the solution process;

(ii) it generates a number of scheduling alternatives that can be beneficial to plant manager who has to consider additional requirements such as work shift constraints;

(iii) each sub-problem can be solved independently and thus parallel computation can be implemented.

As shown in Table 4-1, the optimal cycle length obtained is 23.790 hours with the objective value of 279.029 units. The optimal schedule is shown in Figure 4-4. Additional computational information is presented in Table 4-2 for the sub-model with cycle time range of 2-6 hours.

Cycle time range (h)	Number of event points	Objective function value	Optimal cycle time (h)	CPU time (s)
2-6 hours	4	268.289	5.094	2.86
6-10 hours	6	272.247	9.036	512.00
10-14 hours	7	273.801	12.978	5365.74
14-18 hours	9	276.447	14.407	305.88
18-21 hours	11	277.363	19.709	545.83
21-24 hours	12	279.029	23.790	2884.41

Table 4-1: Solution for Example 1

Relative optimality criterion	0.01
Cycle time range (h)	2-6
Number of event points	4
Binary variables	48
Continuous variables	299
Constraints	530
Optimal Cycle time (h)	5.094
Objective function value	268.289
CPU time (s)	2.86

Table 4-2: Computational Statistics for Example 1

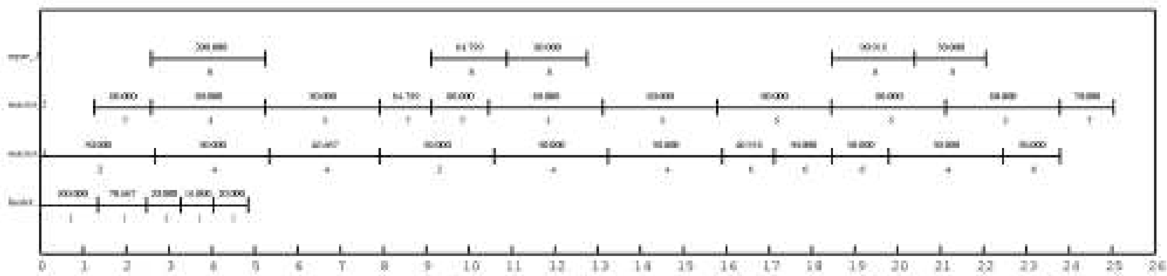

Figure 4-4: Optimal Solution of Example 1

As pointed out in section 4.3, if a time horizon of 168 hours is considered, the short-term scheduling problem is computationally intractable. Therefore, the time horizon is divided in three periods as proposed in section 4.3.2, the initial period when the necessary amounts of intermediates are produced to start the periodic schedule, the main period when periodic scheduling is applied and the final period to wrap up all the intermediates. Applying this approach to the motivating example for a time horizon of 168 hours, the periodic scheduling problem was solved to optimality obtaining an average profit of 279.029 in 2884.41 CPU sec in order to generate the unit schedule where the initial input of intermediates and difference of starting time of each unit were calculated. 6 cycles were then determined to leave enough time for the initial and final periods to cover the necessary production. The problem of make-span minimization was then solved for the initial period to obtain the shortest time in order to provide the intermediates to cyclic operation. The problem of production maximization for the initial and final period are then solved simultaneously based on the make-span calculated from the minimization problem. It required 1590.35 CPU sec to solve the minimization problem and 3187.21 and 611.23 CPU sec to solve the maximization problems for the initial and final period problems, respectively. The overall objective function value representing the total profit over the whole time horizon under consideration is 45698.90. Figures 4-5, 4-6 illustrate the schedule for the period of [0,65] that involves the initial period of 12.47 hours together with two cycles of operations and for the period of [103,168] that involves two cycles and the final period of 12.79 hours. Note that the batch sizes in these figures are round off to the closest integer for clarity in presentation of the schedules.

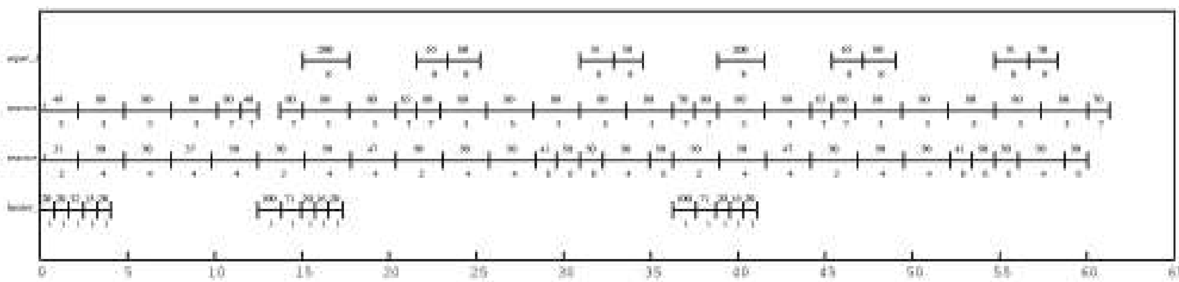

Figure 4-5: Schedule of Initial Period for 168 Hours of Example 1

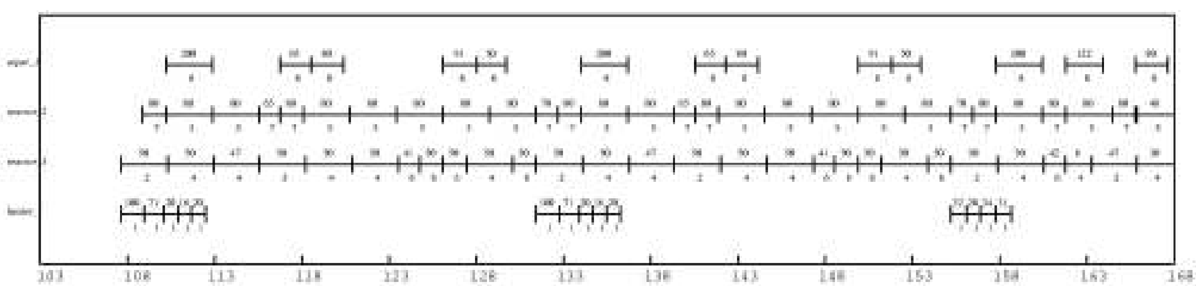

Figure 4-6: Schedule of Final Period for 168 Hours of Example 1

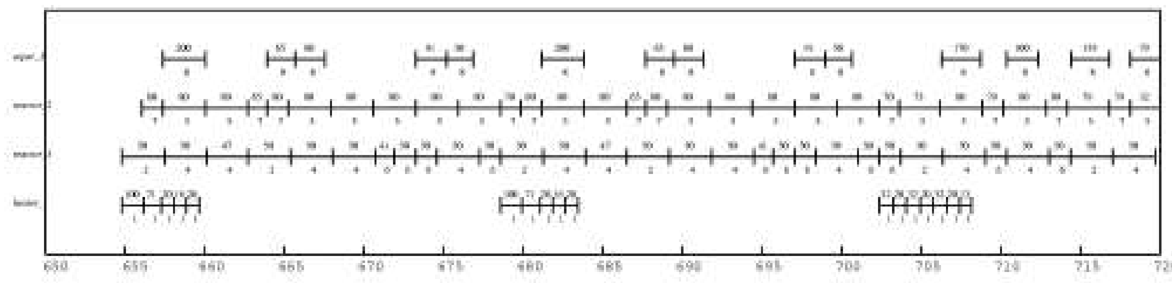

Figure 4-7: Schedule of Final Period for 30 Days of Example 1

A time horizon of 30 days is considered next for the same example. The proposed approach generates the initial and final periods and determines the optimal cyclic operation to be performed for the rest of the time horizon. The overall objective function value obtained is 1.998E5 corresponding to 29 cycles of operation. The schedule for the initial period is the same as obtained for a time horizon of 168 hours shown in Figure 4-5 since the proposed approach does not consider the whole time horizon when generating the schedule for the initial period. The production maximization problem in the final period requires 6129.82 CPU sec. Figure 4-7 shows the schedule for the final period of [650,720] that involves two cycles and the final period of 17.62 hours.

4.4.2 EXAMPLE 3 (MODIFIED EXAMPLE 1)

This example was considered by Schilling and Pantelides (1999) and is similar to the motivating example except the following differences:

(i) there is no heating process in modified Example 1;

(ii) Hot A has both storage capacity and supply for 1000;

(iii) reaction 2 in reactor 1 produces Int AB only;

(iv) all the units have an identical maximum capacity of 80 and different minimum capacities as 20, 30, 40 for reactor 1, reactor 2 and still respectively;

(v) the processing times of all tasks are those of Example 1 multiplied by 10;

(vi) the price for product 2 is 12 instead of 10 in motivating Example 1.

It is denoted as Example 3 in this chapter. The proposed approach is applied to this example in order to compare the results with the results presented by Schilling and Pantelides (1999). In order to obtain the global optimized solution, GAMS/BARON is utilized as MINLP solver. BARON (Sahinidis and Tawarmalani, 2002) is based on conventional branch-and-bound algorithm and integrates range reduction techniques which contract the search space at each node together with a number of compound branching schemes that accelerate convergence of standard branching strategies. BARON guarantees to provide global optimality to the type of MINLP problems involving fractional terms.

In the cycle length range of 20–40 h considered by Schilling and Pantelides, the solution procedure results in approximately (with round off errors) the same objective value and schedule. The required computation is only 26.76 CPU seconds on Pentium*III* 750 MHz using GAMS/BARON. The optimal cycle length is 36.64 h with an objective value of 28.94 producing 61.714 units of product and 32 units of product 2 per cycle. The optimal Gantt-Chart is shown in Figure 4-8. Schilling and Pantelides used their own branch-and-bound algorithm and a parallel computation scheme due to the complexity of the linearized constraints. They implemented the parallel computing with a network of seven Sun ultra workstations and reported 81 CPU seconds for this example. In a recent work Castro et al. (2003) presented a different RTN periodic continuous-time formulation and applied to this example. These results are compared in Table 4-3. Note that significant less number of variables are required using the proposed formulation.

Castro et al. (2003) reported a computational time of 4.36 CPU seconds using DICOPT as MINLP solver on a Pentium*III* 1 GHz machine. When a longer range of 20–80 h is solved, they obtain a sub-optimal solution corresponding to an objective value of 32.210 and cycle length of 63.708, compared to the optimal solution obtained using the proposed formulation that corresponds to an objective value of 32.893 and cycle length of 62.540.

Table 4-4 presents the optimal cycle lengths for different time ranges solved by GAMS/DICOPT. Although GAMS/DICOPT is only a local optimization solver, these results are proved to correspond to the global optimal solution obtained using the global solver GAMS/BARON. It should also be noticed that by confining the cycle time to be less than 40 hours a sub-optimal solution is obtained since the optimal cycle length is found to be 171.575 hours. This result highlights the advantage of the proposed approach.

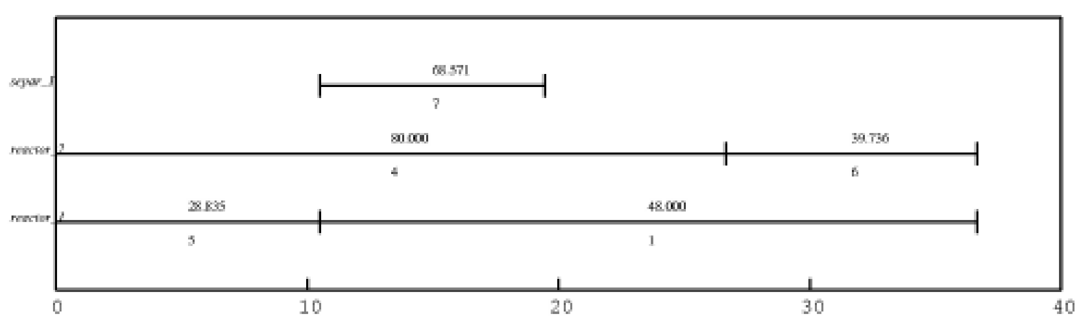

Figure 4-8: Optimal Solution of Example 3

	Proposed approach	Formulation of Schilling and Pantelides	Formulation of Castro et al.
Cycle time range (h)	20-40	20-40	20-40
Event points or Time slots	3	6	3
Binary variables	28	81	42
Continuous variables	112	437	127
Constraints	272	440	164
Optimal Cycle time (h)	36.64	36.81	36.87
Objective function value	28.94	28.72	28.77

* Parallel implementation using 7 Sun workstations

Table 4-3: Comparison of Results of Example 3

Cycle time range (h)	Number of event points	Objective function value	Optimal cycle time (h)	CPU time (s)
20-40 hours	3	28.942	36.645	0.82
40-70 hours	5	32.893	62.540	25.78
70-100 hours	5	33.829	93.333	10.00
100-140 hours	6	34.321	102.828	109.27
140-170 hours	8	34.434	159.048	5601.49
170-200 hours	10	34.957	171.575	312.67
200-240 hours	11	34.725	223.240	6020.55

Table 4-4: Solution for Example 3

4.4.3 EXAMPLE 2

The proposed approach was applied on Example 2 which is stated in section 3.3.3 and the results are shown in Table 4-5. The optimal cycle obtained in 42.55 CPU sec corresponds to 10.667 hours with an objective value of 48946.324 and the optimal schedule shown in Figure 4-9. Considering a time horizon of 30 days for this example, the proposed approach determines 66 cycles of operation, schedules for an initial period of 4.59 hours and a final period of 11.41 hours as shown in Figure 4-10, 4-11 respectively. The overall objective function value is 3.503E7 and the CPU time for solving initial and final periods is 0.54 and 184.47 CPU sec, respectively. The proposed approach shows advantage both in optimality and computational tractability when solving the planning and scheduling problem simultaneously.

Cycle time range (h)	Number of event points	Objective function value	Optimal cycle time (h)	CPU time (s)
2-6 hours	5	48305.00	4.000	3.21
6-9 hours	7	48671.471	6.667	10.51
9-12 hours	10	48946.324	10.667	42.55
12-15 hours	13	48871.364	14.667	12271.43
15-18 hours	15	48840.611	17.333	3234.48
18-21 hours	16	48750.000	18.667	1300.80
21-24 hours	20	48946.324	21.333	900.94

Table 4-5: Solution for Example 2

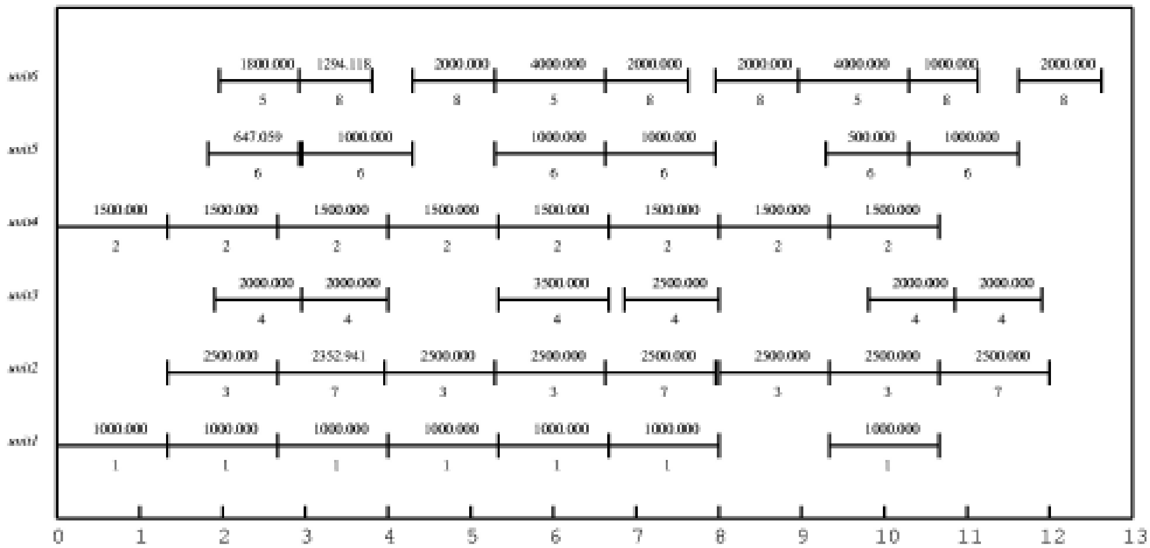

Figure 4-9: Optimal Solution of Example 2

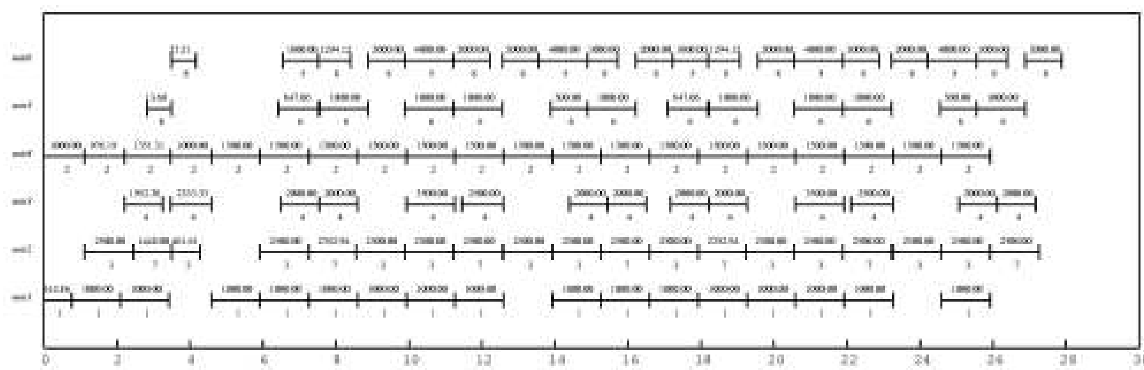

Figure 4-10: Schedule of Initial Period for 30 Days of Example 2

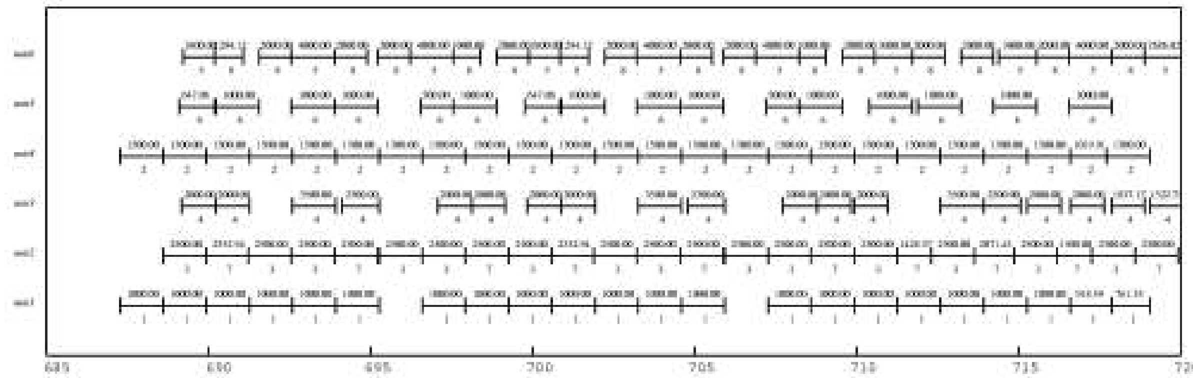

Figure 4-11: Schedule of Final Period for 30 Days of Example 2

4.5 SUMMARY

This chapter presents a new framework for solving the planning and scheduling problem simultaneously based on a continuous-time formulation in order to determine the optimal periodic schedules as well as the optimal cycle length for multipurpose batch plant. Compared with existing approaches, the proposed formulation results in less number of variables and constraints. Although the proposed model corresponds to a nonconvex MINLP problem, it is shown that it can be solved efficiently to global optimality by GAMS/DICOPT. A hierarchical framework is proposed in the next chapter to enable the solution of the planning and scheduling problem when the demand and price vary within the time horizon.

NOTATION

Indices

i	task
j	unit
n	event point representing the beginning of a task
s	state

Sets

I	tasks
I_j	tasks that can be performed in unit j
I_s	tasks that process state s and either produce or consume
J	units
J_i	units that are suitable for performing task i
N	event points within the time horizon
S	states
IS	subset of all involved intermediate states s

Parameters

V_{ij}^{min}	minimum amount of material processed y task i required to start operating unit j
V_{ij}^{max}	maximum amount of material processed by task i required to start operating unit j
ST_s^{max}	available maximum storage capacity for state s
r_s	market requirement for state s at event point n
$\rho^p{}_{si}$, $\rho^c{}_{si}$	proportion of state s produced, consumed from task i, respectively
α_{ij}	constant term of processing time of task i at unit j
β_{ij}	variable term of processing time of task i at unit j
U	upper bound of cycle time length
$price_s$	price of state s

Variables

H	cycle time length
$wv(i,n)$	binary variables that assign the beginning of task i at event point n

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*
d(s,n)	amount of state *s* being delivered to the market at event point *n*
ST(s,n)	amount of state *s* at event point *n*
STIN(s)	amount of state *s* imputed initially
$T^s(i,j,n)$	time that task *i* starts in unit *j* at event point *n*
$T^f(i,j,n)$	time that task *i* finishes in unit *j* while it starts at event point *n*

CHAPTER 5

HIERARCHICAL APPROACH FOR PRODUCTION PLANNING AND SCHEDULING

In this chapter, the solution of production planning and scheduling problem is addressed through a hierarchical framework. The planning problem aggregates orders in the planning period and considers uncertainty utilizing a multi-stage stochastic programming formulation where three stages are considered with increasing level of uncertainty. The planning model includes material balances and time horizon constraints which involve a sequence factor to reflect the recipe complexity. Using a rolling horizon strategy, the production for the current stage is provided to the scheduling problem, which is solved using a continuous-time formulation. A general framework is presented with a mechanism to converge the planning and scheduling results before solving for the next period. This proposed approach is illustrated with a planning example.

5.1 INTRODUCTION

As discussed in the previous chapter, the simultaneous approach emphasizes that planning and scheduling decisions need to be considered in a single model in order to achieve the optimality. A successful periodic scheduling formulation as well as the solution approach is presented in chapter 4. However, cyclic operation is more appropriate for plants operating under a stable demand conditions (Pantelides, 1994), thus limiting the applicability of the periodic scheduling. Uncertainty is also a concern when longer planning horizons are considered. While for a smaller time horizon demands and prices can be considered deterministic, this is not the case for a time horizon of a month or larger. Additional disturbances may also upset the production schedule as for example rush order arrival and machine breakdown.

The hierarchical approaches involve the problem decomposition into decoupled planning and scheduling level problems. McDonald and Karimi (1997) developed production planning and scheduling models for application of single stage processor. The planning model divides the time horizon into a number of time periods with demands due at the end of each time period and compares different time-scale cases of planning periods and individual production events. Two continuous-time formulations are presented for the short-term scheduling problem where discrete time events can be accommodated. Papageorgiou and Pantelides (2000) presented a hierarchical approach attempting to exploit the inherent flexibility with respect to intermediate storage policies and multi-usage of the equipment. A three-step procedure was proposed. First, a feasible solution to the campaign planning problem subject to restrictive assumptions is obtained. Second, the production rate in each campaign is improved by removing these assumptions. Finally, the timing of the campaigns is revised to take advantage of the improved production rates. Harjunkoski and Grossmann (2001) presented a bilevel decomposition strategy for a steel plant production process. In this approach, products are grouped into blocks and scheduling problems for each block are solved separately followed by solving a MILP to find the sequence of these blocks. The solution obtained is not guaranteed to be optimal although near optimal solutions can be determined. Other research includes Bose and Pekny (2000), who used model predictive control ideas for solving the planning problem. A forecasting and an optimization model are established in a simulation environment. The former calculates the target inventory in the future periods while the latter tries to achieve such inventory levels in each corresponding period. The advantage of this approach is that fluctuation in demands and prices could be incorporated into planning level problem. Zhu and Majozi (2001) proposed an integration of planning and scheduling problems as well as a decomposition strategy for solving the planning problem. It is based on the idea that if the raw materials can be allocated optimally to individual plants, solving individual models for each plant can produce the same results as solving an overall model for the site. In the recent work, rolling horizon approach has been widely considered to reduce the computational burden (Dimitriadis et al., 1997, Sand et al., 2000 and Van den Heever et al. 2003). This approach makes decisions for a time period shorter than the planning time horizon and this time period moves forward as the model is solved. Dimitriadis presented RTN-based rolling horizon algorithms for medium-term scheduling of multipurpose plants. Sand et al. and Van den Heever employed rolling horizon as well as a Lagrangean type of decomposition for their planning and scheduling problems, while Sand et al. considered uncertainty in their planning model. Most of the existing approaches however have limitations due to overly simplified planning level problem, the lack of uncertainty and task sequence feasibility consideration.

A general hierarchical framework for the solution of planning and scheduling problems is proposed in this chapter. In the next section, a multi-stage planning model is presented which accounts for uncertainty. A parameter denoted as sequence factor is introduced to simplify the computational complexity and account for recipe time constraints. In section 5.3, a modified scheduling model is proposed to determine the production schedule following the basic ideas of the short-term scheduling formulation in section 1.1. An overall solution framework is presented in section 5.4 and illustrated with a planning example in section 5.5.

5.2 PLANNING MODEL

5.2.1 CONCEPTUAL MODEL

The overall decision process is based on the idea of rolling horizon strategy. The planning time horizon is decomposed into three stages with various durations based on the orders and market uncertainty. The first stage with the smallest duration is denoted as “current” period where operating parameters are considered deterministic. The second stage with larger duration is subject to small variability of demands and prices, and the final stage with largest duration has higher level of fluctuations regarding demands and prices. Uncertainty is modeled using the ideas of multi-stage programming (Dantzig, 1955), where each planning period corresponds to a different stage. Uncertainty is expressed by incorporating a number of scenarios at each stage. More scenarios are considered towards the last stage in order to represent the increasing level of uncertainty. Each scenario is associated with a weight representing the probability of the scenario realization. Moreover, it is assumed that each unit will process a certain number of batches at a full capacity and a single batch at flexible size at every stage, which is a valid assumption for realistic case studies. Using this basic idea the following planning model is developed.

5.2.2 STOCHASTIC MULTI-PERIOD PLANNING-LEVEL FORMULATION

Capacity Constraints

$$V_{ij}^{\min} wv^k(i,j,q^k) \le B^k(i,j,q^k) \le V_{ij}^{\max} wv^k(i,j,q^k),$$
$$\forall i \in I, j \in J_i, q^k \in Q^k \qquad (5\text{-}1)$$

Constraints (5-1) enforce the requirement for minimum batch size, (V_{ij}^{min}) in order for the batch to be executed and put a limit in the maximum batch size due to unit capacities when task *i* is performed in unit *j* at period *k* under scenario q^k.

Material Balances

$$Input^k(s,q^k) = Input^{k-1}(s,q^k) - d^k(s,q^k) + \sum_{i \in I_s} \rho_{si}^p \sum_{j \in J_i} B^k(i,j,q^k)$$
$$- \sum_{i \in I_s} \rho_{si}^c \sum_{j \in J_i} B^k(i,j,q^k), \quad \forall s \in S, q^k \in Q^k, k > 1 \qquad (5\text{-}2)$$

$$Input^1(s) = Init_s, \qquad s \in S \qquad (5\text{-}3)$$

Material balances (5-2) state that the amount of material of state *s* at the end of period *k* is equal to that at period *k-1* adjusted by any amounts produced or consumed between the period *k* and the amount delivered to the market in period *k*. Constraints (5-3) enforce the material balance at the initial period assuming that the initial input, $Input^1$, is given. The objective of the planning-level problem is to determine the amount of materials after first period, $Input^2$, for which the scheduling-level problem will generate an optimal production schedule in the next step.

Demand Constraints

$$d^k(s,q^k) \geq r^k_{s,q^k} - sk^k(s,q^k), \qquad \forall s \in S, q^k \in Q^k \tag{5-4}$$

Constraints (5-4) express the requirement to produce the maximum amount of state *s* towards satisfying the required demand, ($r^k_{s,q}{}^k$). The amount that cannot be produced ($sk^k_{s,q}{}^k$) is thus denoted as a backorder amount and is considered with an associated cost in the objective function.

Duration Constraints

$$Tp^k(i,j,q^k) \quad = \quad \alpha_{ij} wv^k(i,j,q^k), \ + \ \beta_{ij} B^k(i,j,q^k), \\ \forall i \in I, j \in J_i, q^k \in Q^k \tag{5-5}$$

Constraints (5-5) express the processing time of task *i* in unit *j* given the amount of material being processed. $\alpha_{ij,}$ β_{ij} are assumed the same value as in section 1.1. Note that these constrains are only enforced for one batch at each stage under each scenario q^k.

Time Horizon Constraints

$$\sum_{i \in I_j} n^k(i,j,q^k)\left(\alpha_{ij} \ + \ \beta_{ij} V_{ij}^{\max}\right) + Tp^k(i,j,q^k) \leq \mu^k \times H^k, \\ \forall i \in I, j \in J_i, q^k \in Q^k \tag{5-6}$$

Time horizon constraints (5-6) require that all the tasks performed in unit j have to be completed within the time horizon of each stage H^k. μ^k is the sequence factor, which is used to indicate the effects of the sequence constraints on makespan. The use of the sequence factor is considered to reduce the infeasibilities at the scheduling level where the detailed sequence constraints representing the production recipe are considered. A detailed explanation of the use of sequence factor is given in the next section.

Objective: Minimization of Cost

The objective function consists of minimizing the overall cost during the whole time horizon including raw material cost, backorder cost and operating cost as given by equation (5-7).

$$\begin{aligned} Cost = \sum_{k=1}^{3} & Raw\ Material\ Cost^k + Inventory\ Cost^k \\ & + Backorder\ Cost^k + Operating\ Cost^k \end{aligned} \tag{5-7}$$

In particular raw material cost is given by the following equation.

$$\begin{aligned} Raw\ Material\ Cost = \sum_{k=1}^{3} \sum_{s \in S} & \cos t^k{}_s \sum_{q^1 \in Q^1} w^k{}_{q^k} \sum_{i \in I_s} \rho_{si}^c \\ & \sum_{j \in J_i} \left(n^k(i, j, q^k) \times V_{ij}^{\max} + B^k(i,j,q^k) \right) \end{aligned} \tag{5-8}$$

where w^1=1 for the first stage where no scenarios are considered.

The inventory cost represented by equation (5-9) express the cost of storing product at the end of the second and third period. Since in the planning level, it is hard to calculate the exact inventory time due to aggregation of the orders, we use an approximation of half-length of the consecutive periods to represent the inventory time.

$$Inventory\ \ Cost = \sum_{s \in S} hin^1{}_s \times Input^2(s) * tiv^1 + \sum_{s \in S} \sum_{q^1 \in Q^1} w^1{}_{q^1} \times hin^2{}_s \times Input^3(s, q^1) * tiv^2 \tag{5-9}$$

where tiv^1 and tiv^2 is the time interval from period 1 to 2 and period 2 to 3, respectively equal to half of $H^1 + H^2$ and $H^2 + H^3$.

Backorder cost given by equation (5-10) is considered to penalize the partial order satisfaction.

$$Backorder\ Cost = \sum_{k=1}^{3} \sum_{s \in S} \sum_{q^k \in Q^k} w^k_{q^k} \times b\cos t^k_s \times sk^k(s, q^k) \tag{5-10}$$

Operating cost given by equation (5-11) considers the cost of equipment utilization which is related to a fixed cost f^k and a varying cost v^k.

$$Operating\ \ Cost = \sum_{k=1}^{3} \sum_{i \in I} \sum_{j \in J_i} \sum_{q^k \in Q^k} w^k_{q^k} \times \Big(n^k(i, j, q^k) \times (f^k_{ijq^k} + v^k_{ijq^k} V^{\max}_{ij}) + \Big(f^k_{ijq^k} wv^k(i, j, q^k) + v^k_{ijq^k} B^k(i, j, q^k) \Big) \Big) \tag{5-11}$$

5.2.3 SEQUENCE FACTOR

The sequence factor μ^k represents the effect of sequence constraints in the planning problem. Since the scheduling problem is not simultaneously solved with planning problem, the sequence factor is introduced such that the planning results are close to the scheduling solution. In this section, a general procedure is presented for estimating the sequence factor. However, realizing that a gap always exists between the planning problem solution involving the sequence factor and the short-term scheduling problem, an iterative procedure is developed within the planning and scheduling framework presented in section 4, which dynamically adjusts the sequence factor such that it always represents the best estimation.

Thus the following procedure is developed that provides a reliable estimate. First the planning and scheduling problems are solved for a test case where the planning problem is solved using only one stage and one scenario. The ratio of the objective functions from the solutions of planning problem (Obj^k_{plan}) and the corresponding scheduling problem (Obj^k_{sche}), ($Obj^k_{sche}/Obj^k_{plan}$) is used as an approximation of the sequence factor. If the scheduling problem cannot be efficiently solved a smaller time horizon is considered which will provide a lower bound on the sequence factor since sequence constrains are increasingly more important as the time horizon decreases. Moreover, the LP relaxation can be used to estimate the sequence factor for two reasons: 1) the solution can be achieved very efficiently; 2) the ratio of the objective functions usually is larger than that from solving the original problems since the LP relaxation of the planning problem produces a tighter relaxation. Based on these approximations, a good estimate of the sequence factor can be obtained.

It should be noticed however that the target is not to obtain the exact value of μ^k since its value is updated within the overall proposed framework as explained in the next section.

5.3 SCHEDULING MODEL

The scheduling problem is solved after the solution of planning model to ensure a feasible production schedule for the current period given the production requirement determined by the planning problem. Since planning takes into consideration the future time periods, the production required by scheduling could exceed the orders imposed by the market. Assuming that parameters in the first stage are deterministic, the scheduling problem is solved using a continuous-time formulation introduced in section 1.1. In order to incorporate the planning considerations, constraints (1-6) as well as objective function (1-16) are modified as follows:

Demand Constraints

$$\sum_{n \in N, n \leq n'} d(s,n) \geq r_{s,n'} - sk(s), \quad \forall s \in S, n, n' \in N \tag{5-12}$$

The solution of scheduling problem is required to determine a feasible production schedule satisfying the planning production results. Thus, all the orders are required to be satisfied by their due dates as represented by constraints (5-12). According to these constraint, the production of material s by event point n should satisfy the individual order of product s by the due date that corresponds to event point n' (Ierapetritou et al., 1999). Slack variables sk are considered to represent the amount of backorder that are penalized in the objective function.

Required Production Constraints

In order to consider the requirements imposed by the planning problem, additional production may be required at the current period. This requirement is considered separately in constraints (5-13).

$$\sum_{n \in N, n \leq n'} d(s,n) \geq rp_s - slack(s) \quad \forall s \in S, n, n' \in N \tag{5-13}$$

where rp_s is the production requirement of material s obtained from the solution of planning model.

Objective: Minimization of cost

The objective of the scheduling problem (5-14) is to minimize the slack variables of required production, *slack(s)* as well as all the production cost similarly to the planning model.

$$\begin{aligned} Min \quad & \sum_{s \in S} \cos t_s \sum_{i \in I_s} \rho_{si}^{c} \sum_{j \in J_i} \sum_{n} B(i,j,n) \\ & + \sum_{s \in S} hin_s \times \sum_{n} d(s,n) * tiv^1 + \sum_{s \in S} b\cos t_s \times sk(s) \\ & + \sum_{i \in I} \sum_{j \in J_i} \sum_{n} \left(f_{ij} \times wv(i,n) + v_{ij} B(i,j,n) \right) \\ & + \sum_{s} plt_s \times slack(s) \end{aligned} \tag{5-14}$$

The first term represents in (5-14) the cost of raw materials. The second and third terms consider inventory cost and backorder cost, respectively. Since the time length of storing the inventory in the next period is not known, tiv^l is used in the scheduling problem as well. Operating cost is involved in the fourth term. The last term enforces the consideration of the additional production, where plt_s is a penalty considered to indicate which products or intermediate materials are desired to be produced first.

5.4. SOLUTION FRAMEWORK

In this section, the overall hierarchical framework is presented that addresses the solution of planning and scheduling problem. An iterative procedure ensures consistent results between planning and scheduling stages. The overall flowchart is shown in Figure 5-1.

The proposed framework is based on a rolling time horizon approach, which considers several periods of the entire planning horizon at a time. The optimal production schedule of these periods is determined and a new planning problem is formulated and solved following the same procedure. The advantages of rolling horizon approach are the following:

1) It is adaptive to the dynamic production environment. Long term forecast usually suffers large uncertainty. Therefore when planning over a large time horizon, one may not be able to predict the fluctuating demands or price well and foresee all the disturbances such as machine breakdown or rush orders. Consequently the optimized results won't represent the optimal schedule if these care.

2) Solving planning problem over a large time horizon is usually computationally expensive and the optimal solution is not guaranteed within the affordable computational time.

At each decision-making point, the future periods are grouped into threes stages. Product demands are aggregated at each stage and a scenario tree is generated to represent uncertainty. The formulated planning model thus takes into consideration a number of periods although only the decisions for the current period (stage 1) are implemented. The planning model is solved as a MILP problem, or alternatively utilizing a heuristic by solving the LP relaxation of the original problem and fixing the number of full-size batches. The solution of the planning model could result in the following two cases: 1) the production for the current period could not satisfy the aggregated demand in this period; 2) the production meets or exceeds the aggregated demand.

In the first case, we need to further identify if the shortage is due to capacity limitation or inaccurate parameters in the planning model. Therefore, the demands are disaggregated and the short-term scheduling problem is solved in order to obtain a feasible production schedule. If the scheduling problem generates an optimal schedule that satisfies all the orders, this means that the infeasibility of planning problem is due to an underestimated sequence factor. In this case, the sequence factor is increased and the problem is resolved until the convergence between planning and scheduling models is achieved. The sequence factor is updated according to the following equation:

$$\mu^{l} = Min(\ \mu^{l-1} \times \frac{P_{sche}}{P_{plan}}, 1\ ) \tag{5-15}$$

.

where μ^k is the value of the sequence factor at iteration l, P_{sche}, P_{plan} is the production determined by scheduling and planning models, respectively. Note that the sequence factor should not exceed the value of one since this would mean unrealistic production capacity. If the short-term scheduling solution cannot satisfy all the orders, this means that the problem is infeasible and thus backorders are allowed. The optimal solution is the best production schedule for the current period since the objective function minimizes the amount of backorders. Assuming we want to satisfy these backorders as early as possible, the amount of backorder is added into the market demands for the next period. In both cases, the inventory level at the end of current period is updated and the model is reformulated utilizing the rolling horizon approach so that the schedule of the following period can be determined.

In the second case where the planning solution satisfies all the orders, it might result in additional production beyond the demands. In order to achieve the additional production as well as satisfy the orders, we incorporate constraints (5-12) and (5-13) into the scheduling model. The scheduling problem can lead to one of the following cases:

a) the schedule meets the orders and the additional production requirements. In this case, the solution represents the optimal production schedule for the current period and the inventory at the end of current period is updated;

b) the optimal production from the scheduling problem cannot satisfy the orders although the planning model suggest the opposite. This means that the sequence factor underestimate the effects of sequence constraints in production capabilities of the plant. Consequently in this case, the result of the scheduling problem is accepted as optimal allowing backorders adjusting the demand for the next period. Since the results for this period reveals an underestimate of the sequence factor, to avoid such case in the future the sequence factor is updated following equation (5-15). The inventory is updated as well;

c) the optimal production satisfies all the orders but not the additional production determined by the planning model. In this case, the sequence factor is adjusted following equation (5-15) and the planning model is resolved for the same time period to determine more realistic production targets.

At each time point, this iterative procedure continues until convergence is achieved between the planning and scheduling problems. Demand and inventory are updated and the same procedure is followed for the next time period in the rolling horizon approach. Two examples are presented in the next section to illustrate the details of the steps of the proposed methodology.

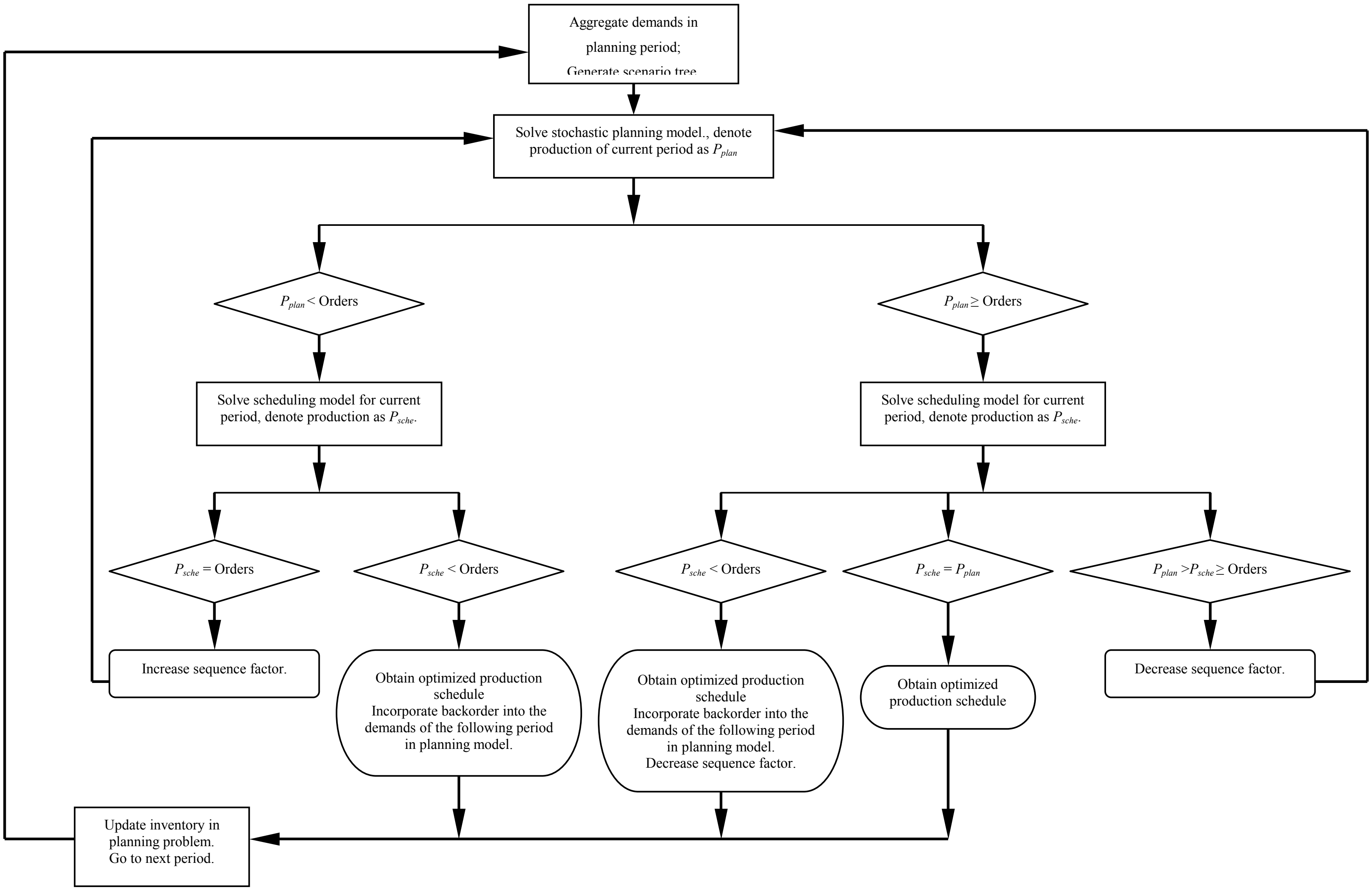

Figure 5-1: Flowchart of Proposed Hierarchical Planning Optimization

5.5 PLANNING EXAMPLE

In this example (section 1.2), a planning problem with time horizon of 10 days is considered where thirty 8-hour schedules need to be determined dynamically. The actual demand is shown in Figure 5-2. In this example all the orders have the due dates at the end of 8-hour period. The planning model involves three stages corresponding to 8, 16 and 48 hours, respectively. Three demand scenarios are considered for each branch of the scenario tree corresponding to high, average and low level of product demand based on demand forecasting. Therefore, there are 3 scenarios for stage 2 and a total of 9 scenarios for stage 3.

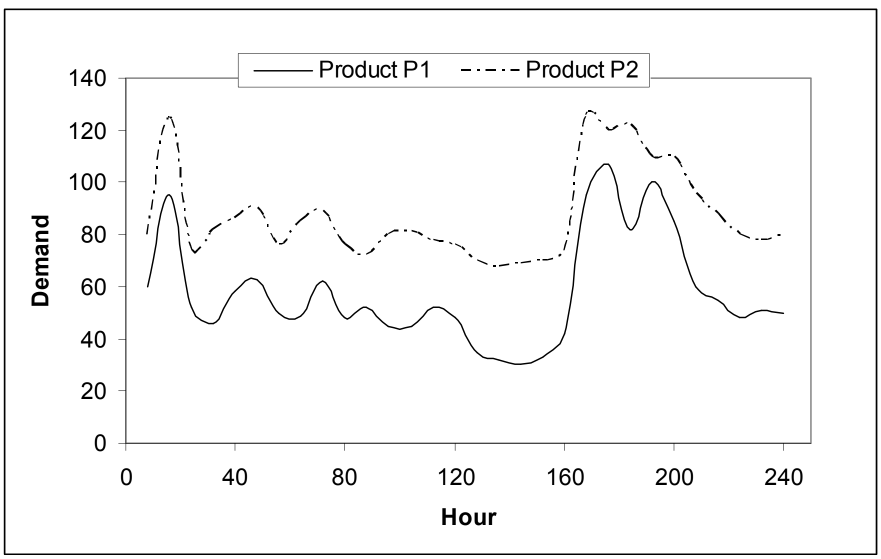

Figure 5-2: Demand of Planning Example

Demand peaks appear at the beginning and later periods, which exceeds the production capacity of the plant. By applying the proposed framework, we expect to leverage the storage capacity such that a smooth production schedule can be generated for the entire time horizon with the minimum amount of backorders.

Following the proposed approach, first we aggregate the orders for each stage. In the first planning model, the demands of the second and third 8-hour periods are aggregated for stage 2 and those of the next six 8-hour periods are aggregated for stage 3. Table 5-1 shows the market demands and aggregated demands for the first two planning problems. Note that in the second planning problem, the demand of stage 1 should also consider the backorder resulted from the previous period. Uncertainty is considered using the introduced scenarios. For example, in the first planning problem we use forecasts for three demand scenarios of stage 2 with values of 142, 147 and 152, respectively; while 9 demand scenarios are created for stage 3. Figure 5-3 shows the aggregated demands and their variation range of scenarios for the first planning problem.

8-hour period	Market demand		Aggregated demands for the first planning problem		Aggregated demands for the second planning problem	
	P1	P2	P1	P2	P1	P2
1	60	80	60	80		
2	95	125			95+ backorder	125+backorder
3	52	74	147	199		
4	46	82			98	156
5	58	86				
6	63	90				
7	50	76				
8	49	85				
9	62	89	328	508		
10	48	76			330	502

Table 5-1: Aggregated Demands for Planning Problem

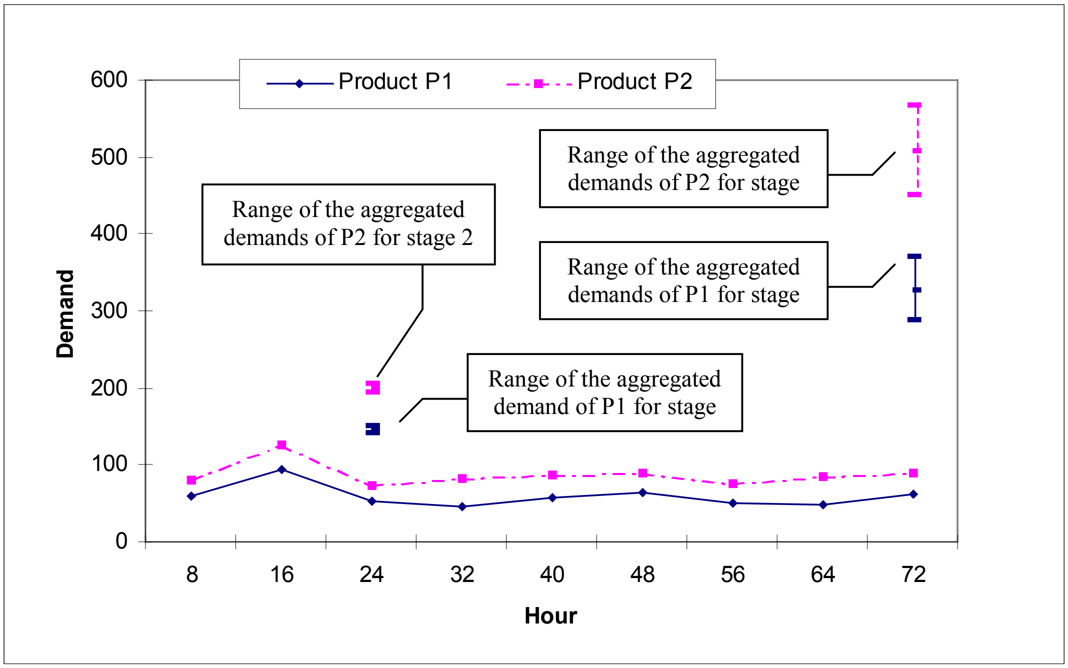

Figure 5-3: Aggregated Demands of First 72 Hours

The next step involves the evaluation of an initial value of the sequence factor for each stage in order to solve the planning problem. As presented in section 5.2.3, a test problem is used for an 8-hour time horizon and the planning and scheduling problems are solved to minimize a makespan required to fulfill the orders. The optimal makespan are 6.25 and 7.85 hours for the planning and scheduling problem, respectively. The ratio of these values is considered to represent the difference of production capacity between planning and scheduling models. Therefore a value of 0.80 is used as an initial approximation of the sequence factor for the current period (stage 1). Similarly the sequence factors for stage 2 and stage 3 are determined. The sequence factor for stage 2 is initialized with a value of 0.90. Since it is computationally expensive to solve the short-term scheduling problem for 48 hours, the LP relaxation is solved for stage 3 providing an approximation of 0.99 for the sequence factor for this stage. Thus for stage 3 a value of 0.95 is used as the initial sequence factor. It should be pointed out that since stages 2 and 3 represent the future for the current decision, the accuracy of sequence factor doesn't play an important role in the framework.

The planning problem is then solved resulting in an optimal production for the first 8-hour period of 30.51 units of IAB, 7.27 units of IBC, 61.82 units of P1 and 80 units of P2, although the product demands only require 60 units of P1 and 80 units of P2. The additional production can be perceived as the response to forthcoming demand peak in the second 8-hour period. These production requirements are considered into the scheduling model as additional production and the problem is solved to optimality as a MILP problem. The results satisfy the required production as well as the market demands. Therefore, the optimal schedule is obtained and the excessive production is considered as initial inventory for the next period. Before rolling into the next planning period, the newest data and information such as forecast scenarios and machine availability are incorporated. Assuming all the operating conditions remain the same, we have a demand of 95 units of P1 and 125 units of P2 for the second 8-hour period (stage 1 in the new planning problem). The optimal solution of the planning model however, results in a production of 12.72 units of IAB, 54.78 units of P1 and 125 units of P2 that doesn't satisfy the demands. The short-term scheduling problem is then solved which determines a production of 39.80 units of IAB, 63.72 units of P1 and 107.43 units of P2. A shortage of 31.28 units of P1 and 17.57 units of P2 are considered as backorders and added into the market demands in the next period (the third 8-hour period). Since a relatively high backorder cost is used in this example, the objective functions of both planning and scheduling problems result in minimizing the backorders. Thus the resulted production schedule represents the optimal decisions. The inventory is thus updated for intermediate IAB at the beginning of the next period with a value of 39.80 units and the planning horizon is updated to include the next 8-hour period and a new planning problem is solved. This process continues until the whole time horizon of 30 periods is considered. Backorders are obtained at period 2, 3, 24, 25 and 26. The production schedule of the first sixteen hours are shown in Figure 5-4 and the demands and inventory level as well as backorders of the entire planning horizon are shown in Figure 5-5, where an effort of increasing inventory against the demand peak is clearly illustrated.

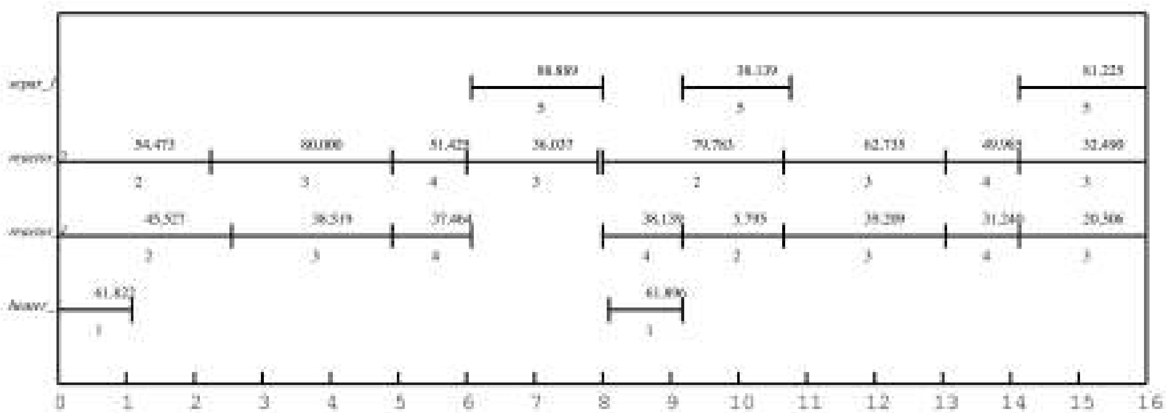

Figure 5-4: First 16 Hours Production Schedule of Planning Example

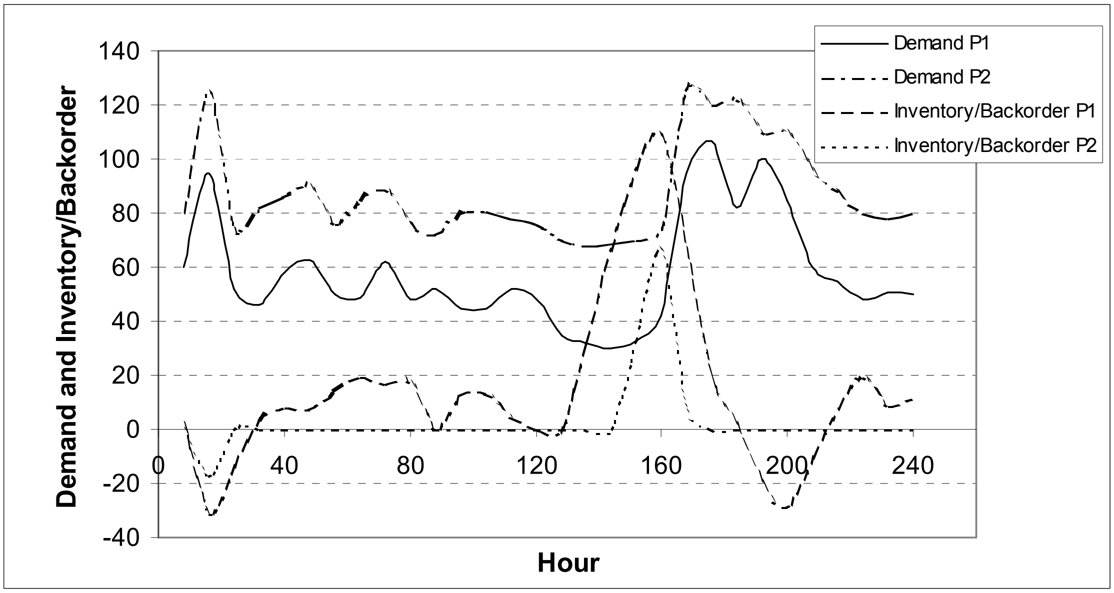

Figure 5-5: Demand, Inventory/Backorder of Planning Example

These results are compared with two other approaches. In the first approach, only short-term scheduling model is used to solve for each 8-hour production schedule. This is the case where scheduling cannot foresee the future demand and it is only trying to satisfy the order for the current period. There are total twelve periods that encounter backorders (period 2, 3, 21, 22, 23, 24, 25, 26, 27, 28, 29 and 30). The second approach uses short-term scheduling model as well, however it considers two 8-hour time periods although only the decision for the first 8-hour period is implemented. This approach exploits the same idea of planning model but it cannot involve more than two periods due to the complexity of the short-term scheduling problem. The production schedule has nine backorder periods (period 2, 21, 22, 23, 24, 25, 26, 27 and 28) and the computation CPU time is two orders of magnitude larger than the proposed approach.

Consequently the overall production schedule exhibits higher backorder cost while it decreases the less expensive inventory cost. Table 5-2 summaries these results.

	One-period scheduling approach	Two-period scheduling approach	Proposed approach
Num. of backorder periods	12	9	5
Obj.	112,011.9	46,002.8	28,804.1
CPU (sec.)	836.8	111,104.5*	1,016.9

* computational limit is 5,000 CPU sec. for each MILP problem

Table 5-2: Results Comparison

Although the results depend on the choice of the initial sequence factor, the iterative process in the proposed approach can prevent from using an impractical sequence factor. In the following case, an initial sequence factor for stage 1 with value of 0.50 is used in the planning model, which obviously underestimate the production capacity. As a consequence, the planning model can only achieve 32.50 units of P1 and 62.68 units of P2 at its optimum, thus resulting in a backorder of 27.50 units of P1 and 17.32 units of P2. The scheduling model, however is able to generate a production schedule that satisfies the market demand (60 units of P1 and 80 units of P2) with an additional 27.78 units of IAB as side product. The discrepancy is due to the value of sequence factor and thus the planning model needs to be revisited. We compare the production of the final products as well as intermediates for both planning and scheduling models, and use the ratio as an adjustment to the original sequence factor. Therefore a new sequence factor of

0.74 is used in the planning model. The new optimal solution of the planning problem increase the production due to the updated sequence factor, but there is still a shortage of 12.12 units of P1 although the scheduling problem shows no backorder. The sequence factor is updated again with the new production ratio of 1.09 and obtains a value of 0.82. In the new iteration, planning results satisfy all the market demand and require an additional production of 12 units of IAB, 4 units of IBC, 8.75 units of IE, 4 units of P1 and 19 units of P2. The scheduling problem however cannot make all of these additional requirements. Again this indicates that an optimistic sequence factor is used in the planning model. Following the same adjustment, the sequence factor is brought down to a value of 0.79 by using the new production ratio of 0.96. With the new sequence factor, the planning and scheduling results become consistent and the approach moves to the next planning period. In the next case, a sequence factor of 0.95 is initially considered for the stage 1 of the planning problem. The optimal solution requires an additional production of 4.78 units of IAB, 4 units of IBC, 17.78 units of IE, 4 units of P1 and 19 units of P2 on top of the market demand. Due to the overestimated sequence factor, the scheduling problem couldn't achieve this production although it can satisfy the market demand. The sequence factor is then adjusted to 0.85 based on the production and the planning model is resolved. Same iterative process continues until the planning and scheduling results converge with a final sequence factor of 0.80. As shown from these cases, the mechanism in the proposed approach can effectively adjust the sequence factor to a reasonable value that reflects the real production capacity.

From the solution of this example, it is illustrated that the proposed framework can effectively considers a long-term trend in the planning model while optimizing the detailed production schedule for the current period. The communication between planning and scheduling problem in the framework ensures consistent results.

5.6 SUMMARY

This chapter presents a hierarchical solution approach for solving the dynamic production planning and scheduling problems. The planning model involves a scenario-based multi-stage formulation while the scheduling model generates detailed schedule for the first planning period using a continuous-time formulation. A discounted parameter, sequence factor, is used in the planning model and an iterative process is developed in the proposed framework that ensures the consistency of the optimality of planning and scheduling problems.

NOTATION

Planning model

Indices

i	task
j	unit
q^k	scenario at period k
s	state

Sets

I	tasks
I_j	tasks that can be performed in unit j
I_s	tasks that process state s and either produce or consume
J	units
J_i	units that are suitable for performing task i
Q^k	scenarios at period k
S	states

Parameters

V_{ij}^{min}	minimum amount of material processed y task i required to start operating unit j
V_{ij}^{max}	maximum amount of material processed by task i required to start operating unit j
r^k_{s,q^k}	market requirement for state s at the end of period k under scenario q^k
ρ^p_{si}, ρ^c_{si}	proportion of state s produced, consumed from task i, respectively
α_{ij}	constant term of processing time of task i at unit j
β_{ij}	variable term of processing time of task i at unit j
H^k	time horizon of period k
$price^k_s$	price of state s at period k
f^k_{ij,q^k}	fixed cost of task i at unit j at period k
v^k_{ij,q^k}	variable cost of task i at unit j at period k
$cost^k_s$	cost of state s at period k
$bcost^k_s$	backorder cost of state s at period k

hin^k_s	inventory cost of state s at period k
$Init_s$	initial input of state s at the first period
wk_q^k	weight coefficient of scenario q^k at period k
μ^k	sequence factor for period k

Variables

$wv^k(i,j, q^k)$	binary variables that assign task i in unit j at period k
$b^k(i,j\ q^k)$	amount of material undertaking task i in unit j at period k
$d^k(s, q^k)$	amount of state s being delivered to the market at period k
$Tp^k(i,j, q^k)$	processing time of task i in unit j at period k
$Input^k(s, q^k)$	input of state s at period k
$sk^k(s, q^k)$	backorder of state s at period k
$n^k(i,j, q^k)$	number of full batches of task i in unit j at period k

Scheduling model

Indices

i	task
j	unit
n	event point representing the beginning of a task
s	state

Sets

I	tasks
I_j	tasks that can be performed in unit j
I_s	tasks that process state s and either produce or consume
J	units
J_i	units that are suitable for performing task i
N	event points within the time horizon
S	states
IS	subset of all involved intermediate states s

Parameters

V_{ij}^{min} minimum amount of material processed y task i required to start operating unit j

V_{ij}^{max} maximum amount of material processed by task i required to start operating unit j

ST_s^{max} available maximum storage capacity for state s

$r_{s,n}$ market requirement for state s at event point n

rp_s required production for state s from planning results

ρ^p_{si}, ρ^c_{si} proportion of state s produced, consumed from task i, respectively

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

β_{ij} variable term of processing time of task i at unit j

$price_s$ price of state s

$f_{ij,}$ fixed cost of task i at unit j

v_{ij} variable cost of task i at unit j

$cost_s$ cost of state s

$bcost_s$ backorder cost of state s

hin_s inventory cost of state s

plt_s penalty term for the slack variable of required production

Variables

H time horizon

$wv(i,n)$ binary variables that assign the beginning of task i at event point n

$yv(j,n)$ 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

$d(s,n)$ amount of state s being delivered to the market at event point n

$ST(s,n)$ amount of state s at event point n

$STIN(s)$ amount of state s imputed initially

$T^s(i,j,n)$ time that task i starts in unit j at event point n

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

$Slack(s)$ slack variable of required production for state s

$sk(s)$ slack variable of orders for state s

CHAPTER 6

LAGRANGEAN DECOMPOSITION USING AN IMPROVED NELDER-MEAD APPROACH FOR LAGRANGEAN MULTIPLIER UPDATE

Lagrangean decomposition is critically dependent on the method of updating the Lagrangean multipliers. This chapter presents a Lagrangean decomposition approach based on Nelder-Mead optimization algorithm to update the Lagrangean multipliers. The main advantage of the proposed approach is that it results in improved objective function values for the majority of iterations. The efficiency of the proposed approach is illustrated with solution of scheduling problems.

6.1 INTRODUCTION

The complexities of realistic problems require the use of large-scale models that often prevent the convergence to the optimal solution. Lagrangean relaxation and Lagrangean decomposition are promising decomposition techniques that reduce the problem size and achieve solution in affordable computational time as shown in chapter 3. As an extended research to the previous scheduling work, this chapter discusses in depth the Lagrangean technique and presents an alternative Lagrangean multiplier updating method.

Lagrangean relaxation was original developed by Held and Karp (1970) and successfully applied to many optimization problems such as production scheduling (Fisher, 1973), planning (Graves, 1982 and Gupta and Maranas, 1999) and lot-sizing problems (Thizy, 1985 and Diaby et. al. 1992). The idea of Lagrangean relaxation is based on the characteristic that mathematical problem formulations involve “hard” constraints, the existence of which increases the complexity of problem. As shown in Figure 6-1 assuming that the first set of constraints are the complicating ("hard") constraints, Lagrangean relaxation proceeds by relaxing these constraints and penalizing the constraint violation in the objective function thus reducing the computational complexity of the solution since the problem with the remaining constraints is easier to solve.

$$
\begin{array}{lll}
(P) \quad \max \quad z = cx & (LR_u) = \max \quad cx + u(b - Ax) & \\
\quad s.t. \;\; Ax \le b \qquad \Longrightarrow & \qquad s.t. \quad Cx \le d \qquad \Longrightarrow & (LR) \quad \min\limits_{u} \, (LR_u) \\
\qquad\quad Cx \le d & \qquad\qquad x \in X, b \ge 0 & \\
\qquad\quad x \in X & &
\end{array}
$$

Figure 6-1: Lagrangean Relaxation

Since the set of Lagrangean multipliers are chosen to be non-negative ($u \geq 0$), for every optimal solution x of the original optimization problem (P) we have

$V(LR_u) \geq V(LR) \geq V(P)$

where the operator $V(.)$ denotes the optimal value. Therefore the resulting objective function from Lagrangean relaxation is an upper bound of the original optimization problem. Let operator $Co(.)$ denotes the convex hull and P^* represents the following problem:

$$
\begin{aligned}
(P^*) \quad & \max \quad z = cx \\
s.t. \;\; & Ax \le b \\
& x \in Co\left\{ Cx \le d, x \in X \right\}
\end{aligned}
$$

It can be further shown that (P^*) and (LR) are duals (Geoffrion, 1974). Thus:

$V(LP) \geq V(LR) \ = V(P^{*}) \geq V(P)$

where (*LP*) is the LP relaxation of problem (*P*). If the multipliers of the complicating constraints from the optimal LP relaxation solution are used to solve (LR_u), the solution corresponds to an upper bound of the original problem at least as tight as tight as the bound from (*LP*), i.e. $V(LP) \geq V(LR_u)$. The integrality property is stated as follows: the optimal value of (LR_u) is not changed by dropping the integrality condition on the x variables, i.e. $Co\{Cx \leq d,\ x \in X\} = \{Cx \leq d,\ x \in X\}$. Geoffrion (1974) proved that only when this property holds, the following equalities hold.

$V(LP) = V(LR) \ = V(P^{*}) \geq V(P)$

However for practical problems, this integrality property usually doesn't hold, thus allows the application of Lagrangean relaxation to provide a tighter bound.

A drawback of the Lagrangean relaxation is that the problem loses its original structure since the complicating constraints is removed from the constraints set and embedded into the objective function. As a method to avoid this, Lagrangean decomposition is presented by Guignard and Kim (1987). It can be regarded as an extension of the Lagrangean relaxation since instead of relaxing constraints it is based on the idea of relaxing a set of variables responsible of connecting important model components. Considering the problem (*P*) shown in Figure 6-2 the set of variables x that connect the set of constrains are duplicated by introducing an identical set of variables y and an equality constraint $x=y$ as shown in Problem ($P^{'}$). Lagrangean relaxation is then applied to the new set of equality constraints resulting in problem (LD_u) which can be further decomposed into two subproblems (LD_u^{1}) and (LD_u^{2}) as shown in Figure 6-2.

$$I: P \qquad\qquad II: \; P' \qquad\qquad III: \; LD_u$$

$$\left\{\begin{array}{rl} \max & z = cx \\ s.t. & Ax \le b \\ & Cx \le d \\ & x \in X \end{array}\right\} \Longleftrightarrow \left\{\begin{array}{l} \max \quad cx \\ s.t.\; Ax \le b \\ \quad Cy \le d \\ \quad x = y \\ x \in X,\, y \in X \end{array}\right\} \Longrightarrow \left\{\begin{array}{c} (LD_u) = \max \quad cx + u(y - x) \\ s.t.\; Ax \le b \\ Cy \le d \\ x \in X,\, y \in X \end{array}\right\} \Longleftrightarrow$$

$$\begin{array}{cc} LD_u^1 & LD_u^2 \end{array}$$

$$\left\{\begin{bmatrix} \max\,(c-u)x \\ s.t.\; Ax \le b \\ x \in X \end{bmatrix} + \begin{bmatrix} \max\, uy \\ s.t.\, Cy \le d \\ y \in X \end{bmatrix}\right\} \Longrightarrow (LD) \quad \min_u (LD_u)$$

Figure 6-2: Lagrangean Decomposition

Denoted u° as the optimal Lagrangean multiplier to (*LR*). Let $v^\circ = u^\circ A$ and x°, y° be the optimal solution of (LD_u), then

$$V\left(LD_{v^\circ}\right) = \begin{bmatrix} \max\; (c - v^\circ)x \\ s.t.\; Ax \le b \\ x \in X \end{bmatrix} + \begin{bmatrix} \max\; v^\circ y \\ s.t.\, Cy \le d \\ y \in X \end{bmatrix}$$

$$= \begin{bmatrix} \max\; (c - u^\circ A)x \\ s.t.\; Ax \le b \\ x \in X \end{bmatrix} + \begin{bmatrix} \max\; u^\circ Ay \\ s.t.\, Cy \le d \\ y \in X \end{bmatrix}$$

$$\begin{aligned} &= (c - u^\circ A)x^\circ + u^\circ A y^\circ \\ &= (c - u^\circ A)x^\circ + u^\circ b + u^\circ(Ay^\circ - b) \\ &= V\left(LR_{u^\circ}\right) + u^\circ(Ay^\circ - b) \end{aligned}$$

Therefore *V(LD)* $\le$ *V(LR)*, which means that the upper bound generated from Lagrangean decomposition is always at least as tight as that from Lagrangean relaxation.

Similarly, if integrality property holds in one of the subproblems, for instance, $Co\{Cx \le d, x \in X\} = \{Cx \le d, x \in X\}$, then (*LD*) is equivalent to (*LR*) and $V(LD) = V(LR)$. If integrality property holds in both subproblems, i.e. $Co\{Ax \le b, x \in X\} = \{Ax \le b, x \in X\}$ and $Co\{Cx \le d, x \in X\} = \{Cx \le d, x \in X\}$, then (*LD*) is equivalent to (*LP*) and $V(LD) = V(LP)$. Otherwise (*LD*) is strictly tighter than (*LR*) as is usually the practical case. Adding surrogate constraints, which means both subproblems have overlapped constraints, can further tighten the bound of Lagrangean decomposition. However, this increases the problem complexity and thus the solution computational time. Figure 6-3 provides a geometric interpretation of the Lagrangean decomposition (Guignard, 1987).

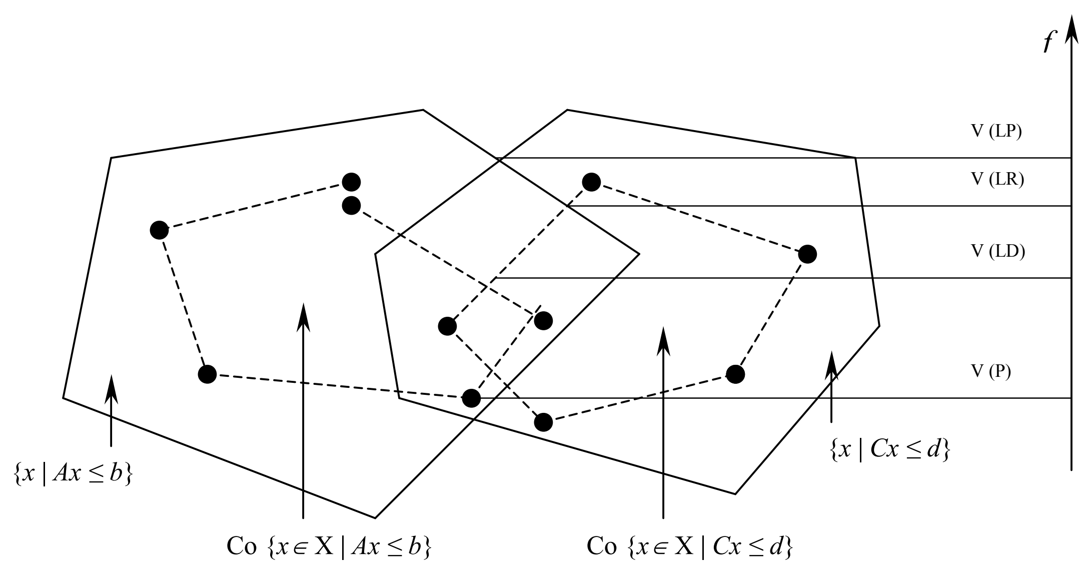

Figure 6-3: Geometric Interpretation of Lagrangean Decomposition

The main difficulty in optimizing Lagrangean decomposition is that it is non-differentiable since the inner maximization problem generally has multiple optimal solutions. The most common approach to deal with this lack of differentiability is the subgradient method (Polyak, 1967, Marin and Pelegrin, 1998, Equi et. al. 1997 and Rana and Vickson, 1991). The subgradient method utilizes the distance between the objective value at the current iteration Z^k and the estimated optimum Z^* to calculate a step size t^k which is used to update the Lagrangean multipliers as follows:

$$u^{k+1} = u^k + t_k (y^k - x^k) \tag{6-1}$$

$$t_k = \frac{\lambda(Z^k - Z^*)}{\left\| y^k - x^k \right\|^2} \qquad 0 \le \lambda \le 2 \tag{6-2}$$

where superscript k corresponds to the iteration number and λ is a scaling factor of the step size to control the convergence, normally considered to be between 0 and 2.

A number of problems however arise with the use of subgradient method. Lagrangean decomposition is based on duality theory, and theoretically the method converges the dual variables to the same value (more generally, $u(x\text{-}y)=0$), which results in the optimal solution of the original optimization problem. However, in practice using subgradient method is reported to have unpredictable convergence (Guignard, 2003).

For some problems, subgradient method generates monotonic improvement in Lagrangean objective function and the convergence is quick and reliable. While other problems result in erratic multiplier sequence and the Lagrangean function value keeps deteriorating. According to the complementary slackness condition, subgradient method updates the Lagrangean multiplier u until the convergence of the dual variables. The gap between the decomposition variables x and y may exist even when the (*LD*) converges to the optimal objective value of the original optimization problem. We will illustrate this case in the following section through solution of an example problem. Other drawbacks of subgradient method are (a) the lack of convergence criterion, (b) the lack of estimate of the optimal objective Z^* and (c) dependency to the heuristic choice of the step size sequence (Crowder, 1976). Due to these problems the subgradient method is not stable when applied to large-scale problems. Another issue related to Lagrangean decomposition is that the dualized variable set must be appropriately chosen so that the resulting subproblems are easy to solve and the solution converges fast, which is case-dependent (Orero and Irving, 1997, Gupta and Maranas, 1999).

Bundle method is an extension of subgradient method presented by Lemarechal (1974) and Zowe (1985). This method considers improving Lagrangean function as well as staying close to the approximated optimal solution. At each iteration the method either update the Lagrangean multipliers or improves the function approximation. A trade-off exists in the small region within which the bundle method allows to move and the small size of the bound improvement.

Realizing these deficiencies a number of techniques have been developed to update the Lagrangean multipliers. Constraint generation method (Kelly, 1960) considers a family of k existing solution (x^k, y^k) and generates a new Lagrangean multiplier u by solving a restricted LP master problem MP^k:

$$\begin{aligned} &\min_{u,\eta} \quad \eta \\ &s.t. \quad \eta \geq cx^k + u(y^k - x^k), \quad k = 1,\ldots,K \end{aligned}$$

The resulting u is then used in the LD_u and a new cut of (x^{k+1},y^{k+1}) is obtained from solving LD_u, which is added into the master problem (Guignard, 1995). The process terminates when $V(MP^k)=V(LD_u)$. However there is no guarantee that the new

Lagrangean multiplier will generate an improved solution, thus the problem of cycling needs to be resolved. This method also depends on the cuts of (x^k, y^k) in the master problem.

Multiplier adjustment method, also referred as dual ascent/descent algorithm was presented by Bilde and Krarup (1977) and reported successful application by Erlenkotter 1978), Fisher and Hochbaum (1980), Fisher and Kedia (1990) and Guignard and Rosenwein (1990). This method generates a sequence of u^k by using the following relationship:

$$u^{k+1} = u^k + t^k d^k \tag{6-3}$$

where t^k is a positive scalar and d^k is a descent direction. d^k is usually determined from form a finite set of directions by evaluating the directional derivative of (LD_u). Typically the direction of the steepest descent is chosen and the step size t^k is chosen to minimize $V(LD_{u^k+td^k})$. Although this method is reported to work better than the subgradient method for some problems (Erlenkotter, 1978), the set of directions to choose from may involve specific knowledge of the problem such that the number of directions is minimized but still contains direction to descent. A good review of the methods solving for Lagrangean multipliers is given by Guignard (2003).

These considerations initiate our efforts towards an improved method for updating the Lagrangean multipliers based on a direct search in Lagrangean multiplier space. In the next section a motivating example illustrates the need for an improved updating method for the Lagrangean multipliers. In section 6.3, the proposed modified Lagrangean decomposition method is presented whereas examples are given in section 6.4.

6.2 MOTIVATING EXAMPLE

The following mixed integer linear programming problem is considered in Wu and Ierapetritou (2003).

$$
\begin{aligned}
& max\ \ 2x_1 + 3x_2 + 4x_3 + 4x_4 \\
& s.t. \\
& x_1 + 2x_2 \leq 8 \qquad (1) \\
& 4x_2 + 3x_3 \leq 13 \qquad (2) \\
& 2x_1 + 5x_4 \leq 11 \qquad (3) \\
& x_3 + x_4 \leq 6 \qquad (4) \\
& x_1, x_2, x_3, x_4 \in Z^+ \cup \{0\}
\end{aligned}
$$

The optimum objective value is 26 that corresponds to two equivalent solutions (x_1,x_2,x_3,x_4) = (5,0,4,0) and (x_1,x_2,x_3,x_4) = (3,0,4,1). Subgradient method is used for the solution of this problem with the initial value of λ equal to 2. The updating strategy halves λ when the objective value is not improving for 5 iterations. As described in previous section, the choice of the decomposing variables is case dependent and affects the convergence of the Lagrangean decomposition. Table 6-1 shows the results for this example when different variables are used for the decomposition.

Decomposing variables	x_1 , x_2	x_2 , x_3	x_1 , x_4	x_3 , x_4
Optimal solution (x_1 , x_2 , x_3 , x_4)	(5,0,4,0)	(5,0,6,0)	(0,0,4,0)	(5,0,4,0)
Dual variables	y_1 , y_2	y_2 , y_3	y_1 , y_4	y_3 , y_4
Dual variable values	(0,4)	(0,4)	(3,1)	(6,0)
u_1, u_2	(0.00, 0.00)	(3.37, 4.00)	(2.00, 4.37)	(0.00, 0.00)
Optimal objective function value using subgradient method	26.012	26.000	26.279	26.000
CPU (sec.)	1.19	0.27	2.40	0.12

Table 6-1: Subgradient Method Results of Motivating Example

Although the Lagrangean decomposition converges to the optimal solution, the primary variables are not guaranteed to be optimal. For example, the Lagrangean decomposition returns the optimal objective function value of 26 when x_2 and x_3 are decomposed, the solution of Lagrangean decomposition however, corresponds to (x_2,x_3) = (0,6) whereas (y_2,y_3) = (0,4). The subgradient method terminates since $Z^k = Z^*$. Therefore we obtain an upper bound, which converges to the optimal objective function of the original problem but the variables do not converge to the optimal values and a gap between the dualized variables exists. When x_1 and x_2 are used for decomposition as follows:

$$\begin{array}{llcll}
max & 4x_3 + 4x_4 + (2 - u_1)x_1 + (3 - u_2)x_2 & & max & u_1 y_1 + u_2 y_2 \\
s.t. & & + & s.t. & \\
 & 4x_2 + 3x_3 \le 13 & & & y_1 + 2y_2 \le 8 \\
 & 2x_1 + 5x_4 \le 11 & & & y_1, y_2 \in Z^+ \cup \{0\} \\
 & x_3 + x_4 \le 6 & & & \\
 & x_1, x_2, x_3, x_4 \in Z^+ \cup \{0\} & & &
\end{array}$$

Lagrangean decomposition converges to a suboptimal solution of 26.012. This results from the fact that subgradient method use step size t_k to calculate the updated Lagrangean multiplier. So when the solution of $(x_1, x_2, x_3, x_4, y_1, y_2) = (5, 0, 4, 0, 0, 4)$ is generated, the duality gaps of x_1,y_1 and x_2,y_2 forces u_1 and u_2 in the area where the same solutions are always generated until the step size becomes too small and the algorithm terminates. Table 6-2 lists the details for each iteration.

Iteration	$(x_1, x_2, x_3, x_4, y_1, y_2)$	(u_1, u_2)	t^k	(LD_u)
1	(0 ,0 ,4 ,2 ,8 ,0)	(1.000, 1.000)	1.0000	32.000
2	(5 ,0 ,4 ,0 ,0 ,4)	(0.000, 1.000)	0.1875	30.000
3	(0 ,0 ,4 ,2 ,8 ,0)	(0.976, 0.220)	0.1951	31.805
4	(5 ,0 ,4 ,0 ,0 ,4)	(0.000, 0.220)	0.1814	26.878
5	(5 ,0 ,4 ,0 ,8 ,0)	(0.214, 0.048)	0.0428	26.642
6	(5 ,0 ,4 ,0 ,0 ,4)	(0.000, 0.048)	0.1428	26.193
7	(5 ,0 ,4 ,0 ,8 ,0)	(0.024, 0.029)	0.0047	26.071
8	(5 ,0 ,4 ,0 ,0 ,4)	(0.000, 0.029)	0.0078	26.118
...	…	…	…	…
32	(5 ,0 ,4 ,0 ,0 ,4)	(0.004, 0.008)	0.0000	26.012

Table 6-2: Intermediate Results from Iterations

However the performance of Lagrangean decomposition depends on the value of Z^* used as shown in Table 6-3 where the results of this example are illustrated using $Z^*=27$. Let's consider the case where x_3 and x_4 are dualized using $Z^*=27$, the Lagrangean decomposition results in an objective function value of 26.462 compared to the optimal value of 26 when $Z^*=26$ is used (Table 6-1). In the same example when Z^* is estimated as 25, the Lagrangean decomposition converges to the optimal value of 26 as the lowest upper bound but requires 85 iterations compared to 4 iterations required when Z^*=26 is used. Table 6-4 illustrates the results for Z^*=25.

Decomposing variables	Optimal objective function value using subgradient method
x_1, x_2	26.816
x_2, x_3	26.500
x_1, x_4	27.000
x_3, x_4	26.462

Table 6-3: Lagrangean Decomposition Results Based on Estimation of Z^*=27

Decomposing variables	Optimal objective function value using subgradient method
x_1, x_2	26.00
x_2, x_3	26.00
x_1, x_4	26.00
x_3, x_4	26.00

Table 6-4: Lagrangean Decomposition Results Based on Estimation of Z^*=25

Such behavior is observed also with initial λ equal to 1 when subgradient method is used to update the Lagrangean multipliers, therefore the need of alternative methodology becomes imperative for the efficient utilization of Lagrangean decomposition in large-scale problem describing realistic case studies.

6.3 PROPOSED APPROACH

The proposed approach uses a direct search method to update the Lagrangean multipliers in order to improve the performance of the Lagrangean decomposition. The main idea is that given a fairly good estimation of Lagrangean multipliers, only the promising directions need to be searched. Thus the computational complexity decreases and the objective of the Lagrangean decomposition is improved at each iteration.

Nelder-Mead method (Nelder and Mead, 1965) is used to determine the promising search directions since it is proven to be very efficient direct search algorithm. For a function of n variables, the algorithm maintains a set of $n+1$ points forming the vertices of a simplex or polytope in n-dimensional space. The result of each iteration is either (1) a single new vertex which replaces the one having the worst function value in the set of vertices for the next iteration, or (2) if a shrink is performed, a set of $n+1$ new points form the simplex at the next iteration. Four scalar parameters must be specified to define a complete Nelder-Mead method: coefficients of reflection, expansion, contraction, and shrinking. As every direct search method Nelder-Mead method has the advantage of not requiring derivative computations, but they tend to be efficient in relatively low dimensions. The details of Nelder-Mead algorithm are given in the appendix.

In order to be able to efficiently use Nelder-Mead method we need to determine a good initial set of Lagrangean multipliers in order to reduced the search space. Moreover, the promising new search directions should be easily determined. These two questions are addressed as follows. As a common practice an effective way to generate an initial set of Lagrangean multipliers for integer linear problems is to use the dual values of LP relaxation. This is because the Lagrangean decomposition problem is equivalent to LP relaxation if we drop the integrality constraints, thus most of these values are already near the optimum and we only need to examine the reduced Lagrangean multiplier space for the next update. The second question is most critical to the proposed algorithm. Promising search direction is defined as the direction resulting in improvement of the objective function. In order to adjust the Lagrangean multipliers independently, the axes in the Lagrangean multiplier space are used as the possible directions. Thus we have *n* orthogonal directions to search where *n* is the number of Lagrangean multipliers and at each iteration we should choose among the directions that lead to the objective function improvement. In practice the directions examined are much less than the actual number of Lagrangean multipliers because of their good initial values.

The steps of the proposed approach are shown in Figure 6-4. In particular first the LP relaxation of the original optimization problem is solved and the Lagrangean multipliers are initialized as the dual values of the corresponding dual equality constraints. Assuming that there are *n* pairs of dual variables, the Lagrangean multiplier space has *n* dimensions. To generate the initial points for Nelder-Mead algorithm, we fix the n-1 variables at their original values and change only one dimension by a value of $\pm\Delta u$, which is α times the original value $\Delta u = \alpha * u_0$. In this way two new points are generated. We apply the same procedure for all dimensions and generate $2n$ points. Each point corresponds to a set of Lagrangean multipliers. The next step is to solve $2n+1$ Lagrangean decomposition problems associated with these points and sort them based on their objective values of Lagrangean decomposition problem. Although this might be a time-consuming step, a potential advantage is that such multi-direction search can be computed in parallel since these problems are completely independent of each other. Those with improved objective values are selected to form the reduced space where Nelder-Mead algorithm is applied. The final point is guaranteed to be at least as good as the previous best point, and thus it is used as the new starting point to generate the next 2*n* neighboring points. The iterations continue until the difference in the objective function values is within tolerance or the number of iterations reaches a limit.

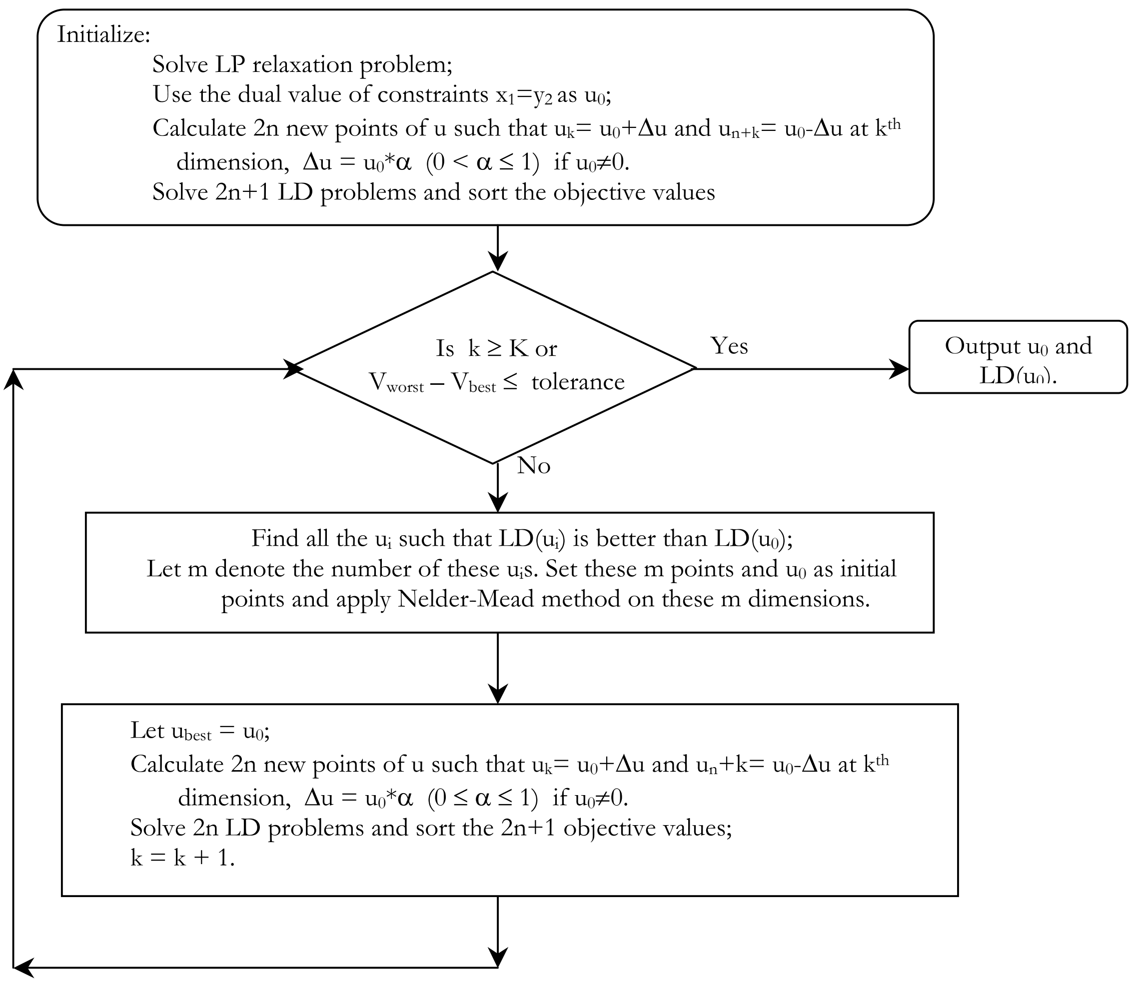

Figure 6-4: Proposed Approach for Lagrangean Decomposition Using Nelder-Mead Algorithm

Since each Nelder-Mead iteration returns a new point with equal or better objective function than the previous point, convergence is guaranteed. However the algorithm's efficiency depends on the value α. A large value of α gives more emphasis in the most promising directions in Lagrangean multiplier space and results in larger changes; while small values of α concentrates on small areas and attempts to find all promising directions although it may result in slower convergence. An adaptive strategy is thus proposed starting with a value of α at the initial iterations in order to improve the values of the Lagrangean multiplier faster, and reducing α when no new improving directions can be found. This strategy will be illustrated in the following section where first the motivating example of section 6.2 is revisited using the proposed approach and the performance is compared with the subgradient method. An additional example and case studies in the area of process operation are considered to illustrate the importance of the proposed approach in the solution of large- scale problems.

6.4 CASE STUDIES

6.4.1 MOTIVATING EXAMPLE

Unlike the subgradient method, the proposed approach based on Nelder-Mead algorithm is not sensitive to the selection of the decomposition variables since it has the advantage of searching for the optimal direction in the Lagrangean multiplier space. As shown in Table 6-5 the proposed algorithm converges to the optimal solution despite of the choice of the decomposing variables. The CPU time in most of the cases is much better than that of the subgradient method. The results are obtained on Pentium 1200 PC with CPLEX 7.5.

Dual variables	Subgradient method		Proposed algorithm	
	Obj.	CPU (sec.)	Obj.	CPU (sec.)
x_1, x_2	26.012	1.19	26.000	0.10
x_2, x_3	26.000	0.27	26.000	0.33
x_1, x_4	26.279	2.40	26.000	0.73
x_3, x_4	26.000	0.12	26.000	0.10

Table 6-5: Comparison of Results and CPU Times for Motivating Example

The optimal solutions in terms of primal and dual variables, Lagrangean multipliers and number of iterations of the proposed algorithm are listed in Table 6-6. The α parameter is fixed to correspond to 0.1% of the original values. When α is used with value of 1% and 0.05%, the algorithm converges to the same solution, however the corresponding numbers of iterations (Table 6-7) are different as discussed in the previous section.

Decomposing variables	(x_1, x_2, x_3, x_4)	u_1, u_2	Dual variables	Dual variable values	Iterations
x_1, x_2	(5,0,4,0)	(0.00, 0.00)	y_1, y_2	(0,0)	5
x_2, x_3	(5,0,0,0)	(3.00, 4.00)	y_2, y_3	(0,4)	21
x_1, x_4	(0,0,4,0)	(2.00, 4.00)	y_1, y_4	(5,0)	34
x_3, x_4	(5,0,4,0)	(0.00, 0.00)	y_3, y_4	(0,0)	5

Table 6-6: Solution of Motivating Example

Decomposing variables	α = 1%	α = 0.1%	α = 0.05%
x_1, x_2	5	5	5
x_2, x_3	13	21	21
x_1, x_4	20	34	39
x_3, x_4	5	5	5

Table 6-7: Number of Iterations Using Different α

6.4.2 EXAMPLE 4

Guignard and Kim (1987) used the following example to illustrate that Lagrangean decomposition generates a bound at least as good as Lagrangean relaxation:

$$\begin{aligned}
\max\ & 2x_1 + 3x_2 + 4x_3 \\
s.t.\ & 12x_1 + 19x_2 + 30x_3 \le 46 \\
& 49x_1 + 40x_2 + 31x_3 \le 76 \\
& x_1, x_2, x_3 \in \{0,1\}
\end{aligned}$$

In this example, the optimal objective value is 4 with the solution of $\{x_1,x_2,x_3)$ $=\{0,0,1\}$. The authors stated that with Lagrangean multipliers of u_1=2, u_2=0.5 and u_3=1.5, Lagrangean decomposition can achieve an upper bound of 4.5, tighter than the upper bound of 5.84 obtained by Lagrangean relaxation. Using the LP relaxation to initialize Lagrangean multipliers of u_1,u_2,u_3, subgradient method converges to the optimal value of 4.5 in 39 iterations; while the proposed approach reduces the number of iterations by 25% (29 iterations vs. 39 iterations). The α parameter is set to an initial value of 1 and reduced to 60% of its value when promising new directions cannot be obtained.

6.4.3 SCHEDULING PROBLEM

The proposed Lagrangean decomposition approach is used for example 1 in section 1.2. The schedule considered in this section spans over three time periods of 8, 8 and 12 hours, respectively. Simultaneous solution of this problem is computationally expensive mainly due to the large number of binary variables involved. With Lagrangean decomposition, the problem is decomposed into 3 subproblems by dualizing the storage variables at the end of each time period.

Both subgradient method and the proposed algorithm are utilized to obtain an upper bound whereas a lower bound is generated by fixing the binary variables for the first period from the Lagrangean decomposition solution. The results are obtained on Sun Ultra 60 workstation using CPLEX 7.5. Table 6-8 compares the results of simultaneous solution, subgradient method and the proposed algorithm for a maximally allowable computation time of 10,000 sec.

	Simultaneous solution	Lagrangean decomposition with subgradient method	Lagrangean decomposition with the proposed algorithm
Lower bound (feasible schedule)	7067.90	7115.31	7115.31
Upper bound (LD Obj.)	7220.79[1]	7185.78	7176.11
Relative gap (%)	2.16%	0.99%	0.85%

[1] the upper bound provided by the solver at the time of termination.

Table 6-8: Comparisons for Scheduling Example

For maximum computation time of 20,000 sec, the proposed approach results in an improved upper bound of 7174.93 and lower bound of 7165.74 with 0.13 % gap, while the simultaneous solution and subgradient method provide no improvement in the solution.

6.4.4 SCHEDULING PROBLEM WITH UNCERTAINTY

Uncertainty receives lots of attention in the research of scheduling and planning problems. A commonly used approach is a multi-period optimization that utilizes a set of scenarios to represent the possible parameter values (Dantzig, 1955). This however, increases the problem size due to the scenarios introduced. Lagrangean decomposition is thus an ideal technique to use in order to efficiently address the problem of scheduling under uncertainty.

The problem in section 3.2.3 is considered here. Two periods are considered, a first period of 6 hours where the parameters are assumed deterministic and a second period of 6 hours with uncertainty in demand. In this example, 3 scenarios are used for the second period to describe the uncertain demands corresponding to high, medium and low probability of occurrence. The objective function combines the profit of both periods and thus determines the optimal production schedule of the first period that will benefit most the entire time horizon under consideration. The mathematical formulation of this problem corresponds to a MILP problem, which was solved on Sun Ultra 60 workstation using CPLEX 7.5. The branch-and-bound algorithm is not efficient for this problem as shown by the results presented in Table 6-9. We thus applied the proposed Lagrangean decomposition approach for solving this problem. First the original planning problem is decomposed into 4 subproblems by dualizing the storage variables between the first and second periods. The objective function of Lagrangean decomposition corresponds to an upper bound of the original optimization problem while a lower bound is generated by fixing the binary variables of the first period at the values of the Lagrangean decomposition solution and solving the original problem. Table 6-9 compares the objective value, upper bound and relative gap obtained using the proposed approach and solving the original problem using branch-and-bound.

	Directly solving the problem	Proposed Lagrangean decomposition approach
Iteration	7th Feasible solution	2
CPU (sec.)	30,000 [1]	17,995
Objective function value	4.131E5	4.148E5
Upper bound	4.740E5 [2]	4.329E5
Relative Gap	14.7%	4.4%

[1] Computational time limit

[2] the upper bound provided by the solver at the time of termination.

Table 6-9: Results of the Scheduling Problem with Uncertainty

It should be pointed out that the upper bound is improving with the number of iterations. For example the Lagrangean decomposition value is reduced to 4.300E5 with a relative gap of 3.7% at iteration 9. However we need to balance the tradeoff between the objective function improvement and the computational efficiency. Note that for this problem the subgradient method could not get an improved solution after the first iteration resulting in a Lagrangean decomposition value of 4.347E5.

6.5 SUMMARY

This chapter presents a Lagrangean decomposition scheme using a modified Nelder-Mead algorithm to search for promising directions in Lagrangean multiplier space. At each iteration the proposed method provides a solution at least as good as that of the previous iteration, thus enabling better convergence. Focusing only on promising directions prevents the algorithm from evaluating large number of points, which is the main disadvantage of direct search algorithms. The case studies illustrate that the algorithm can be utilized as an alternative for the problems where subgradient method performs poorly. It can also be employed in combination with subgradient method in a scheme where this method was only utilized when subgradient stops improving the objective function. Special emphasis has been placed in the performance of the proposed approach to scheduling problems where promising results are obtained.

CHAPTER 7

DISCUSSION AND FUTURE DIRECTIONS

7.1 COMPREHENSIVE MODELS FOR SHORT-TERM SCHEDULING APPLICATIONS

In the proposed hierarchical approach for production planning and scheduling described in chapter 4, a continuous-time formulation is presented for the scheduling problem which utilizes less number of variables especially binary variables and constraints. This model works efficiently as illustrated with the case study in chapter 4. However, scheduling is a complex problem in practice which involves many considerations such as profit, staffing and safety. The solution of our model is only economically justified as the optimum, but it may not be the best choice when evaluated from all perspectives of a plant. Ideally we would like to model every constraint and solve for a solution balancing all considerations. Unfortunately this is not the case for realistic industry problems due to the following reasons. First, modeling all the considerations is an elaborate work. Due to the management policies and other ad-hoc situations, the constraints are unique for every plant, which makes the problem case-dependent. Thus it is hard to utilize a general short-term scheduling model for all production problems. Second, it is not easy to set priority for all these considerations. For example, some are hard constraints while the others can be relaxed under certain circumstances. This will be translated into the mathematical model using a large number of binary variables, which dramatically increase the complexity of the problem solving. Third and most importantly, current optimization techniques are still far from being able to handle large-scale MILP and MINLP problems.

Comprehensive models do not necessarily result in good solutions in an affordable time using the available commercial solvers. All these limitations bring us a question: how can we make the scheduling model take part in business decisions for realistic industry problems.

Short-term scheduling problems share some common characteristics such as material flow and sequencing requirements, which are usually the core consideration for a production schedule. This suggests us to design a solution framework that considers different type of constraints hierarchically. In a higher level (or core model), a generalized mathematical model can be used to optimize the production economic goal such as the proposed scheduling formulation and those by other authors. In other levels, different consideration can be modeled separately. For example, staffing requirements can be considered in an independent model. The solution process can employ different strategies in order to obtain the decision. It may make the decisions flow from most important levels to less important ones. For example, the optimal production schedule as well as the second optimal and other alternative schedules obtained from the core model can be considered by the staffing model, where their feasibility are evaluated in terms of staffing requirements. The main reasons of adopting a hierarchical solution strategy are to 1) decompose the entire large problem into smaller models that can be solved separately; 2) keep core model as general as possible and make the rest models interchangeable such that the modeling time for a new application can be largely reduced; 3) make the data collection easier when dealing with different function groups in a plant. The hierarchical structure can also rely on the nature of industry and require experience in decision-making. However the anticipated big savings by utilizing optimization techniques motivate the introduction of short-term scheduling optimization to industry applications.

7.2 SUPPLY CHAIN MANAGEMENT

In a highly integrated global market, supply chain management becomes extremely important in order to meet the requirements of fast response to custom demands, optimal resources allocation, development of new product and survival from competitors. An efficient management of supply chain will result in significant savings to the enterprises' investing capital in inventory and logistics costs, which makes a distinguished impact on business in today's highly competitive environment.

A typical supply chain involves a number of plants, product warehouses, product distribution centers and retailers. Decisions that are optimized independently for each site do not guarantee the optimum of overall objective for the global supply chain. Thus an approach needs to be developed to integrate local decision-making tools such as planning and scheduling, inventory management, transportation optimization, trading optimization in order to model and analyze the whole supply chain, determine the optimal production and transportation plan as well as the detailed schedules of production and storage with given conditions.

The books by Handfielf and Nicholos (1999), and Shapiro (2001) are great reviews of this field. In addition, an extensive literature review of supply chain models was presented by Vidal and Goeschalckx (1997). Among the various industrial applications, Schenk (1998) and Dempster et al. (2000) presented a supply chain model for oil company. Kafoglis (1999) addressed the application of supply chain management in refinery operations. Other work includes Bodington and Shobys (1996) and Zhou et al. (2000) for petrochemical industry, Papageorgiou et al. (2001) for pharmaceutical industry, Philpott and Everett (2001) for paper industry and Escudero et al. (1999) for automotive sector. Edgar et al. (2001) presented a novel approach for supply chain management. In their proposed framework, they capture the dynamic property of decentralized supply chain and they compare different control laws to improve the performance of the supply chain. However, most of the existing approaches lack of an integrated dynamic procedure that considers the entire supply chain, especially the production scheduling, a very important factor for chemical industry. For example, Edgar's model only considers single-stage multi-product batch reactor such that one kind of product can be produced at a time. Production optimization was proven to be a critical bottleneck in chemical industry, which directly influences the performance of the entire supply chain. Therefore, a supply chain management model needs to address the complexity of production planning and scheduling and thus involve the results from the previous chapters.

The following assumptions are considered in supply chain modeling.

i) The supply chain contains several independent nodes as plants, product warehouses, distribution centers and retailers.

ii) Instead of having a centralized system to make decisions for all levels, all the decisions are made locally although they are highly related to the overall optimum of supply chain.

iii) Order is received by retailers first, then comes to the nodes which they are connected, i.e. orders are placed from node to node inside the chain.

iv) Each node makes decisions only when it has been assigned an order for processing.

We are interested in how the supply chain satisfies orders while minimizing the cost or maximizing the profits, how they are acting when unexpected events happen such as rush orders or order cancellations and how production planning affects the performance of supply chain. This involves the determination of amount, location and timing to buy raw materials, production planning decisions, products transportation and inventory policies. There are a number of challenges in addressing this problem.

(1) It is a highly dynamic problem. Orders occur frequently and there is a great degree of freedom in operating conditions of each node.

(2) The hierarchical structure is more complicated than planning problems since decisions at one level may impact those at level above and beneath. For example, poor production planning in one plant results in backorders and thus impact the ability of the whole supply chain to satisfy orders. On the other hand, excessive production causes overstock which increases the inventory cost. Therefore, nodes interaction is a crucial part aiming at improving the performance of the whole supply chain.

(3) At each stage, the optimal solution of large-scale problems is required which calls for efficient solution methodologies.

A hierarchical approach with bottleneck identification can be adopted. The idea is to identify the bottlenecks in production, inventory management and transportation before making top-down decisions. These bottlenecks can be translated to operating constraints and added in the decision model at top level so that the solution brings reasonable results to the lower level. In other words, this approach attempts to provide a better initial point for the following recursive decisions. The bottleneck identification decomposes the overall problem into smaller ones that can be solved simultaneously. Future work includes the development of a recursive mechanism and building of mathematical models at all levels.

References

Acevedo, J. and Pistikopoulos, E. N. (1998). Stochastic optimization based algorithms for process synthesis under uncertainty. *Comput. Chem. Eng.,* 22(4/5), 647-672.

Adhya, N., Tawarmalani, M, Sahinidis, N.V. A Lagrangean Approach to the pooling problem, *Ind. Eng. Chem. Res.*, 38, 1956.

Applequist, G., Samikoglu, O., Pekny, J., & Reklaitis, G. V. (1997). Issues in the use, design and evolution of process scheduling and planning systems, ISA Transactions, 36 (2), 81

Basset, M.H., Doyle III, F.J., Kudva, G.K., Pekny, J.F., Reklaitis, G.V., Subrahmanyam, S., Zentner M.G. and Miller. D.L. (1994). Perspectives on Model-based Integration of Process Operations. *Proc. fifth Int. Symp. on Process systems Eng.* Kyongju, Korea.

Basset, M.H., Pekny, J.F., & Reklaitis, G.V. (1996). Decomposition Techniques for the Solution of Large-Scale Scheduling Problems. *AICHE J.* 42,12, 3373.

Benders, J.F. (1962). Partitioning Procedures for Solving Mixed Variables Programming Problems. *Numberische Mathematik,* 4, 238.

Bilde and Krarup (1977). Sharp Lower Bounds and Efficient Algorithms for the Simple Plant Location Problem. Ann. Dis. Math. 1, 79-97

Biegler, L.T. Grossmann I.E. and Westerberg, A.W. (1997) Systematic Methods of Chemical Process Design, Section A.3.4, *Prentice Hall*

Birge, J.R. and Louveaux, F. (1997). *Introduction of stochastic programming*, New York, Springer.

Bodington, C.E. and Shobrys, D.E. (1996). Optimize the supply chain. *Hydrocarbon Processing*, Vol. 75, No.9, pp.5

Bose, S. and Pekny, J. (2000). A model predictive framework for planning and scheduling problems: a case study of consumer goods supply chain. *Comput. Chem. Eng.*, 24:329-335

Boskma, K. (1982). Aggregation and the Design of Models for Medium-term Planning of Production. *Eur. J. Oper. Res.* 10, 244-249.

Castro, P., Barbosa-Povoa, A. and Matos, H. (2003). Optimal Periodic Scheduling of Batch Plants Using RTN-based Discrete and Continuous-time Formulations: A Case Study Approach. *Ind. Eng. Chem. Res.*, 42, 3346-3360

Clay, R. L. and Grossmann, I. E. (1997). A Disaggregation Algorithm for the Optimization of Stochastic Planning Models. *Comput. Chem. Eng. 21*, 751.

Crowder H. (1976), Computational Improvements for Subgradient Optimization, Symposia Mathematica, Vol. XIX, *Academic Press*, London.

Dantzig, G.B. (1955). Linear Programming Under Uncertainty. *Manage. Sci.*, 1, 197-206

Dantzig, G.B. (1963). Linear Programming and Extensions. *Princeton University Press*, Princeton, NJ.

Dantzig, G.B. (1989). Decomposition Techniques for Large-Scale Electric Power Systems Planning Under Uncertainty. *Ann. Oper. Res.*

Dempster, M.A.H., Pedron, N.H., Medova, E.A., Scott, J.E and Sembos, A. (2000). Planning logistics operations in the oil industry, *Jour. Oper. Res. Socie.*, 51, pp. 1271-1288

Diaby, M., Bahl, H.C., Karwan, M.H. and Zionts, S. (1992). A Lagrangean Relaxation Approach for Very-Large-Scale Capacitated Lot-Sizing. Manag. Sci. 9,1329.

Dimitriadis, A.D., shah, N. and Pantelides, C.C. (1997). RTN-bases Rolling Horizon Algorithms for Medium Term Scheduling of Multipurpose Plants. *Comput. Chem. Eng.*, S21, S1061

Equi, L., Giorgio, G., Marziale, S. and Weintraub, A. (1997) A combined transportation an scheduling problem, *Euro. Jour. Oper. Res.*, 97, 94.

Erlenkotter, D. (1978). A Dual-Based Procedure for Uncapacitated Facility Location, *Oper. Res.* Vol.26, 1, 992-1009

Escudero, L.F., Galindo E., Garcia G., Gomez E. and Sabau V. (1999), Schumann: a modeling framework for supply chain management under uncertainty, *Euro. Jour. Oper. Res.*, 119, pp. 14-34

Feltenmark, S. (1997). On optimization of power production, *Doctoral Thesis, Department of Mathematics/optimization and systems theory, Royal Institute of Technology,* Sweden

Fisher, M.L. (1973). Optimal Solution of Scheduling Problems Using Lagrange Multipliers: Part I, *Oper. Res.*, Vol.21 1114-1127

Fisher, M.L. and Hochbaum, D.S. (1980). Database Location in a Computer Network. *Journ. ACM.* 27, 718-735

Fisher, M.L. (1981). The Lagrangean Relaxation Method for Solving Integer Programming Problems. *Manag. Sci.* 27, 1.

Fisher, M.L. and Kedia, P. (1990). Optimal Solution of Set Covering/Partitioning Problems Using Dual Heuristics. *Manag. Sci.* 39, 67-88

Floudas, C.A. (1995). Nonlinear and Mixed Integer Optimization: Fundamentals and Applications, *Oxford University Press*

GAMS Solver Manual © GAMS Development Corporation

Geoffrion, A.M. (1974). Lagrangean Relaxation and Its Uses in Integer Programming. *Math. Program. Stud.* 2, 82.

Graves, S.C. (1982) Using Lagrangean Techniques to Solve Hierarchical Production Planning problems. *Manag. Sci.* 28, 260.

Grossmann, I.E., Viswanathan, J., Vecchietti, A., Raman, R. and Kalvelagen, E. (2002). GAMS/DICOPT: A discrete continuous optimization package. *GAMS Corporation Inc.*

Guignard, M., & Kim, S. (1987). Lagrangean Decomposition: A model yielding stronger Lagrangean bounds, *Math. Prog.* 39, 215.

Guignard, M. and Rosenwein, M.B. (1990). An Application of Lagrangean Decomposition to the Resource-Constrained Minimum Weighted Arborescence Problem. *Networks.* 20, 345-359

Guignard, M. (1995). Lagrangean Relaxation: A Short Course. *Belgian. Jour. Oper. Res.* Vol.35. (3-4) 95

Guignard, M. (2003). Lagrangean Relaxation. *Top*, Vol.11 (2) 151-228

Gupta, A. & Maranas, C. D. (1999). A Hierarchical Lagrangean Relaxation Procedure for Solving Midterm Planning Problem. *Ind. Eng. Chem. Res.* 38, 1937.

Handfield, R.B. and Nichols Jr, E.L., (1999). Introduction to supply chain management, *Prentice-Hall Inc.*, New Jersey.

Harjunkoski, I. and Grossmann, I.E. (2001). A decomposition approach for the scheduling of a steel plant production. *Comput. Chem. Eng.*, 25:1647-1660

Heever, S.A.V.D. and Grossmann, I.E. (1999). Disjunctive multiperiod optimization methods for design and planning of chemical process systems. *Comput. Chem. Eng.*, 23:1075-1095

Held, M and Karp, R. M. (1970). The Traveling Salesman Problem and Minimum Spanning Trees, *Oper. Res.* Vol.18 1138-1162

Holmberg, K. and Yuan, D. (1981). A Lagrangean relaxation method for solving integer programming problems, *Manag. Sci,* 27 (1), 1.

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

Ierapetritou, M.G., Hene, T.S. and Floudas, C.A. (1999) Effective Continuous-Time Formulation for Short-Term Scheduling. 3. Multiple Intermediate Due Dates. *Ind. Eng. Chem. Res.*, 1999; 38, 3446-3461.

Johnson, E. L., Nemhauser, G. L. and Savelsbergh, M. W. P. (2000). Progress in linear programming based branch and bound algorithm: exposition. *INFORMS Jour Comput.*, 12(1), 2-23

Kafoglis, C.C. (1999). Maximize competitive advantage with a supply chain vision, *Hydrocarbon Processing*, Vol. 78, No. 2, pp.47-50

Kall, P. and Wallace, S. W. (1994). *Stochastic programming*, New York, Wiley.

Karimi, I.A. and McDonald, C. M. (1997). Planning and scheduling of parallel semicontinuous processes. 2. Short-term scheduling. *Ind. Eng. Chem. Res.*, 36:2701-2714

Kelley, J.E. (1960). The Cutting-Plane Method for Solving Convex Programs. *Journ. SIAM*, 8, 703-712

Kelley, C.T. (1999), Iterative Methods for Optimization, *siam*.

Khmelnitsky, E., Kogan, K., & Maimon, O. (2000). A Time-Decomposition Method for Sequence-Dependent Setup Scheduling under Pressing Demand Conditions. *IEEE Trans. Anl. Control*, 45, 4.

Kondili, E. and Pantelides, C.C. and Sargent, R.W.H. (1993a). A general algorithm for scheduling batch operations-Part I. MILP formulation. *Comput. Chem. Eng.*, 17:211-227

Kondili, E. and Shah, N. and Pantelides, C.C. (1993b) Production planning for the rational use of energy in multiproduct continuous plants. *Comput. Chem. Eng.*, 17

Lemarechal, C. (1974) An Algorithm for Minimizing Convex Functions, *Proceedings IFIP'74 Congress.* North Holland. 552-556

Li, P., Wendt, M. and Wozny, G. (2000). Robust Model Predictive Control under Chance Constraints. *Comput. Chem. Eng. 24*, 829.

Liu, M. and Sahinidis, N. V. (1996). Long range planning in the process industries: a projection approach. *Comput. Ops. Res.*, 23:237-253

Makridakis, S., Wheelwright, S.C. and McGee, V.E. (1983) Forecasting: Methods and Applications, *John Wiley and Sons*

Maranas, C. D. (1997). Optimization Accounting for Property Prediction Uncertainty in Polymer Design, *Comput. Chem. Eng.* 21, Suppl., 1019-1024.

Marin, A. and Pelegrin, B. (1998) The return plant location problem: Modeling and resolution. *Euro. Jour. Oper. Res*. 104, 375

McDonald, C.M. and Karimi, I.A. (1997). Planning and scheduling of parallel semicontinuous processes. 1. Production planning. *Ind. Eng. Chem. Res.*, 36:2691-2700

Mockus, L., and Reklaitis, G. (1996). A new global optimization algorithm for batch process scheduling. In state of the art in global optimization. Floudas, C., Pardalos, P., Eds., Kluwer Academic Publishers: Dordrecht, The Netherlands, 521-538

Mockus, L., and Reklaitis, G. V. (1996). Continuous-time representation in batch/semi-continuous processes: randomized heuristic approach. *Comput. Chem. Eng.*, 20, S1173-1177.

Nelder, J.A. and Mead, R. (1965) A Simplex Method for Function Minimization. *Comput. J.* 308-313

Orcun, S. and Altinel, I.K. and Hortacsu, O. (2001). General continuous-time models for production planning and scheduling of batch processing plants: Mixed Integer linear program formulations and computational issues. *Comput. Chem. Eng.*, 25:371-389

Orero, S.O. and Irving, M.R. (1997) A Combination of the Genetic Algorithm and Lagrangian Relaxation Decomposition Techniques for the Generation Unit Commitment Problem. *Electric Power Sys. Res.* 43, 149-156

Olaf, E. F, Alexander, H.G. & Kan, R. (1993). Decomposition in General Mathematical Programming. *Math. Prog*. 60, 361.

Pantelides, C.C. (1994) Unified frameworks for optimal process planning and scheduling. *Proceedings of the Second conference on FOCAPO*. p. 253-274

Papageorgiou, L.G. and Pantelides, C.C. (2000). A hierarchical approach for campaign planning of multipurpose batch plants. *Comput. Chem. Eng.*, 17:S27-S32

Papageorgiou, L.G., Rotstein G.E. and Shah N. (2002), Strategic supply chain optimization for the pharmaceutical industries, *Ind. Eng. Chem. Res.*, 40, pp. 275-286

Pekny, J. F., and Reklaitis, G. V. (1998). Towards the convergence of theory an practice: a technology guide for scheduling/planning methodology. *American Institute of Chemical Engineering Symposium Series*, 94(320), 91-111.

Perea-Lopez, E., Grossmann, I. E. and Ydstie, B. E., Dynamic Modeling and Decentralized Control of Supply Chains. *Ind. Eng. Chem. Res*., 40, 3369-3383.

Pinto, J. and Grossmann, I. E. (1995). A continuous-time mixed integer linear programming model for scheduling of multistage batch plants. *I&EC Res.*, 34, 3037-3051.

Pinto, J. and Grossmann, I. E. (1998). Assignment and sequencing models for the scheduling of chemical processes. *Ann. Oper. Res.*, 81, 433-466.

Pistikopoulos, E. N. and Ierapetritou, M. G. (1995). Novel approach for optimal process design under uncertainty. *Comput. Chem. Eng.*, 19(10), 1089-1110.

Prekopa, A. (1995). *Stochastic programming*, Kluwer: Dordrecht, The Netherlands.

Rana, K and Vickson, R.G. (1991) Routing Container Ships Using Lagrangean Relaxation and Decomposition, *Trans. Sci*. 25(3).

Reklaitis, G. V. (1992), Overview of scheduling and planning of batch process operations. NATO advanced study institute-batch process systems engineering, Antalya, Turkey.

Sahinidis, N.V and Grossmann, I.E. (1989). Optimization model for long range planning in the chemical industry. *Comput. Chem. Eng.*, 13:1049-1063

Sahinidis, N.V and Grossmann, I.E. (1991). Reformulation of multiperiod MILP models for planning and scheduling of chemical processes. *Comput. Chem. Eng.*, 15, 255-272.

Sahinidis, N.V. and Tawarmalani, M. (2002). GAMS/BARON 5.0: Global optimization of mixed-integer nonlinear programs. *GAMS Corporation Inc*.

Sand, G., Engell, S., Märkert, A., Schultz, R. and Schultz, C. (2000). Approximation of An Ideal Online Scheduler for A Multiproduct Batch Plant. *Comput. Chem. Eng*., 24, 361-367

SAS Institute Inc. (1999). SAS OnlineDoc®, Version 8, Cary, NC: SAS Institute Inc.

Scheck, A.R.J. (1998). Resource sharing and strategic alliances-reduction of oil field costs through innovative supply chain management, *APPEA J.*, pp. 570-576

Schwarm, A. T. and Nikolaou, M. (1999). Chance-constrained Model Predictive Control. *AIChE J. 45*, 1743.

Subrahmanyam, S., Pekny, J. F. and Reklaitis, G. V. (1994). Design of Batch Chemical Plants under Market Uncertainty. *Ind. Eng. Chem. Res. 33*, 2688.

Philpott, A. and Everett G. (2001). Supply chain optimization in the paper industry, *Appita Journal*, 54(2), pp. 275-286

Schilling, G. and Pantelides, C.C. (1999). Optimal periodic scheduling of multipurpose plants. Comput. Chem. Eng., 23:635-655

Shah, N. and Pantelides, C.C and Sargent, R.W.H. (1993). Optimal periodic scheduling of multipurpose batch plants, *Ann. Oper. Res.*, 42:193-228

Shah, N., Pantelides, C.C., & Sargent, R.W.H. (1993). A general algorithm for short-term scheduling of batch operations – II. computational issues. *Comput. Chem. Eng.,* 17, 229

Shah, N. (1998). Single and multisite planning and scheduling: current status and future challenges. In: J. Pekny, G. Blau, Foundations of computer aided process operations, American Institute of Chemical Engineers symposium Series 320, 94, pp. 91–110.

Shapiro, J.F. (2001). Modeling the supply chain, *Duxbury Thomson Learning Inc.*, Duxbury, Pacific Grove, CA.

Smith, A.R. and Klosek, J. (2001) A review of air separation technologies and their integration with energy conversion processes Fuel Processing Technology 70 (2): 115-134

Thizy, J.M. and Wassenhove, L.N.V. (1985) Lagrangean Relaxation for the Multi-Iterm Capacitated Lot-Sizing Problem: A Heuristic Implementation. *IIE Trans*, 17, 308.

Turkay, M. and Grossmann, I.E. (1996). Logic-based MINLP algorithms for the optimization of process systems with discontinuous investment costs-multiple size regions. *Ind. Eng. Chem. Res.*, 35:2611-2623

Van den Heever, S.A. and Grossmann, I.E. (2003). A Strategy for the Integration of Production Planning and reactive Scheduling in the Optimization of a Hydrogen Supply Network. *Chem. Eng.*, 27, 1813-1839

Vidal, C.J. and Goetschalckx, M. (1997). Strategic production-distribution models: A critical review with emphasis on global supply chain models, *Euro. Jour. Oper. Res.*, 98, pp. 1-18

Wu, D. and Ierapetritou, M.G. (2003), Decomposition Approaches for the Efficient Solution of the Short-term Scheduling Problems, *Comp. Chem. Eng.* 27, 1261-1276

Wu, D. and Ierapetritou, I.G. (2004). Cyclic Short-term Scheduling of Multiproduct Batch Plants Using Continuous-time Representation. *Comput. Chem. Eng.*, 28, 2271-2286

Zhang, X., & Sargent, R. (1996). The optimal operation of mixed production facilities – general formulation and some solution approaches for the solution. *Comput. Chem. Eng.*, 20, S897-S904

Zhou Z, Cheng C. and Hua B. (2000). Supply chain optimization of continuous process industries with sustainability considerations, *Comput. Chem. Eng.,* 24, pp. 1151-1158

Zhu, X.X. and Majozi,T. (2001). Novel continuous-time MILP formulation for multipurpose batch plants. 2. Integrated planning and scheduling. *Ind. Eng. Chem. Res.*, 40:5621-5634

Zowe, J. (1985). Nondifferentiable Optimization. In: Schittkowski, K., *Comput. Math. Progrm.*, NATO ASI Series F: Computer and Systems Science, 15. Springer-Verlag, 323-356

Curriculum Vita

Dan Wu

Sep. 99 – Oct. 05 Rutgers, the State University of New Jersey
Chemical and Biochemical Engineering Ph.D.
Sep. 94 – Jun. 99 Tsinghua University
Chemical Engineering B.S.
Sep. 96 – Jun. 99 Tsinghua University
Economics B.S.

D. Wu and M.G. Ierapetritou, "Cyclic Short-term Scheduling of Multiproduct Batch Plants Using Continuous-time Representation", Computers & Chemical Engineering, 28, 2271-2286, 2004

D. Wu and M.G. Ierapetritou, "Hierarchical Approach for Production Planning and Scheduling Under Uncertainty Using Continuous-time Formulation", Foundations of Computer-Aided Process Design 2004 Conference Proceedings, Princeton University, July 2004

D. Wu and M.G. Ierapetritou, "Decomposition Approaches for the Efficient Solution of Short-term Scheduling Problems", Computers & Chemical Engineering, 27, 1261-1276, 2003

D. Wu and M.G. Ierapetritou, "A Continuous-time Formulation for Simultaneous Consideration of Planning and Scheduling Decisions", Foundations of Computer-Aided Process Operations 2003 Conference Proceedings, Coral Springs, January 2003

M.G. Ierapetritou, D. Wu, J. Vin, P. Sweeney, M. Chigirinskiy. "Cost Minimization in an Energy-Intensive Plant Using Mathematical Programming Approaches", Industrial & Engineering Chemistry Research, 41,5262-5277, 2002